Nr./No. 5 – Oktober/Octobre 2001

INFORMATIK

INFORMATIQUE

Zeitschrift der schweizerischen Informatikorganisationen

Revue des organisations suisses d'informatique

Ubiquitous Computing

A World Model for Location-Aware Systems

Matthias Grossmann, Alexander Leonhardi, Bernhard Mitschang, Kurt Rothermel

Advanced location-aware applications require a detailed model of the real world. The goal of the Nexus platform is to provide such a model together with generic functionality to a wide variety of location-aware applications. We describe the characteristics of this Augmented World Model and the architecture of the Nexus platform. We look in more detail at the two main components responsible for the main aspects of the world model, namely the spatial data and the position information of mobile objects.

Keywords: Mobile Computing, Location-Aware Applications, Augmented World Model.

1 Introduction

Recently, location-aware systems, under the name of Location Based Services, have received much attention as a promising area of application for mobile communication systems. To provide location-awareness, for example to show the current position of the user on a street map or to perform spatial queries like finding the restaurant next to the user's current position, these applications rely on a built-in model of the real world (see, for example, [Cheverst et al. 00]). The level of detail of the underlying model determines what functionality can be provided by the application. More advanced functionality that requires a detailed model could allow, e.g., to identify a real world object by pointing at it with a special device or even to create a so-called *Augmented Reality* by integrating information directly into the user's view of the real world using a heads-up display. Up to now, all location-aware applications use their own special world model, which is expensive to create and to maintain, especially in case of an up-to-date detailed 3D model.

The goal of the Nexus project is to develop a platform (shown in Fig. 1) that provides a detailed model of the real world to location-aware applications, both for indoor and outdoor usage [Hohl et al. 99]. This model, called the *Augmented World Model* (AW model), consists of representatives for real world objects and of virtual objects that provide, among other things, links to external information spaces like the WWW (Fig. 2). The model is kept up-to-date by integrating the update of sensor systems, for example to obtain the current position of mobile objects.

Location-aware applications may query the current state of this model by using the Augmented World Querying Language (AWQL) and receive as answer information about the model described by the Augmented World Modelling Language (AWML). We envision the Nexus platform as a middleware that brings together different providers and consumers of location-based information. Such a platform will hopefully enable the development of more powerful location-aware applications, as these can rely on a detailed model and generic functionality. To evaluate our concepts we plan to develop a city guide for the town of Stuttgart as an example application on top of the Nexus platform.

Matthias Grossmann works as a research assistant in the Department of Applications of Parallel and Distributed Systems at Stuttgart University. He received his M.S. degree (Diploma) from Stuttgart University in 2000. He is a member of the Nexus Research Group, where he is engaged in spatial data bases and federated information systems. <grossmms@informatik.uni-stuttgart.de>

Alexander Leonhardi works as a research assistant in the Distributed Systems Research Group at Stuttgart University. He received his M.S. degree (Diploma) from Stuttgart University in 1997. His current research interests are mobile computing, location-awareness, location management and ubiquitous computing. <leonhaar@informatik.uni-stuttgart.de>

Bernhard Mitschang is a professor for Database and Information Systems at Stuttgart University. He received the Ph.D. degree (Dr.-Ing.) in Computer Science from the University of Kaiserslautern in 1988. He got the *venia legendi* for practical Computer Science from the University of Kaiserslautern. From 1994 to 1998 he was a professor at the *Technische Universität München*. Prior to this he has been in the management of a project of the *Sonderforschungsbereich 124 der Deutschen Forschungsgemeinschaft* at the University of Kaiserslautern. From 1989 to 1990 he was on leave to IBM Almaden Research Center, San Jose, CA as a visiting scientist. His research interests are database management systems, including object-oriented support, semantic modelling, query languages, optimizations, and parallelism, and application-oriented areas like middleware and component technology, engineering information systems, and knowledge base management systems. <mitsch@informatik.uni-stuttgart.de>

