Highly Available Location-based Services in Mobile Environments

Peter Ibach and Matthias Horbank

Humboldt University, Computer Science Department, Computer Architecture and Communication Group, Rudower Chaussee 25, 12489 Berlin, Germany
{ibach, horbank}@informatik.hu-berlin.de

"*We're not lost. We're locationally challenged.*"
John M. Ford

Abstract. *We show how to use Web Services standards for propagation, discovery, and composition of location-based services in mobile environments. To achieve semantic interoperability we express location information in XML using context-specific ontologies. The achieved interoperability allows for context-aware on-demand service composition making the composite service highly available and resilient to environmental dynamics and uncertainties.*

1 Introduction

Location-based services (LBS) are services that utilize their ability of location-awareness to simplify user interactions and adapt to the specific context. With advances in automatic position sensing and wireless connectivity, the application range of mobile LBS is rapidly developing, particularly in the field of geographic, telematic, touristic, and logistic information systems.

However, present LBS are to a large extent incompatible with each other and unable to interoperate on location semantics. They are mostly bound to a specific technology reflecting the preferences of the service provider. Typically, proprietary protocols and interfaces are employed to aggregate the different system components for positioning, networking, content, or payment services. In many cases, these components are glued together to form a monolithic and inflexible system. If such a system has to be adapted to another technology, e.g., the change from GPS positioning to inhouse WLAN or Bluetooth positioning, it has to be entirely reengineered. Due to the dynamic nature of mobile environments, available resources as well as achievable quality of service levels are incessantly changing. Thus, adaptivity – the ability of steady interoperation of variable resources under changeable connection conditions – becomes crucial for service end-to-end availability in mobile environments.

Let us consider a position sensing service, for example, a satellite-based GPS. If a mobile device moves from outdoor to indoor environments, the signal will likely become unavailable and position sensing will fail. Without the location information expected from this subservice, composite services depending on it will become unavailable as well. To arrive at seamless operation, on-the-fly switchover to an alternative position sensing service using a different technology is required. To choose from multiple possible position sensing services, the decision has to consider service availability, quality of service properties, and costs. In the near future, most mobile and wearable devices are expected to have multiple position sensing technologies at disposal, e.g., GPS, GSM, WLAN, and Bluetooth. Nevertheless, new technologies, like at present WiMax or RFID, are continuously emerging. Thus hardware devices and software components, their interfaces and architecture have to be able to deal with changing conditions to make mobile location-based services highly available.

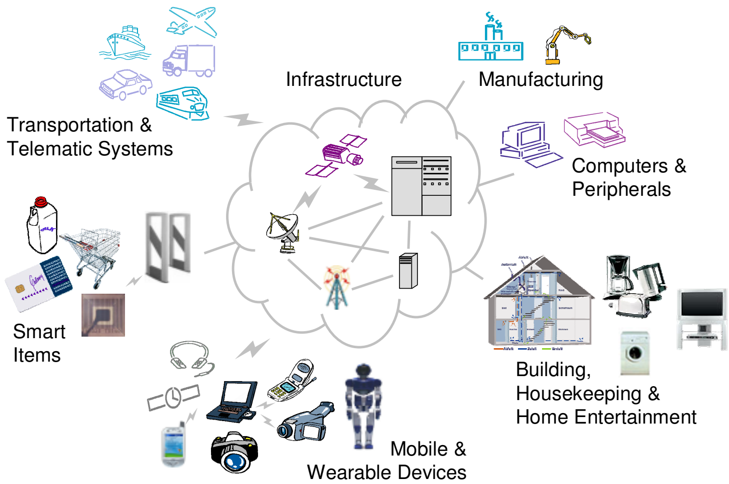

Fig. 1. NOMADS – *N*etworks *o*f *M*obile *A*daptive *D*ependable *S*ystems (for further investigations see [8]). High mobility and dynamics require systems' self-reliance, context awareness, and adaptivity to accomplish dependable operation

Generally, flexible and open standards are the key to enable interoperability and open the door for a prospering variety of new business ideas and services. Web Services Grid and Semantic Web standards are expected to overcome past hurdles by a universal standard for managing services and resource on the web. These standards might accelerate the next chapter of the electronic revolution. Frameworks such as *NOMADS* (see Fig. 1) promise to deliver mobile, adaptive, and dependable interoperability, challenging numerous new applications and business opportunities.

