Towards a Global Spatial Data Infrastructure Using Web Services

Peter IBACH and Matthias HORBANK, Germany

Key words: Location-based Services, Web Services, Spatial Data Infrastructure

SUMMARY

Location-based Services (LBS) are services that utilize their ability of location-awareness to simplify user interactions and adapt to the location-specific context. Unfortunately, current LBS applications do not provide the degree of adaptivity and interoperability required for mobile environments in a global scale. Therefore, a global spatial data infrastructure (GSDI) is mandatory that is flexible enough and accessible via open, interoperable standards for data formats, interfaces, and protocols.

In this paper, we investigate how to setup a GSDI - we call our approach the *Sema*ntic *Lo*cation *N*etwork (*SEMALON*) - based on open Web Service standards and we describe how Web Service Description can be enriched with location information to allow for service discovery with respect to spatial criteria and how LBS applications can be dynamically composed during runtime. Primary focus is put on smooth integration, i.e., without the need for changing the standards.

Another major issue in SEMALON is location semantics. Currently location information is mostly represented by geographic coordinates, i.e., values describing latitude, longitude, and altitude in some coordinate systems. Unfortunately, there are many cases where geo-coordinates are not sufficient and more meaningful location descriptions are demanded, e.g., the building or room where a user is located. In SEMALON, ontologies can be used to define objects and relations between objects. However, it will be difficult to get any ontology globally accepted and finally, various ontologies will be required to suit the multiplicity of applications. Thus, multiple application-specific ontologies can coexist in our approach while ontology translation allows for adaptive service composition and semantic interoperability.

We implemented an example chain of Web Services comprising position sensing, semantic location determination, content delivery, and accounting. Different or new services can be easily integrated according to environmental dynamics. This achieves a high degree of adaptivity. Our prototype implementation for the Science and Technology Park Berlin-Adlershof allows for dynamic switching from GPS positioning to in-house WLAN positioning. It also integrates support for stationary or mobile objects which may provide further descriptive content (text, pictures, graphics, and videos). Because of its fully distributed architecture, its open Web Services foundation, its adaptivity, and its multi ontology support, our SEMALON approach seams promising as a basic GSDI and gives valuable insight for further developments.

Towards a Global Spatial Data Infrastructure Using Web Services

Peter IBACH and Matthias HORBANK, Germany

1. INTRODUCTION

As part of NOMADS (Malek2003, Malek2004) which is a project for building a comprehensive service-oriented framework for ubiquitous computing, we are developing the *Sema*ntic *Lo*cation *N*etwork (*SEMALON*) based on open Web Services standards. In SEMALON all resources are uniformly accessed via services. Services expose interfaces that can be semantically interpreted using ontologies. Multiple layers contain objects, their locations, and associated services (see Fig. 1). Physical resources might be stationary, such as streets and buildings, or mobile, such as mobile devices or RFID-tagged items. Informational resources comprise Internet pages**,** objects in databases**,** and services.

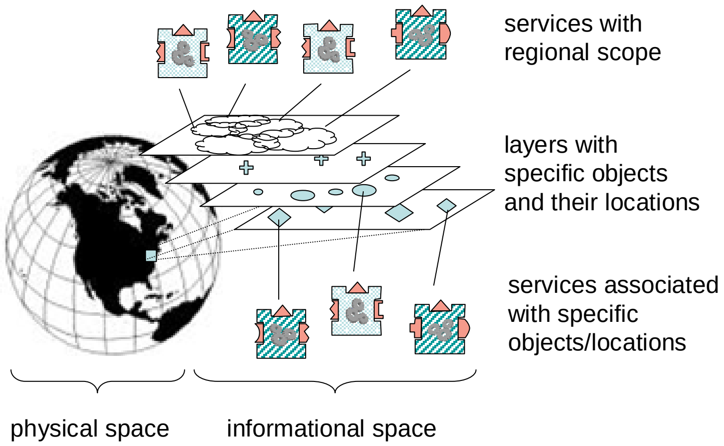


Fig. 1: SEMALON – a globally scalable semantic network of Location-based Services

A number of approaches are trying to connect physical objects and correlated Web resources (e.g., the Geoweb (Goochild2002); Geotags: www.geotags.com; GeoURL: www.geourl.org) or propose protocols to exchange location information over the Internet (e.g., the Mobile Positioning Protocol from Sony Ericsson or the Spatial Location Protocol of the IETF). Still, these approaches do not jointly consider necessary requirements such as:

- Integration of both information access as well as activity invocation
- Versatile protocols supporting mobility, adaptivity, and dependability
- Global interoperability, discoverability, and scalability
- Interface to human users, business processes, and embedded devices
- Application-specific location semantics

In SEMALON we are pursuing these issues by employing open Web Services standards. Their use for LBS has its appealing strengths (Ibach2004/1, Pinto2003), and particularly the Open Geospatial Consortium (www.opengeospatial.org) is focusing on Web Services standards for publishing, finding, and binding geospatial services. However, location-based discovery and composition were not originally provisioned in these standards and the integral Web Services support for mobility and ad-hoc adaptivity to changing conditions is still under development. This gives rise to a number of difficulties. In particular, Web Service discovery processes do not support application-specific customization that incorporates a distance measure or other ranking functions.