Kurt Rothermel is a professor for Computer Science at Stuttgart University, where he leads the Distributed Systems Research Group. He received his doctoral degree in Computer Science from Stuttgart University in 1985. He then spent a sabbatical at the IBM Almaden Research Center in San Jose, and joined IBM's European Networking Center. His current research interests are communication architectures and protocols, distributed multimedia systems and mobile computing. <rothermel@informatik.uni-stuttgart.de>

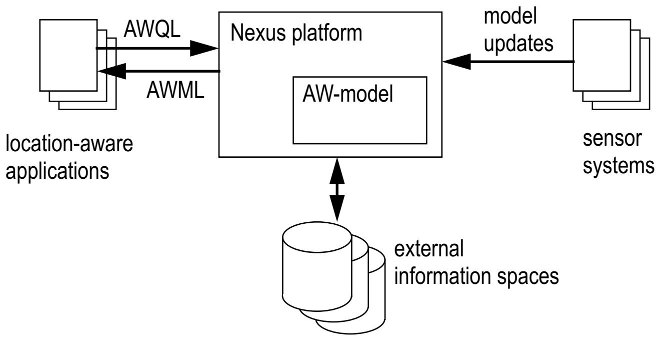

Fig. 1: Basic interactions of the Nexus platform.

In the remainder of this paper we describe the basic architecture of the Nexus platform in Section 2. The two major components that are responsible for managing the most important aspects of the AW model, namely the spatial model data and the location information of mobile objects, are described in Section 3 and Section 4, respectively.

2 The Nexus Platform for Spatially Aware Applications

2.1 Augmented World Model

The AW model [Nicklas et al. 01] is the central information model of the Nexus platform, providing the applications with an integrated, homogenous view on the available data. The most important types of information are models of geographic objects (e.g. buildings, streets or cities), models of mobile objects (e.g. Nexus users), and virtual objects. Examples for virtual objects are Virtual Information Towers [Leonhardi et al. 99] or Virtual PostIts. They have no counterpart within the real world, but can be visualized by an appropriate application. Virtual Objects provide additional information or services for Nexus users.

We are using an object-oriented approach, which allows the AW model to be easily extended. A subway station can, for example, be modelled as a subclass of *Station*, thus inheriting all attributes of *Station*. There are two categories of classes: members of the Standard Class Schema and the Extended Class Schema. The Standard Class Schema contains classes that we consider fundamental and have a well known structure, so they can be used by every application. There may be usage scenarios where additional classes are needed for special purposes. Such classes can be defined as subclasses of members of the Standard Class Schema – as shown for the subway station above – and belong to the Extended Class Schema. As *SubwayStation* inherits all attributes of *Station*, it still can be used as a *Station* object by applications that only know the Standard Class Schema, but special applications are needed to take advantage of the additional attributes of *SubwayStation*.