Here, we investigate applicability of Web Services Grid standards in the envisaged domain of location-based services in mobile environments and propose a generic model for improved flexibility and availability. Core strategy is the on-the-fly adaptation to a specific situation. This is accomplished by tracking contextual changes and context-aware on-demand composition of appropriate services.

The paper is structured as follows: In Section 2 we summarize background and related work in the area of location-aware computing. In Section 3 we discuss the economic impact of location-based services, service-oriented architectures, and the challenges to be tackled. Section 4 describes the technical approach we are pursuing, demonstrates how it helps delivering improved *m*obility, *a*daptivity, and *d*ependability (we call these the *MAD* properties), and how location information can be processed semantically. Section 5 describes our implementation experiences and some use case scenarios and Section 6 concludes the results and gives an outlook to future work.

2 Background and Related Work

While in the 60s and 70s mainframe computing flourished and in the 80s and 90s personal computing got predominant we are now entering the age of ubiquitous computing [20] where – in the near future – almost any device will have certain computation and communication power. Pervasive computing further emphasizes the expectation that computational devices and connecting infrastructure are getting more and more omnipresent. With the widespread of mobile and wearable devices capable of location sensing, lots of research has focused on location-based services [14] and location-aware computing [5]. A central problem in context-aware computing is the appropriate description of environmental characteristics [10].

Another technology leap is induced by Radio Frequency Identification (RFID). Through RFID labels, almost any device or item can be cost-effectively transformed into a “smart” information source. In conjunction with the electronic product code infrastructure (EPC Network), smart items will be capable of wirelessly telling who and where they are, what status they have, and which services they offer. An “Internet of things” [1] with billions and soon trillions of seamlessly interconnected devices is about to take over the era of traditional computers and computer networks.

By these developments, information technology is taking a big step towards providing a comprehensive real-time picture of the physical world. Through the spatial organization of information together with location-based services and semantic interoperability, virtual and real spaces will tightly interconnect.

Since the vast possibilities also involve misuse, security and privacy issues are accompanying the developments and receive growing attention. To ensure that humans are not overwhelmed by omnipresent and omnipotent technology, “Ambient Intelligence” [4] represents a vision of sensitive and responsive electronic environments putting humans in the centre of technological developments.

Considering the question of how to organize the evolving number of interacting devices led to various approaches trying to exploit analogies from examples in nature. A variety of concepts emerged which are inspired by physical, biological, sociological, or economical analogies. Basic idea is to design and structure composite systems such that they are able to meet upcoming challenges by actions of its largely autonomous entities. Under the term “autonomic computing” IBM summarizes eight* core

* In current implementation, they have been compressed to four.

"elements" [9] – comprising self-configuring, self-healing, self-optimizing, and self-protecting – that are intended to guide the development.

Fundamental concept is the composition of systems by extensive reuse of commodity software/hardware components. Component-based software engineering [9] tries to extend paradigms of object-oriented software engineering such that components – in contrast to objects – can be adapted to the specific conditions of a run time environment without additional interventions (hot deployment). The access to a component is exclusively accomplished through its interface. Syntax and semantics of a component should be comprehensively described in its specification.

Agents (also referred to as actors) are especially "intelligent" components that show improved adaptivity in dynamic environments through autonomous goal tracking, context sensitivity, mobility, reactivity and proactivity. This direction is pursued by agent-based software engineering [7, 13] and further extensions for business environments referred to as business agents or agentified enterprise components [17]. Those actors have negotiation capabilities, possess context models to adapt to different deployment contexts, and are able to deal with uncertainties that may arise from unforeseen changes and errors.

To accomplish interoperability on higher levels of semantics, one has to agree on a suitable ontology which defines terminology and relations for each specific application area. A widely accepted ontology that models physical objects and their location is used in the Geographic Information System (GIS), standardized by the OpenGIS Consortium. GPS position sensing together with geocoding services for visualization of geographic information [15] enjoys growing popularity on mobile devices. Yet, seamless outdoor to indoor transitions, global scalability, and changeover to different services, e.g., providing different cartographic material, are usually not addressed. The Physical Markup Language of the EPC Network, standardized by the Auto-ID Center, is intended for product classification, but also allows for spatio-temporal annotations for object tracking and supply chain management. Moreover, the World Wide Web Consortium is extending the Resource Definition Framework to relate Web Content to its associated physical location. In all these attempts, however, expressiveness of location semantics is still in its infancy.