In this paper we describe how location information can be placed in the Web Service repository to allow for service discovery with respect to spatial criteria. Primary focus is put on smooth integration, i.e., without the need for changing the Web Services standards. To overcome limitations of predominant location description with bare static geo-coordinates, modeling location information semantically using spatial ontologies is supported. Multiple application-specific ontologies can coexist in our approach, while ontology translation allows for adaptive service composition and semantic interoperability in a global scale.

2. LOCATION BASED SERVICES - CURRENT APPLICATIONS AND ARCHITECTURES

Location-based Services have been hyped as the “killer app” during the Internet bubble – whereas true market developments could not accomplish the exaggerated expectations. But with the advances of mobile devices, position sensing, and wireless connectivity, the market for Location-based Services is rapidly developing, particularly in the field of geographic, telematic, touristic, and logistic information systems.

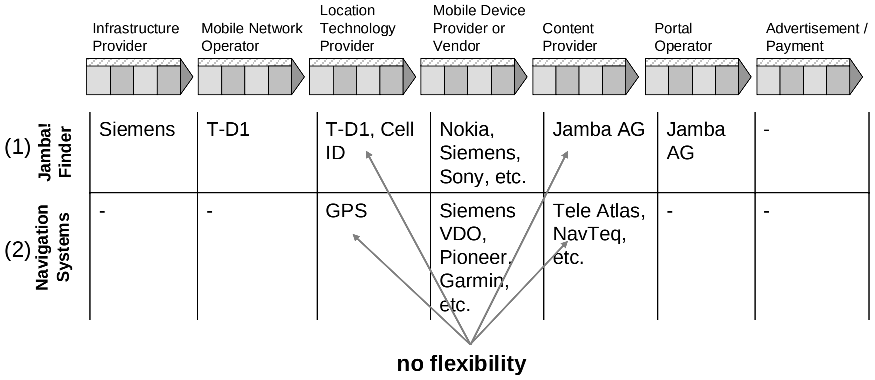
Wireless emergency services require the ability to pinpoint the location of a cell phone placing an emergency call, e.g., for fire brigade, ambulance, or police. E911 Phase II legislation in the US requires cell phone companies to be able to locate handsets within 150 meters by 2006. E112 initiatives in Europe are similar.

Positioning techniques now are maturing to provide accurate positioning in outdoor *and* indoor environments at affordable cost, small size, and low power consumption. Hamerhead, for example, is a single chip assisted GPS solution at €6.50 and sufficiently sensitive that it works in most indoor environments (Infineon2004). Infineon expects a market of more than 700 million mobile phones to be sold 2008 where 25% of those will be equipped with A-GPS functionality.

Commodity mobile devices, such as laptops, PDAs, and cell phones can sense their position even without extra GPS receivers. Intel’s PlaceLab project therefore has mapped the positions of millions of existing GSM, WLAN, or Bluetooth base stations allover the world. Their experiments in the greater Seattle area indicate 20 to 40 meter median accuracy and close to 100% coverage exploiting “radio beacons in the wild” (Schilit2003). These positioning techniques may bootstrap the broad adoption of location-aware computing.

However, there are a lot of problems to be solved before LBS markets will leap off and release their enormous economic potential.

Although there are numerous proprietary applications that deal very well with location information, interoperability of location information across application boundaries in a standardized open format over the Internet is still unaccomplished. Considering location semantics and mobility, the situation is even worse.



	Infrastructure Provider	Mobile Network Operator	Location Technology Provider	Mobile Device Provider or Vendor	Content Provider	Portal Operator	Advertisement / Payment
(1) Jamba! Finder	Siemens	T-D1	T-D1, Cell ID	Nokia, Siemens, Sony, etc.	Jamba AG	Jamba AG	-
(2) Navigation Systems	-	-	GPS	Siemens VDO, Pioneer, Garmin, etc.	Tele Atlas, NavTeq, etc.	-	-

Fig. 2: Examples of current LBS value chains with participating service providers. Multiple barriers hamper flexible service composition

Present LBS are mostly bound to a specific technology reflecting the preferences of the service provider. Fig. 2 shows two exemplary applications popular in the German LBS market: (1) The Jamba Finder allows cell phone users to look for objects nearby, e.g., public buildings, fuel stations, cinemas, or restaurants. (2) Navigation systems as deployed in cars or independently usable via GPS equipped PDAs enjoy rapidly growing popularity.