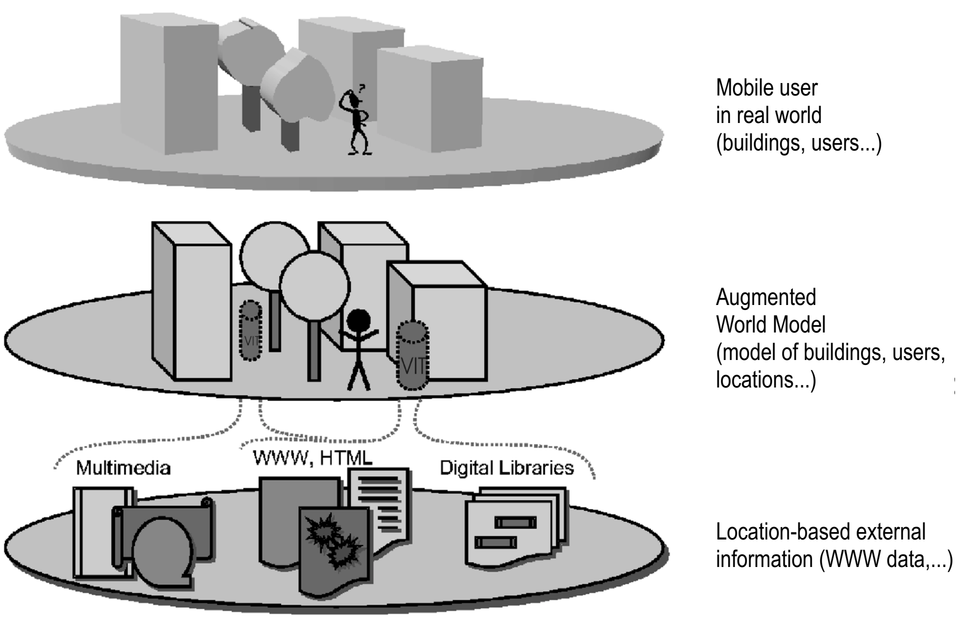

Fig. 2: Augmented World Model.

2.2 Architecture

Fig. 3 shows the main components of the Nexus architecture. It is organized in three tiers: The Service Layer contains the servers that provide the data needed by the applications. Spatial Model Servers provide data about static geographic objects, and Location Servers return on demand the location of mobile objects. Each Spatial Model Server stores information about spatial objects within a particular area. For example there could be a Spatial Model Server that stores 2D and 3D shapes of buildings located within Stuttgart. The Federation Layer contains Nexus nodes, which do not store data themselves, but receive requests from applications, look up appropriate servers, transmit the query to those servers and integrate the results into an homogenous result set [Busse et al. 99]. Recall that the WWW – the most popular open information system today – only provides the top (web browser) and the bottom layer (http, ftp and other servers). Compared to the WWW, Nexus needs the Federation Layer for two reasons: We want to present the data in a uniform and structured model with well known semantics (which does not exist on the WWW) and we want to enable applications to perform spatial queries without knowledge about the respective servers. In contrast, a centralized approach would not necessarily need the federation layer, but

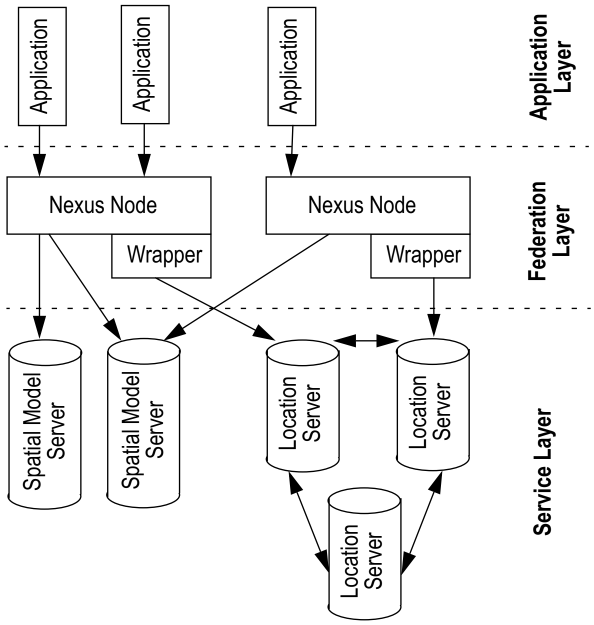

Fig. 3: Architecture of the Nexus system.

its scalability would be poor and comprehension of different providers would be difficult.

In brief, a query is processed by a Nexus node in the following way: Consider a client requesting data of users and buildings within a particular area. This query is sent to an arbitrary Nexus node. The Nexus node first has to distribute the query among the different services. Knowing that user is a subclass of *MobileObject* (stored at Location Servers) and building is a subclass of *StaticObject* (stored at Spatial Model Servers), the Nexus node creates two queries: One for users within the specified area and one for buildings within that area. The first query is sent to an arbitrary Location Server, which will contact additional Location Servers as necessary. The second query needs to be forwarded to the respective Spatial Model Servers. An additional component – the Area Service Register – is required to find out the addresses of those Spatial Model Servers. We will not discuss the Area Service Register further in this article, it can be regarded as a kind of geographic Domain Name System.