3 Challenges and Expected Benefit

Enterprise applications were initially developed on closed, homogeneous mainframe architectures. In the explosively growing heterogeneous landscape of IT systems in the 80's and 90's integration of intra- and inter-company business processes became one of the most important and most cost-intensive tasks of the IT economy. Due to missing or non-transparent standards many enterprises pursued integration by extremely expensive ad-hoc-solutions. With the spreading of the Internet and the increasing importance of electronic business, open Internet-oriented solutions have emerged. Enterprise-internal monolithic software was broken into smaller, autonomous, and flexible components. This enabled the access to services not only enterprise-internally, but along the whole value chain to suppliers, distributors, and customers. We characterize this observation as a shift from rigid systems to flexible service-based architectures, where open and flexible services are the basic building blocks (see Fig. 2).

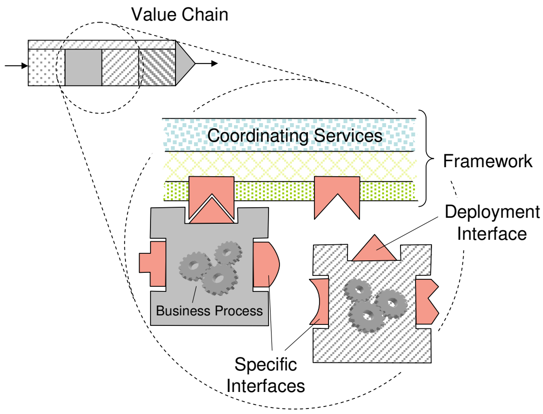

Fig. 2. Loosely coupled, open architectures play an increasingly important role in value chains due to improved possibilities of interoperability, integration, composability, flexibility, reusability and thus increased efficiency at reduced total cost of operation

Grid computing is a service-based form of shared resource usage that is intended to make the access to computation power as simple and omnipresent as electric power supply. It extends peer-to-peer computing as well as cluster computing in a global scale in order to "enable the sharing, selection, and aggregation of geographically distributed heterogeneous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements" [3].

The Web Services Grid is aimed to overcome the two predominant challenges at the same time: uniform access to services *and* processing resources. Web Services are intended to facilitate the application to application interaction extending established Internet technologies. While some skeptics ask why the services paradigm might prevail against preceding concepts like CORBA, significant advantages are:

- market power of stakeholders
- productivity of tools
- integrated support for asynchronous dependable messaging
- service choreography with workflow engines ("two stage programming")
- interoperability and flexibility – paid by some performance loss – through loose coupling based on the eXtensible Markup Language (XML)
- enhanced protocols for propagation, discovery, and invocation of "lightweight" services in ad-hoc networks

Web Services and Grid toolkits like the Globus Toolkit or the Emerging Technology Toolkit (ETTK) based on the Open Grid Services Architecture (OGSA) have helped establishing standards. Component based software for embedded systems [12] and lightweight services [16, 11] expanded the domain to span from distributed client-server applications and globally networked e-business processes down to next generation heterogeneous embedded systems. These developments paved the way towards the general paradigm of service-oriented computing where all kinds of entities are providing, using, searching, or mediating services while efficiently exploiting the available resources. Driving the widespread acceptance of the service-oriented paradigm, LBS might reveal the enormous economic potential of dynamic value webs [6] in mobile business.

4 Adaptive Location-based Services

From the perspective of a mobile user, the environment is ever-changing as he moves from one location to another. As earlier explained, adaptivity to location characteristics is essential for mobile service availability. In our approach, adaptivity of a composite location-based service – we call these services Adaptive Location-Based Services (ALBS) – is accomplished by choosing the appropriate chain of subservices for composition (see Fig. 3).

Prerequisites are general discoverability, interoperability and composability of subservices through standardized communication protocols and directory services. In the Web Services standard, interoperable communication is accomplished by exchanging XML data over HTTP. But Web Services are not restricted to the WWW or a specific protocol. Rather, it is a promising solution for adaptive application synthesis in distributed, dynamically changing environments. The notion of Web Services goes beyond prescribed client-server communication. Emerging standards for directory services such as UDDI [19] or Web Services Choreography (e.g., BPEL4WS [2]) allow for dynamic discovery of services and composition of multiple Web Services to fully-fledged distributed applications.