Typically, proprietary protocols and interfaces are employed in these LBS applications 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 new conditions, it very likely requires entire reengineering.

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 emerging. Thus, hardware devices and software components, their interfaces and architecture, have to be able to deal with changing conditions. Thus, adaptivity - the ability to cope with incessantly changing conditions - is crucial to make mobile Location-based Services highly available and overall successful.

Lots of research has focused on Location-based Services combining the concept of location-aware computing with distributed geographic information services based on Internet standards, see (Hazas2004, Hodes2003, Peng2004, Rao2003, Reichenbacher2004). Unfortunately, a number of specific interoperability barriers exist in current LBS value chains, resulting in the "Multi-X Problem":

- Multiple connection technologies (GSM, UMTS, WLAN, Bluetooth, …)
- Multiple location technologies (GPS, Cell-ID, WLAN, Bluetooth, RFID, …)
- Multiple hardware, software, and service providers
- Multiple operating systems, programming languages, and system architectures
- Multiple application-specific ontologies describing location semantics
- Multiple content depending on specific location and granularity demands

Flexible service composition requires interoperability despite of increasing multiplicity. Web Services standards seem promising to solve this challenging problem.

3. WHY WEB SERVICES ?

In service-oriented computing, resources are accessed via services. Services expose well specified interfaces and are the basic building blocks for flexible and efficient composition of more complex applications. Fundamental concept is the composition of systems by extensive reuse of commodity software/hardware components. Many approaches share this very general concept of compositionality.

However, a number of differences - e.g., in wording, perception, implementation, and practical use - are indicating advantages of the service-oriented paradigm over previous approaches that were focusing on components, objects, modules, or other compositional entities. At the forefront, Web Services and Grid technologies are attracting a lot of attention accompanied by mixed opinions whether the expectations in reusability, composability, flexibility, maintainability, and return on investment that previous approaches have struggled with can finally be accomplished. See, for example, (Linthicum2003) for the growing importance of Web Services in Enterprise Application Integration and (Bloomberg2002, Gokhale2002) for a detailed discussion of pros and cons comparing Web Services to preceding concepts like CORBA.

Web Services and Grid toolkits like the Globus Toolkit or the Emerging Technology Toolkit have helped establishing standards. Component-based software for embedded systems (Müller2001) and lightweight services (Milanovic2004, Schwan2002) 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 available resources. Driving the widespread acceptance of the service-oriented paradigm, Location-based Services are challenging numerous new applications and business opportunities and might reveal the enormous economic potential of

Dynamic Value Webs: the on-demand aggregation of services even across enterprise boundaries.

Furthermore, as the “Internet of things” with billions and soon trillions of seamlessly interconnected devices is about to take over, we expect for the next years a literally exploding number of services that not only provide information about physical objects, originating from Web pages, database entries, or sensors, but also allow to trigger activities and control the objects by some actuators. Through the spatial organization of physical and informational objects, virtual and real spaces will tightly interconnect (Andersen2002).

4. ADAPTIVE SERVICE COMPOSITION

While Web Services and Grid standards appear promising to overcome a number of interoperability hurdles, they have not been designed for some specific LBS demands. Unlike in enterprise or desktop computing, mobile users tend to strongly interact with their environments. To support processes in the physical world by information technology a location-based mapping that connects physical objects and their correlated information is required. This spatial interrelationship is what will put mobile users in the position to navigate through the growing complexity and dynamics of physical and informational spaces.

From the perspective of a mobile user, the environment is ever-changing as he moves from one location to another. Available resources, demanded services, as well as achievable quality of service levels are incessantly changing, and adaptivity to changing conditions becomes crucial.

A general idea to cope with high complexity and dynamics is to design and structure composite systems such that they are able to meet upcoming challenges by actions of their largely autonomous entities. Under the term “autonomic computing,” for example, IBM summarizes eight core “elements” (Kephart2003) – comprising self-configuring, self-healing, self-optimizing, and self-protecting – that are intended to guide the development.

Similar approaches consider software agents (also referred to as actors) 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 (Jennings1998, Papazoglou2001) and further extensions for business environments referred to as business agents or agentified enterprise components (Sutherland2002). 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.

In our approach, adaptivity of composite Location-based Services – 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 elementary services through standardized communication protocols and directory services.