Nexus nodes and Spatial Model Servers are accessed via a query-response protocol. The query contains a specification of the objects the application is interested in, e.g. objects within a particular area or of a particular type. The answer is a list of matching objects. We are using SOAP [W3C 00] as communication protocol because of its independency of the underlying transport protocol and programming language. As we will show in the following section, most transferred data is already encoded using XML. In contrast to what has been said before, the Location Service provides a specialized interface which is contacted via a wrapper.

3 Management of Model Data

Products like DBMS with extensions for spatial objects or GIS are already available. However, we focus on the development of an interface that provides the functionality needed for typical location-aware applications and hides the details of the underlying data management. Queries to this interface are formulated in AWQL. This is an XML-based language, which contains the following elements: (a) A restriction, which is a boolean expression made of relations between attributes of objects and fixed values, (b) a filter, which allows an application to remove attributes it is not interested in from the result and (c) a closest-predicate, which allows to narrow the result to the *n* objects closest to a given position. The example in Fig. 4 shows a query for the addresses and menus of all restaurants within Stuttgart.

The answer to this query would be a list of restaurant objects with address and menu attributes, serialized as AWML, which also is XML-based. We will use common data formats where possible, such as the GML polygon [OpenGIS 01] in the example above. As XML becomes more popular as format for data-exchange, and many newly developed data formats are XML based, using an XML data format for Nexus has the advantage that such data can easily be embedded in AWQL or AWML.

4 Management of Location Information

The up-to-date location information of mobile objects, which is a important aspect of the AW model, has a very different characteristic from the much more static spatial model data, especially if the information is required with a high accuracy (in case of a differential GPS the information is available with an accuracy of 5 meters or less). As it is very volatile (in case of a walking person it has to be updated about once per second to achieve the above accuracy), updates will have to be very frequent. Existing databases are not well suited for these

```
<awql>
    <restriction>
        <and>
            <equal>
                <attr name="type"/>
                <nexusdata>restaurant</nexusdata>
            </equal>
            <inside>
                <attr name="base"/>
                <gml:polygon>...GML representation
                of base of Stuttgart...
                </gml:polygon>
            </inside>
        </and>
    </restriction>
    <filter>
        <includes>
            <attr name="adress"/>
            <attr name="menu"/>
        </includes>
        <excludeallother/>
    </filter>
</awql>
```

Fig. 4: Example of an AWQL query.

requirements, as they assume a low number of updates and the extensions required for performing spatial queries are not yet able to perform a higher number of transactions. Consequently, mobile objects are managed in the Nexus platform by a special distributed component, the Location Service (LS), which primarily stores the current location of each mobile object. This location is given as a geographic coordinate of the WGS84 format, which is used in GPS.

The LS is contacted by the Federation Layer, whenever a query to the AW model concerns the current location of a mobile object. To this end it provides three main types of queries that are reflected in the AWQL: (a) *position queries* return the current location of a certain mobile object, for example that of the guide of a tourist group, (b) *range queries* determine all mobile objects inside a certain area, for example for finding all members of the tourist group that are currently in a restaurant, and (c) *nearest neighbour queries* return the mobile object nearest to a given position, for example the nearest taxi cab. For security reasons the LS allows the user to set an upper limit on the accuracy with which it handles his/her position information. The semantics of the queries were designed to handle this issue. For details see [Leonhardi/Rothermel 01].