Consider a location-based service that requires some input, e.g., accurate position information or the user's choice of payment. The user might present these data to the LBS manually. Likewise, this information might be the result of a preceding Web Service which, for example, reads the geographic position from an attached GPS device. In case of payment, information about the user's choice could be sent to an accounting service which, for example, uses a direct debit authorization. For service composition it is not necessary to know how the accounting is actually performed or how access to the GPS device is implemented, as long as one can trust the responsible Web Services. Authorization and trust will be fundamental for the success of location-based services and Web Services composition. Moreover, protecting privacy regarding the user's trace of information is a severe issue. Further dangers of intrusion to take care of are service spamming, where undesired services are propagated, and service spoofing, where insecure services are offered under disguised identity. Ongoing developments in the Web Services Trust Language (WS-Trust) therefore accommodate a wide variety of security models.

4.1 Using Web Services for ALBS Implementation

We use Web Services standards to implement the appropriate selection of subservices and to process their composition. These comprise the service interface description in the Web Services Description Language (WSDL). In an interface description the *port* *type* specifies the service's request/response-behavior. A service instance is accessed through a *port*. Each port has to bind to a port type and has to support additional binding information, e.g., the used protocol. In Web environments the Simple Object Access Protocol (SOAP) might be a primary candidate, but other protocols such as CICS (Customer Information Control System) or dependable message queuing can be utilized as well.

For each application to be composed of subservices, a flow through *required* port types and *optional* port types guides the composition process. This flow can be specified using choreography languages (e.g., WSCL or BPEL4WS). Ongoing developments in Web Services Choreography incorporate far-reaching flow control where, for example, a group of services can be processed transactional or the flow may branch with respect to optional services availability or in case exception occurs. The composition process can be graphically expressed by a path through a network of accessible ports:

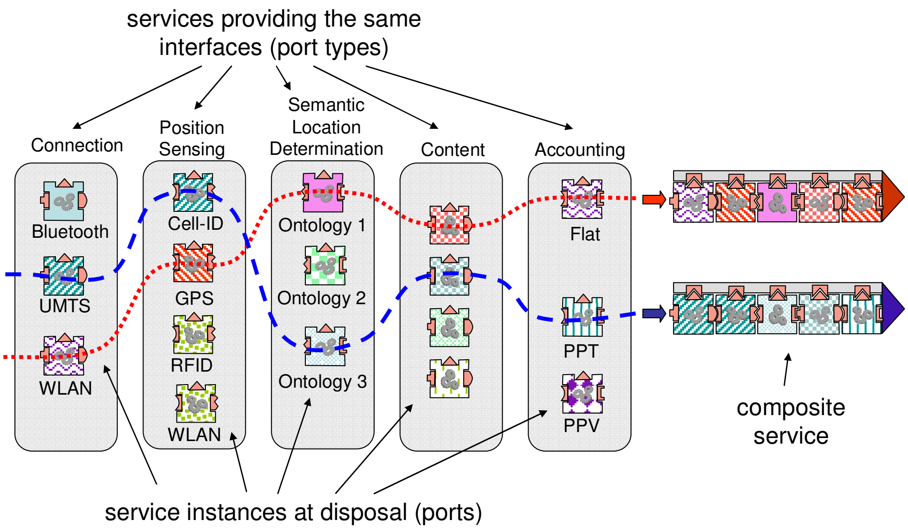

Fig. 3. Composition Process

The composition process is triggered at service invocation. Whenever an ALBS is invoked, it is dynamically composed of suitable ports. Among the ports of each port type, the best match will be taken with respect to the specific context that determines availability and suitability of each port. Therefore, the Web Services Policy Framework defines general purpose mechanism for associating policy expressions with subjects. Thus, specific property assertions can easily be attached to ports and registered in the repository.

For successful composition, at least one port of each required port type has to be accessible. Of course, subsequent composition processes will not take place if no relevant context change occurred. Context tracking, event filtering, and event notification combined with prediction techniques might further enhance the composition process and reduce latency time.