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) over the Hypertext Transfer Protocol (HTTP) might be a primary candidate, but other protocols 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. The composition process can be graphically expressed by a path through a network of accessible ports:

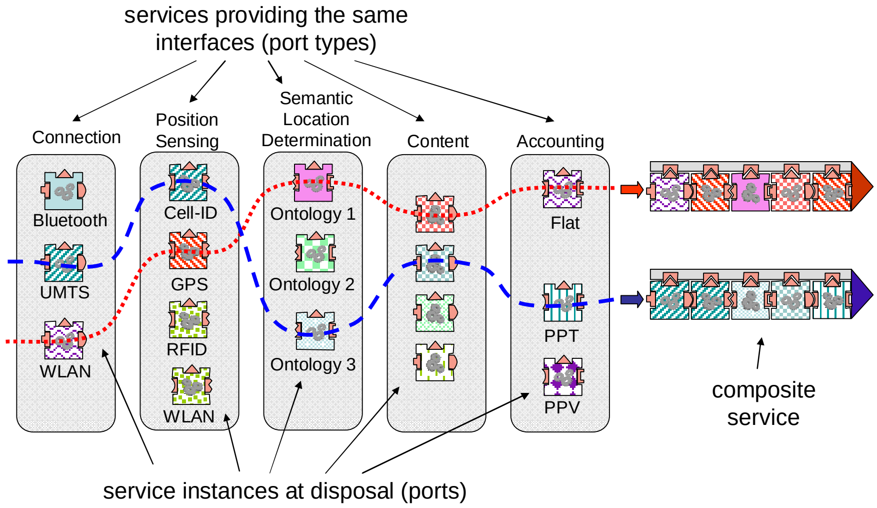


Fig: 3. Building adaptive composite services by on-demand service composition

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. For successful composition, at least one port of each required port type has to be accessible. For detailed description of parameters, service interaction, and ALBS architecture see (Ibach2004/1).

In our ALBS approach, 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 without being explicitly designed for every possible change of interoperation. Furthermore, they hardly adapt to emerging technologies, specific user profiles, or application demands 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.

On the global scale, however, the number of available services of a certain type will quickly get out of hand. Thus, selection of services cannot be done solely at the client side; instead, the discovery process has to restrict the query result by location-based scope and/or a ranking of the resulting services incorporating some distance measure.

5. LOCATION-BASED SERVICE DISCOVERY

Location-based service discovery is a prerequisite to arrive at Adaptive Location-based Services. Thereby, the client must be able to control the volume of query results according to their relevance for its actual context and take the available network and processing capacity into consideration as well.

For example, a user might be looking for a city map of Berlin. Then, he does not want to get an unrestricted number of city map providers – possibly thousands all over the world – but only those of relevance for the specified location. An even more fine-granular search could request a map focusing on Potsdamer Platz located in Berlin's city center, or, regarding future LBS applications, an inhouse map of a particular building.

5.1 Scope-based Discovery using Spatial Ontologies

The Universal Description, Discovery and Integration (UDDI) specification defines how to describe, publish, and discover Web Services using a registry. To discover a service, the client at first has to decide which registry to query. Typically, a default and a backup registry server will be preconfigured at client's site. The discovery service is itself a Web Service and instances can be registered in a repository. Thus, if required, the client may query a known registry for other available registry instances. Using some categorization scheme that defines an overlay-map, the registries can be organized, e.g., hierarchically as the Distributed Name Service (DNS), or according to other application-specific demands.

Using policy assertions, descriptive annotations can be attached to any service type or service instance and registered in a repository. To allow Web Service providers and their requestors to interpret these attachments semantically, ontologies, i.e., standardized, commonly available, and machine-interpretable categorization schemes, must be provided.

When querying a registry for a specific service, the client can search for values that describe the entry according to a categorization scheme. The category levels can be combined using Boolean operators, e.g., lookup services where <City> is <Berlin> *and* <Building> is <Sony Tower>. This can be combined with various string operations, e.g., querying for substrings or using string concatenation.

Microsoft, for example, has introduced a spatial categorization scheme "microsoft-com:geoweb:2000". This is a first step to attach location information to Web Services, but it does not allow fine-granular or application-specific location description. If searching is limited to this ontology, in our example, Berlin would be the finest available granularity. This might fit for some applications, but does not fulfil general needs. Section 0 therefore describes different spatial ontologies representing application-specific objects and their relationships.

5.2 Multi-Ontology Support using an Ontology Translation Service

One way to cope with multiple ontologies would be to register each service with multiple policy assertions representing the location information, each of which according to a different spatial ontology possibly being used in subsequent discovery requests. This would require service providers to constantly check for new ontologies and update their entries in the registry accordingly. Obviously, this is not a suitable approach.

The opposite way is to register for every entry only a single policy assertion representing the spatial information according to one specific ontology. Then, each client would have to know the ontologies being used and how to express a given position in these ontologies to query for corresponding services. Again, this approach is hardly feasible since clients cannot be expected to have the required knowledge.

Therefore, we suggest allowing service entries to have spatial policy assertions according to any ontology of their choice, as long as information about the used ontology and means for translating it into other ontologies are commonly available. For the common translation we propose an *Ontology Translation Service* (OTS). Distributed instances of this service should be accessible for any client and translate a location given in one ontology into one or more target ontologies (see Fig. 4 and Section 0).