To achieve the scalability necessary for a large-scale deployment of the platform, the servers that constitute the LS are organized in a hierarchical fashion, similar to that of the Globe location system for software objects [v. Steen et al. 98]. Leaf servers in this hierarchy are responsible for managing the position and registration information for the mobile objects inside their disjunct service areas, while the higher level servers are responsible for forwarding queries and handovers. As a fast processing of queries and especially updates concerning location information is of great importance, the location information will be managed in a special main memory data structure based on a Quad-tree [Samet 90], while the registration information is stored in a traditional database. The volatile position information, which will be out-of-date after a server failure anyway, can be recovered from the mobile objects. To evaluate the achievable performance and scalability, we have developed a prototype of the LS, which is able to process as many as 2000 updates and 1000 queries per second on a standard Solaris workstation.

5 Conclusion

In this paper, we described the main idea of the Nexus platform, that is to provide location-aware applications and service providers with an infrastructure for a detailed up-to-date world model. We have looked at the main components of the Nexus platform that are responsible for managing the model information, the Spatial Model Servers and the Location Service. Other research topics that are addressed in the Nexus project but are not the focus of this paper are a communication component that allows a transparent handover between different mobile communication systems, techniques for the automatic generation of detailed realistic models, appropriate sensor systems and the location-aware hoarding of information items.

Currently, four departments at three institutes of the University of Stuttgart are engaged in the Nexus project. We are building a small-scale prototype of the platform and an example application to demonstrate the basic features of the system. Future work will focus on refining the internal components and their interaction in order to enhance performance.

References

[Busse et al. 99]
S. Busse, R.-D. Kutsche, U. Leser, H. Weber: Federated Information Systems: Concepts, Terminology and Architectures. *Forschungsberichte des Fachbereichs Informatik*, Bericht Nr. 99-9, Technische Universität Berlin.

[Cheverst et al. 00]
K. Cheverst, N. Davies, K. Mitchell and A. Friday: Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project, *Proceedings of the Sixth International Conference on Mobile Computing and Networking* (MobiCom 2000), pp. 20–31, ACM Press, 2000.

[Hohl et al. 99]
F. Hohl, U. Kubach, A. Leonhardi, K. Rothermel and M. Schwehm: Next Century Challenges: Nexus – An Open Global Infrastructure for Spatial-Aware Applications, *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking* (MobiCom'99), Seattle, Washington, pp. 249–255, ACM Press, 1999.

[Leonhardi et al. 99]
A. Leonhardi, U. Kubach, K. Rothermel and A. Fritz: Virtual Information Towers – A Metaphor for Intuitive, Location-Aware Information Access in a Mobile Environment, *Proceedings of the Third International Symposium on Wearable Computers* (ISWC'99), San Francisco, CA, USA, IEEE Press, 1999.

[Leonhardi/Rothermel 01]
A. Leonhardi and K. Rothermel: *Architecture of a Large-scale Location Server*, Technical Report TR-2001-01, Faculty of Computer Science, University of Stuttgart, 2001.

[Nicklas et al. 01]
D. Nicklas, M. Grossmann, T. Schwarz, S. Volz and B. Mitschang: A Model-Based, Open Architecure for Mobile, Spatially Aware Applications. *Proceedings of the 7th International Symposium on Spatial and Temporal Databases* (SSTD'2001). (to appear)

[OpenGIS 01]
Open GIS Consortium: *Geography Markup Language* (GML) 2.0. OGC Recommendation Paper, 20 February 2001, http://www.opengis.net/gml/01-029/GML2.html

[Samet 90]
H. Samet: *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990, ISBN 0-201-50255-0.

[v. Steen et al. 98]
M. v. Steen, F. Hauck, P. Homburg and A. Tanenbaum: Locating Objects in Wide-Area Systems, *IEEE Communications Magazine*, pp. 2–7, 1998.

[W3C 00]
World Wide Web Consortium: *Simple Object Access Protocol* (SOAP) 1.1. W3C Note 08 May 2000 http://www.w3.org/TR/SOAP.