In the example presented in Fig. 3 there are five port types:

- **Connection**: Services of this type allow for access to available networks. This could be WLAN, LAN, Bluetooth, GSM, GPRS, or UMTS connections. Properties attached to these ports comprise information about bandwidth, costs, power consumption, encryption, or latency time.
- **Position Sensing**: This port type provides location information. Ports can be GPS-receivers, services based on stationary RFID-tags, or Bluetooth transmitters. Other services, e.g., based on WLAN-positioning or Cell-ID in cellular networks, are possible candidates as well. The properties should contain information about the accuracy of the location information. Extensions of position sensing services might be able to recognize direction, speed, and variance of movement. (WLAN and Bluetooth positioning base on signal strengths of different access points. For each position, these signal strengths exhibit a certain characteristics that can be translated into location information by a signal characteristics map. Since the signal strengths vary, the map needs periodic update. Nevertheless, coverage and accuracy of the positioning may be insufficient for some LBS. However, this way PDAs, laptops, or other mobile devices can locate themselves independently of GPS availability.)
- **Semantic location determination**: Information about location semantics is offered by this port type. Input is the context-specific sensor data (containing geographic location, RFID numbers, available services, or other specific characteristics that could be utilized to reason about the position). The response includes the semantic position according to a given ontology.
- **Content**: This port type offers content for a given geographic or semantic location. It receives a message with the location information which then is processed. The returned message contains information (text, pictures, audio, or video if requested) about the given location. To process the location information semantically, some common ontology is required (see Section 4.4. for further investigations on location semantics).
- **Accounting**: Accounting port type allows on-demand billing of services used.

4.2 Run-time Adaptation

The sequence chart (see Fig. 4) shows the message sequence of service interaction. The setup in this example consists of a service instance supervising the application control flow, the registry, e.g., an UDDI-implementation, two ports connecting to position sensing services (a GPS service and a WLAN positioning service), and two content ports. The example indicates how the composite service remains viable if some of its ports (here, the GPS positioning service) become temporarily unavailable, and how on-the-fly switchover to a replacement service (here, a WLAN positioning service[*]) takes place.

---

[*] Here, a connection switchover, e.g., from UMTS to WLAN connection, will probably occur. However, this is processed analogously and is not addressed in the Figure.

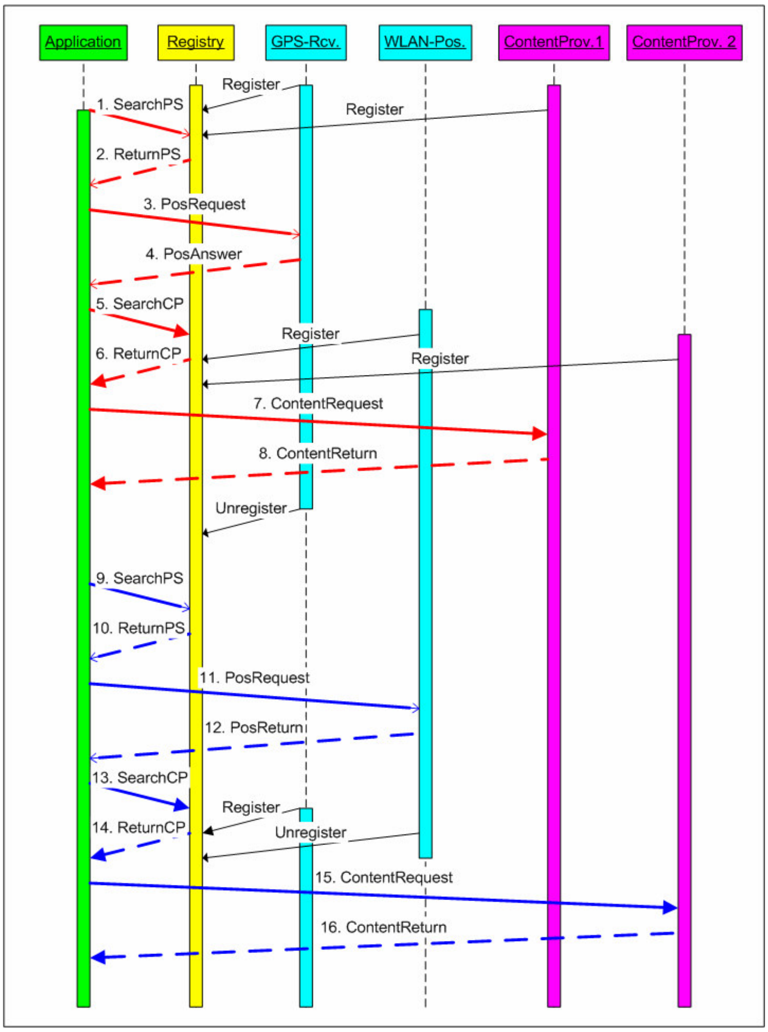

Fig. 4. Sequence chart of service interaction

The first sequence (messages 1-8) shows a service request using the available GPS service:

1. Search the registry for position sensing ports
2. Registry returns a GPS-receiver service port
3. Request position from returned port
4. GPS-receiver returns position
5. Search the registry for content port
6. Registry returns port of Content Provider 1
7. Request content from returned port
8. Content Provider 1 returns content data, e.g., a city map in which the building is located

Before message 9 is being sent, possibly the mobile user is entering a building, where the GPS device cannot receive the satellite signal and therefore unregisters its service from the registry. Supposing an in-house WLAN positioning service becomes available, the second sequence (9-16) shows the service request after this context change:

9. Search the registry for position sensing port
10. Registry returns port of WLAN-positioning service
11. Request position from returned port
12. WLAN-positioning service returns position
13. Search the registry for content provider port
14. Registry returns port of Content Provider 1 and Content Provider 2
15. Supposing semantic information is available that indicates the user is inside the building, Content Provider 2 providing corresponding content will be prioritized and requested
16. Content Provider 2 returns content data, e.g., a map that provides location-based guidance inside the building

As the sequence chart indicates, adaptivity results from context-sensitive service composition. Thereby, the messaging behavior of each subservice remains independent of context changes. This is possible because ports of the same port type can be interchangeably replaced without interfering with the ports' WSDL-prescribed request/response-behavior.

Traditional monolithic LBS typically do not provide this degree of context adaptivity (here, to switch to WLAN positioning in case the GPS becomes unavailable) without being explicitly designed for every possible change of interoperation. Furthermore, they hardly adapt to emerging technologies that were not foreseeable at design time. In contrast – provided that messaging behavior of new services remains compatible with the given type definition – ALBS can adapt to changing or newly emerging conditions without extra programming effort.

4.3 Using ALBS in Mobile Environments

The fundamental concept of the service-oriented paradigm is to enable uniform access to all resources via services – including mobile or embedded devices, and hardware resources, for example, GPS receivers or WLAN adapters. In dynamic environments, where network topology, connections, and bandwidth are unstable and connected devices may have limited resource power, this requires specific methods for service propagation, discovery, invocation, and processing:

- **WS-Discovery**: The Web Services Dynamic Discovery (WS-Discovery) standard defines a multicast protocol to propagate and locate services on ad-hoc networks in peer-to-peer manner. It supports announcement of both service offers *and* service requests. Efficient algorithms (caching, multicast listening, discovery proxies, message forwarding, filtering, scope adjustment, and multicast suppression) keep network traffic for announcing and probing manageable. Thus, the protocol scales to a large number of endpoints.

- **Lightweight Services**: For efficient invocation and processing of Web Services on embedded devices with limited processing and communication power, "light-weight" services utilize specific real-time protocols, programming languages, scheduling algorithms, message queuing policies, or XML coding and parsing schemes.

In our example, the mobile device multicasts its request to the devices within local reach and collects the service announcements. The GPS receiver announces a service for position sensing and the WLAN adapter announces two services, one for position sensing and one for connection (see Fig.3). These ports are stored in the local registry cache. Retrieved entries from the global registry are cached as well. To locate a service, the discovery service is instructed to retrieve the corresponding port type. Additionally, the discovery service can look for certain assertions to be satisfied. Thus, the ALBS application communicates with local services the same way it does with remote services. All services are propagated, discovered, and invoked by standard Web Services protocols.

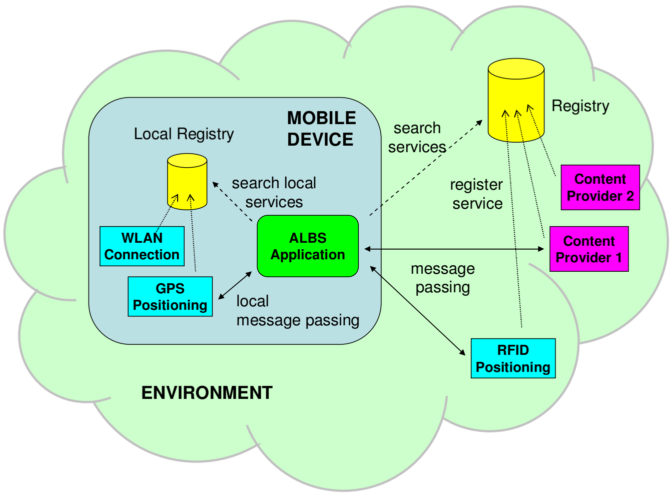

Fig. 5. The ALBS architecture allows for location-aware computing based on universal service-oriented communication