Thus, clients can discover services based on location information according to various ontologies. They can query the registry even if they do not know anything about the other ontologies, as long as at least one available OTS instance supports appropriate translation from the source to the target ontology.

5.3 Customizable Discovery using a Dedicated Service

Still, the described scope-based discovery has a number of disadvantages:

- Very likely, the client has to probe a granularity level and, if the expected number or quality of results is not achieved, has to redo the lookup at a different granularity level.
- It allows only querying for exact matches within the registry. In scope-based queries, users look for services within a specified scope, e.g., a particular country, region, city, or even a particular building or room. Services that are within this scope are typically deemed to have the same relevance while services outside this scope are not considered relevant. However, some Location-based Services require a ranking of the resulting services incorporating some distance measure.

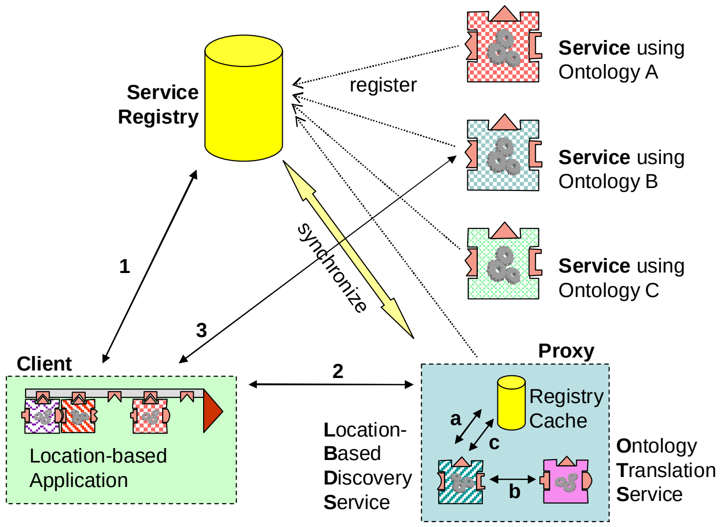


Fig. 4: Using an Ontology Translation Service (OTS) to cope with multiple ontologies and a Location-based Discovery Service (LBDS) for customizable search functions

Imagine, for example, a tourist visiting Potsdamer Platz in Berlin. Using his GPS equipped PDA, he might pose the query: “Find all content providers offering points of interest information within the range of 500 meters of geo-coordinate N52°30'54" E13°22'59".” This query should return information about associated points of interest – and of course, the closer they are the more likely they are of interest to the user. Thus, the search should be customizable by some spatial relation. For many cases, it will be sufficient to express the spatial relation in meters describing the radial distance to a central point. But more expressive spatial relationships are desirable. Therefore, further distance measures and object topologies are described in Section 0.

To solve the above problems we suggest a *Location-based Discovery Service* (LBDS) offering customizable search functions for location-based applications. To speed up access to registry entries, we propose to cache recent lookups in a proxy. The proxy may host instances of the OTS and LBDS (see Fig. 4) allowing for fast conversion between ontologies without causing additional network traffic. The client processes the following steps:

- *Lookup LBDS***:** query registry for LBDS,
- *Lookup service*: employ the LBDS to search for services regarding a given position and distance measure,
- *Select and call service***:** choose adequate service out of the returned list due to specific application requirements and use selected service

Whereas, LBDS and OTS take care for customizable search functions dealing with objects’ spatial relationship and multiple ontologies:

- *Lookup ontologies***:** request information about ontologies used by services of the requested port type,
- *Translate ontologies***:** transform given location information from source ontology into identified target ontologies,
- *Lookup service***:** search for service instances regarding the requested position and distance measure (i.e., hand over location information in all target ontologies).

This solution allows clients to easily lookup desired services according to spatial information. They do not need to bother about troublesome ontology translation or complex distance-based searches.

Obviously, if LBDS and OTS do not know about a requested search function or ontology, they cannot process the queries and the request has to be processed by another LBDS/OTS. Thus, to reduce network traffic, they should, to a great extent, cover the specific functions the clients in their vicinity want to use.

In mobile environments, connections and available services are subjected to changes; in particular, it cannot be assured that mobile clients have continuous access to a registry. Therefore, the Web Services Dynamic Discovery (WS-Discovery) standard defines protocols to propagate and locate services on ad-hoc networks in peer-to-peer manner. It supports announcement of both service offers *and* service requests and provides efficient algorithms (caching, multicast listening, discovery proxies, message forwarding, filtering, scope adjustment, and multicast suppression) to keep network traffic for announcing and probing manageable.