4.4 Semantic Location Information

For semantic interpretation we distinguish the following LBS classes: Location-based services can be provided by some immobile unit, e.g., a museum or a botanical garden. Typically such immobile units provide *stationary LBS* which are fixed to a given location. A common problem is to semantically detect the location and find or filter stationary services related to that location. For example, a user's movement in a museum can tell that he might be interested in information about a specific exhibition object (e.g., he moves to that object and then, while looking at it, stops moving for some seconds). A location-aware device then could request the assigned service. Likewise, some immobile units may provide *general LBS* that are location-independently accessible but require a location parameter. Examples are a regional weather forecasting service or a service that processes queries like "where is the next subway station?" Regarding *mobile LBS*, the location is a parameter of the behavior of a mobile device. Imagine a user who travels with his laptop. If the laptop recognizes the availability of a specific LAN connection, it could conclude where it is located (e.g., in the user's office) and adapt its behavior (e.g., synchronize certain files). Finally, *interdependent LBS* require multiple related location parameters, e.g., a people finding service that guides mobile users to meet at some intermediate place. All these cases demand for appropriate semantic interpretation of location.

Let us consider the following example of a mobile LBS in more detail: A user wants his mobile phone to automatically activate the hands-free speaking system inside a car or mute when inside a theatre. However, a cellular phone cannot tell from GPS coordinates or from its cell ID that it is inside a theatre. But if there is a service that translates the GPS coordinates or the cell ID to semantic location information like "this is a theatre" or "this is a place to be silent", the "mute feature" can be automated. The Resource Description Framework (RDF) addresses these issues and may be used to accomplish such communication.

For example, an extended position sensing service may return the semantic location "prater.theatres.berlin.de". To know how to act on this location information, a service assigned to it might return the following RDF message indicating that mobile phones and other possibly "obtrusive" devices should be switched off. Since the theatre is an "ambient" place in the following ontology, the device can understand that it should mute:

```
<rdf:Description about="urn://prater.theatres.berlin.de">
<rdf:type resource="urn://myontology.myID.de/Schema/theatre"/>
<rdf:type resource="urn://myontology.myID.de/Schema/places/ambient"/>
<t:Name>Prater</t:Name>
<t:DesiredCellPhoneActivity>Silent</t:DesiredCellPhoneActivity>
</rdf:Description>
```

Unfortunately, there are various ways to express such additional semantic location information. The device, therefore, not only must be able to access the ontology that is applicable, it moreover needs to know how to map this ontology to its decision alternatives. However, to the best of our knowledge, comprehensive ontologies widely accepted and suitable for broad semantic location processing are not available as yet.

5 Case Study

Currently, we are working on an adaptive location-based service prototype. It provides the basic functionality of a mobile information system for the "WISTA Adlershof Science and Technology Park" [21] in Berlin, where natural science departments of the Humboldt University are located as well. Based on the Emerging Technologies

Toolkit (ETTK) available from IBM developerWorks, we provide part of the ALBS infrastructure. We are defining some basic port types (Content and Position Sensing) allowing all companies from the WISTA area to provide further location-based information and services. Additionally, a web crawler will scan the regional websites for location information and, if location can be determined, assigns the website to that location. Geographic information and a geo-referenced map are supplied by the Geographic Institute of Humboldt University, which takes part in this case study. A widely available WLAN-infrastructure gives mobile devices access to remote services and allows the setup of WLAN-based position sensing services.

Location-based service will also be available for "virtual travelers". They explore the WISTA map on the Internet that visualizes location-specific information and stationary services. By point-and-click, it is possible to directly access these stationary LBS or to forward the specific position to some general LBS. That way LBS link virtual to physical spaces. For example, if a user is visiting the Internet site of a company, the company's physical location will be determined and can serve as input for subsequent LBS. Vice versa there are LBS that link from physical to virtual spaces, e.g., one that processes instructions such as "show me the website of the restaurants located within 5 minutes walk". In future, a position sensing service can as well determine the semantic position within the virtual space. For example, if position sensing detects that the user is visiting some product information site, it can take him to product-related offers, e.g., test reports or best price comparisons.

6 Outlook and Conclusions

We have shown how to flexibly compose Adaptive Location-Based Services (ALBS) with Web Services technology achieving high service availability. Further, we outlined how location information can be processed semantically. We anticipate that the methodology will be applicable to future context-aware computing in distributed, heterogeneous, and dynamic environments at great degree of interoperability – across various protocols, interfaces, programming languages, devices, connection lines, operation systems, platforms, enterprise boundaries, vendors, and service providers.

We also expect that broad interconnection and seamless interoperability of processes and devices together with tight interconnection of physical and digital spaces will bring convergence of virtuality and reality, changing the way we think, work, and live. As public dependence on information systems will continue to rise and there will be insufficient human resources to continually support and maintain the computing/communication infrastructure, the vision of adaptive, maintenance-free, self-relying systems must become a reality. The goal will be to deliver MAD properties (Mobility, Adaptivity, and Dependability) at low cost and in intelligent and highly semantic manner, making it possible to enter the age of "subdued computing", where the information infrastructure at best supports human computing and communication needs, but discreetly fades into the background.

7 References

1. Auto-ID Center: Creating an Internet of Things. www.epcglobalinc.org (accessed April 2004).
2. BPEL4WS, Business Process Execution Language for Web Services, Version 1.1, www-106.ibm.com/developerworks/library/ws-bpel
3. Buyya, R.: Economic-based Distributed Resource Management and Scheduling for Grid Computing. Ph.D Thesis, Monash University, Melbourne, Australia, April 2002
4. European Symposium on Ambient Intelligence (EUSAI), www.eusai.net (accessed April 2004)
5. Hazas, M., Scott, J., Krumm, J.: Location-Aware Computing Comes of Age. IEEE Computer, 37(2):95–97, Feb 2004
6. Hawaii International Conference on System Sciences, Value Webs in the Digital Economy Mini-Track, Hawaii, Jan 2005, www.value-webs.org
7. Jennings N., Sycara, K. et al.: A Roadmap of Research and Development. Autonomous Agents and Multi-Agent Systems, 1(1):7–38, Jan 1998
8. Malek, M.: The NOMADS Republic. International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine and Mobile Technologies on the Internet, Scuola Superiore G. Reiss Romoli (SSGRR), Telecom Italia, L'Aquila, Italy, 2003
9. Kephart, J., Chess, D.: The Vision of Autonomic Computing. IEEE Computer, 36(1): 41-50, Jan 2003
10. Meissen, U., Pfennigschmidt, St., Voisard, A., Wahnfried, T.: Context- and Situation-Awareness in Information Logistics. International Conference on Extending Database Technology (EDBT), Workshop on Pervasive Information Management, Heraklion, Greece, March 2004
11. Milanovic, N., Richling, J., Malek, M.: Lightweight Services for Embedded Systems. IEEE Workshop on Software Technologies for Embedded and Ubiquitous Computing Systems (WSTFEUS), Vienna, Austria, 2004
12. Müller, P., Stich, Ch., Zeidler, Ch.: Components @ work: Component Technology for Embedded Systems. 27th Euromicro Conference, Warsaw, Poland, Sep 2001.
13. Papazoglou, M.: Agent-oriented Technology in Support of E-Business: Enabling the Development of "Intelligent" Business Agents for Adaptive, Reusable Software. Communications of the ACM, 44(4):71–77, April 2001
14. Rao, B., Minakakis, L.: Evolution of Mobile Location-based Services. Communications of the ACM, 46(12):61-65, Dec 2003
15. Reichenbacher, T. : Mobile Cartography – Adaptive Visualisation of Geographic Information on Mobile Devices. Dissertation submitted at the Institute of Photogrammetry und Cartography, Technical University, Munich, 2004
16. Schwan, K., Poellabauer, Ch., Eisenhauer, G., Pande, S., Pu, C.: Infofabric: Adaptive Services in Distributed Embedded Systems. IEEE Workshop on Large Scale Real-Time and Embedded Systems (in conjunction with RTSS 2002), Austin, TX, Dec 2002
17. Sutherland, J., van den Heuvel, W.-J.: Enterprise Application Integration and Complex Adaptive Systems. Communications of the ACM, 45(10):59–64, Oct 2002
18. Szyperski, C.: Component Software: Beyond Object-Oriented Programming. Addison-Wesley, 2003
19. UDDI, Universal Description, Discovery and Integration of Web Services, www.uddi.org
20. Weiser, M.: Some Computer Science Problems in Ubiquitous Computing. Communications of the ACM, 36(7):74-83, July 1993
21. WISTA Adlershof Science and Technology Park, www.wista.de