Under specific network and connectivity circumstances, different policies for service aggregation might be beneficial; at highly available connection lines, for example, service aggregation by dedicated intermediate parties appears to be advantageous, whereas in mobile environments with unstable connections, we expect client-centric aggregation with nearby caching and service execution to be preferred.

6. LOCATION SEMANTICS

For semantic location determination 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 certain 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 could then request the appropriate 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 describing the context of a mobile device. Imagine a user traveling with his laptop: If the laptop recognized 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. To accomplish semantic interoperability, one has to agree on suitable ontologies which define objects and relations for each specific application area.

6.1 Spatial Ontologies

Typically, locations are represented by geographic coordinates. In common use is the projection according to the World Geographic System 1984 (WGS84), other projections comprise Universal Transverse Mercator, Swissgrid, Gauss-Krüger-Grid, or the Military Grid Reference System, which can be interchangeably converted by software algorithms.

Going beyond bare geo-coordinates or free-form textual descriptions, spatial ontologies can be used to define objects and relations by means of spatial semantics. A widely accepted ontology that models physical objects and their location is the Geography Markup Language (GML), standardized by the OpenGIS Consortium and used in the Geographic Information System (GIS). 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. The World Wide Web Consortium is extending the Resource Definition Framework (RDF) to relate Web Content to its associated physical location. The DARPA Agent Markup Language (DAML) combines multiple schemata for location description.

Using GML, DAML, or RDF, complex schemata can be designed. Related elements can be grouped and hierarchically structured to represent different aspects of location information. The following GML example describes the Sony Tower at Potsdamer Platz in Berlin, comprising address, surface area, and geo-coordinates:

```
<exp:Building fid = “Sony Tower”>                               ← name of the building
    <exp:noFloors>26</exp:noFloors>                             ← number of floors
    <exp:use>Commercial</exp:use>                               ← commercial type of use
    <exp:surfaceArea>216700</exp:surfaceArea>                   ← surface area in m2
    <exp:frontsOn>Neue Potsdamer Straße </exp:frontsOn>         ← street
    <gml:locationOf>
            …                                                   ← geo-coordinates, in WGS84 standard
    </gml:locationOf>
</exp:Building>
```

Typically, discovery by means of spatial semantics is done describing some known objects and, based on those, query for other spatially related objects. In previous examples we have used radial distance to a central point.

Tab. 1 shows how this can be expanded regarding two- or three-dimensional objects, different prepositions, and custom distance measures.

Object	**Preposition**	**Distance**	**Use Case Example: look up …**
point	undirected	meters	… WLAN hot spots *within a radius of 50 meters*
	south	flying time	… holiday destinations *south* of your location
	undirected	walking time	… restaurants *within 5 minutes of walk*
polygon	undirected	driving time	… SOS-telephones *along* a highway
	west	driving time	… customs facilities *west* of the country's border
polyhedrons	undirected	boolean	… people *inside* a specific room
	undirected	boolean	… *adjacent* offices in a tower block
	below	walking time	… parking levels *below* the first floor

Tab. 1: Describing spatial relations between objects

6.2 Ontology Translation

As described, ontology translation is handled by dedicated Ontology Translation Service (OTS) instances. Inputs are: source ontology instances, including a reference to the source ontology, and references to one or multiple target ontologies. It outputs the information translated into the requested target ontologies. Typically, ontology translation is pursued in three steps:

- *Discovery:* manually, automatically, or semi-automatically defining the relations between ontologies
- *Representation:* A language to represent the relations between the ontologies
- *Execution:* Changing instances of the source ontology to instances of target ontology

One approach for ontology translation is to provide an explicit *m*-*n* mapping for any given pair of *m* source to *n* target ontologies. This potentially achieves maximum translation quality but the required number of mappings grows quadratically, i.e., at O($m \cdot n$). At the other extreme, the *m*-1-*n* translation introduces an intermediate ontology into which all source ontologies are translated and from which all target ontologies are derived (see Fig. 5). This minimizes the number of required mappings to linear growth, O($m+n$), but for many cases, it results in unacceptable loss of information. Therefore, we pursue a hybrid approach, were the best path over a manageable number of intermediate ontologies is chosen.

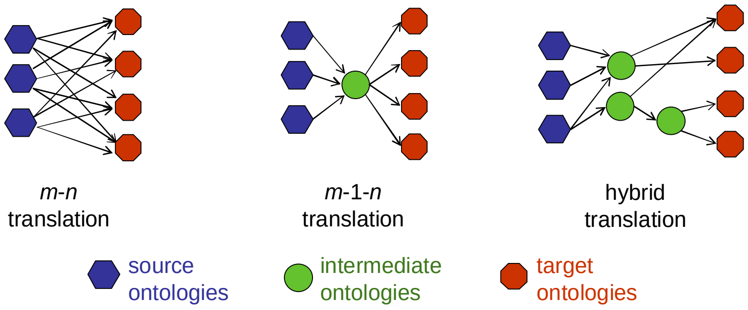


Fig. 5: Ontology translation as a hybrid of *m*-*n* and *m*-1-*n* mapping

If a suitable translation path can be found, the source information can be exploited, e.g., to trigger certain activities. Imagine, for example, a user who wants his mobile phone to automatically activate the hands-free speaking system inside a car or mute when inside a theater. Usually, a cellular phone cannot tell from geo-coordinates - determined, e.g., by a GPS signal or by its cell ID - that it is inside a theater. But if the location description contains information that allows deriving from given geo-coordinates that the location is a theater, belonging to the category <silent space>, the "mute feature" could be automated (see Fig. 6).

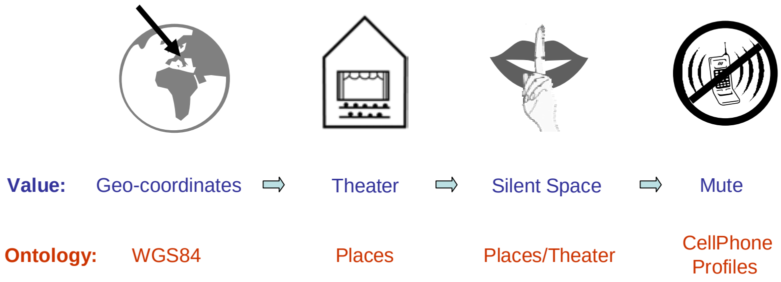


Fig. 6: Nested ontology translation from WGS84 geo-coordinates into the value <mute> in the <CellPhoneProfiles> ontology

Accordingly, the location <Prater>, a theater in Berlin, would need to indicate that it is a <theater> in the <places> scheme, which defines that a <theater> is a <silent space>:

```
<rdf:RDF          xmlns:rdf="…"  xmlns:plc="…"        ← XML namespaces
    xmlns:geo="…"                xmlns:scp="…">
    <plc:Places rdfID="…">                            ← place attributes
    <plc:category>theater</plc:category>              ← it's a theater
    </plc:Places>
    <scp:Scope rdfID="…">                             ← object's extent:
        <scp:type>undirected</scp:type>               ← radial distance
        <scp:distance>20</scp:distance>               ← of 20
        <scp:metric>m</scp:metric>                    ← meters to
    </scp:Scope>
    <geo:Point rdfID="…">                             ← the point with geo-coordinates:
        <geo:lat>52.539833</geo:lat>                  ← Latitude: N52° 32.39'
        <geo:long>13.410666</geo:long>                ← Longitude: E13° 24.64'
        <geo:alt>91.000</geo:alt>                     ← Altitude: 91 meters above sea level (WGS84 standard)
    </geo:Point>
</rdf:RDF>
```

Nested ontology translations of such kind are a major challenge of the Semantic Web.

7. CONCLUSIONS

We investigated the applicability of Web Services standards in the domain of mobile environments, in particular in the field of Location-based Services, to overcome the "Multi-X Problem" where multiple barriers block adaptive interoperability. In particular, we described how to discover Web Services due to spatial criteria, and how to flexibly compose location-specific elementary services along typical value chains using Web Services technology. The proposed methodology enables on-demand discovery and compositions of Web Services with respect to changing locations. It achieves high adaptivity at the composite service level and allows mobile services to adapt to incessantly changing environments.

Furthermore, we outlined how location information is processed semantically in SEMALON using ontologies. We described how multiple ontologies can coexist and how ontology translation is accomplished on spatial ontologies.

All these features are mandatory for a Spatial Data Infrastructure to enable adaptivity and interoperability in distributed, heterogeneous, and dynamic environments at a global scale.

REFERENCES

Andersen, Peter Bøgh, Nowack, Palle, 2002, Tangible Objects, Connecting Informational and Physical space. In: Qvortrup, L. (ed.): Virtual Space: Spatiality of Virtual Inhabited 3d Worlds, 190-210, Springer Publishers, London, Great Britain, 2002.

Bloomberg, J., Schmelzer, R., May 2002, The Pros and Cons of Web Services. ZapThink Report, http://www.zapthink.com

Gokhale, A., Kumar, B., Sahuguet, A., May 2002, Reinventing the Wheel? CORBA vs. Web Services. The Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA.

Goochild, Michael F., Kimerling, A. Jon (eds.), 2002, Discrete Global Grids, a Web Book, University of California, Santa Barbara, National Center for Geographic Information & Analysis, http://www.ncgia.ucsb.edu/globalgrids-book.

Goodstein, Lauren, 2004, Location Based Service: Analysis of Carrier Spending, Subscribers, Devices and Applications for Handset-based and Telematics Services. Report form ABI Research, http://www.abiresearch.com.

Hazas, M., Scott, J., Krumm, J., 2004, Location-Aware Computing Comes of Age. IEEE Computer, 37(2):95-97.

Hodes, Todd D., 2003, Discovery and Adaptation for Location-Based Services, Ph.D. Dissertation at UC Berkeley CS, USA.

Ibach, Peter, Horbank, Matthias, 2004/1, Highly Available Location-based Services in Mobile Environments. International Service Availability Symposium 2004, Munich, Germany.

Infineon Technologies, 2004, Infineon and Global Locate Develop an Extremely Sensitive, Low Power Single-Chip Enabling Assisted GPS Services for Mobile Phones. Infineon press release, http://www.infineon.com.

Jennings N., Sycara, K. et al., 1998, A Roadmap of Research and Development. Autonomous Agents and Multi-Agent Systems, 1(1):7–38.

Kephart, J., Chess, D., 2003, The Vision of Autonomic Computing. IEEE Computer, 36(1): 41-50.

Linthicum, D. S., 2003, Next Generation Application Integration: From Simple Information to Web Services. Addison-Wesley.

Malek, Miroslaw, 2004, Introduction to NOMADS. Computing Frontiers 2004, Ischia, Italy.

Malek, M., 2003, 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.

Milanovic, N., Richling, J., Malek, M., 2004, Lightweight Services for Embedded Systems. IEEE Workshop on Software Technologies for Embedded and Ubiquitous Computing Systems (WSTFEUS), Vienna, Austria.

Müller, P., Stich, Ch., Zeidler, Ch., 2001, Components @ work: Component Technology for Embedded Systems. 27th Euromicro Conference, Warsaw, Poland.

Papazoglou, M., 2001, 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.

Peng, Zhong-Ren, Tsou, Ming-Hsiang, 2004, Internet GIS – Distributed Geographic Information Services for the Internet and Wireless Networks, John Wiley & Sons.

Pinto, Helder, José, Rui, Boas, Noé Vilas, 2003, Using a Private Uddi For Publishing Location-Based Information To Mobile Users. Proceedings of the ICCC/IFIP 7th International Conference on Electronic Publishing (ElPub 2003), Guimarães, Portugal.

Rao, B., Minakakis, L., 2003, Evolution of Mobile Location-based Services. Communications of the ACM, 46(12):61-65.

Reichenbacher, T., 2004, Mobile Cartography – Adaptive Visualisation of Geographic Information on Mobile Devices. Dissertation submitted at the Institut of Photogrammetry und Cartography, Technical University, Munich, Germany.

Schilit, B., LaMarca, A., Borriello, G., Griswold, W., McDonald, D., Lazowska, E., Balachandran, A., Hong, J., and Iverson, V., 2003, Challenge: Ubiquitous Location-Aware Computing and the Place Lab Initiative. In Proceedings of The First ACM International Workshop on Wireless Mobile Applications and Services on WLAN (WMASH 2003), San Diego, CA, USA.

Schwan, K., Poellabauer, Ch., Eisenhauer, G., Pande, S., Pu, C., 2002, Infofabric: Adaptive Services in Distributed Embedded Systems. IEEE Workshop on Large Scale Real-Time and Embedded Systems (in conjunction with RTSS 2002), Austin, TX, USA.

Sutherland, J., van den Heuvel, W.-J., 2002, Enterprise Application Integration and Complex Adaptive Systems. Communications of the ACM, 45(10):59–64.

BIOGRAPHICAL NOTES

Peter Ibach

Oct. 1988-Aug. 1994:	study of computer science, Universität Dortmund, Germany
Sep. 1994-Feb. 1996:	System Engineer, IBM Service Center, Cologne, Germany
March 1996-March 1999:	Fellow of DFG graduate college "Communication-based Systems", Humboldt University, Berlin, Germany
April 1999-Sep. 1999:	Research Engineer, IBM, Böblingen, Germany
since Oct. 1999:	Research Assistant at Humboldt University, Berlin, Germany
since Aug. 2003:	Member of the Berlin Research Center on Internet Economics

Matthias Horbank

Since Oct. 1996: student of Computer Science and Economy, Humboldt University Berlin, Germany
since Oct. 1998: co-founder and CEO of IT Company Araneus Systemlösungen GbR, Berlin, Germany
March 2004 - Feb. 2005: Member of the Berlin Research Center on Internet Economics
since July 2004: working on master thesis on "Adaptive Location-based Services" at Humboldt University, Berlin, Germany

CONTACTS

Peter Ibach
Humboldt University Berlin
Computer Science Department
Computer Architecture and Communication Group
Rudower Chaussee 25
12489 Berlin
GERMANY
Tel. + 49 (0)30 2093 3038
Fax + 49 (0)30 2093 3029
Email: ibach@informatik.hu-berlin.de