

Eine dezentral organisierte Middleware zur Entwicklung ortsbezogener Dienste

Jörg Roth
Fernuniversität Hagen
58084 Hagen
Joerg.Roth@Fernuni-hagen.de

1 Einleitung

Der Aufenthaltsort eines mobilen Benutzers stellt eine wichtige Information für Anwendungen aus den Bereichen *Mobile Computing*, *Wearable Computing* oder *Ubiquitous Computing* dar. Ist ein mobiles Endgerät in der Lage, die aktuelle Position des Benutzers zu bestimmen, kann diese Information von der Anwendung berücksichtigt werden – man spricht dabei allgemein von *ortsbewussten* oder *ortsbezogenen Anwendungen* bzw. von *ortsbezogenen Diensten*. Die Positionsbestimmung hat eine zentrale Bedeutung für ortsbewusste Anwendungen. Häufig wird bei der Entwicklung solcher Anwendungen eine präzise und einfache Positionsbestimmung, z.B. durch Satellitennavigation, vorausgesetzt. Die Satellitennavigation steht jedoch oft nicht zur Verfügung, beispielsweise innerhalb von Gebäuden. Als Alternative können zellulare Netze des Mobilfunks zur Positionsbestimmung eingesetzt werden. Hierbei ist die Positionsauflösung jedoch sehr ungenau. Eine Fülle weiterer Positionsbestimmungssysteme steht insbesondere in Gebäuden zur Verfügung.

Generell kann man feststellen, dass die verschiedenen Positionsbestimmungssysteme höchst unterschiedliche Eigenschaften bezüglich Abdeckung, Präzision oder Art der zur Verfügung gestellten Positionsangaben haben [1, 9]. Kein Positionsbestimmungssystem berücksichtigt zurzeit alle Belange denkbarer ortsbewusster Anwendungen. Betrachtet man aktuelle Entwicklungen auf dem Gebiet ortsbewusster Anwendungen, z.B. auf dem Mobilfunksektor, kann man konkret folgende Probleme identifizieren:

- Ein einzelnes System zur Positionsbestimmung ist häufig fest, ohne Möglichkeit der späteren Änderung, in die Anwendung integriert. Anwendungen können daher auf Änderungen auf dem Gebiet der Positionsbestimmung nur schwer reagieren. Günstiger wäre eine vollständige Trennung von Anwendung und Positionsbestimmung.
- Die Positionsangaben durch die Positionierungssysteme sind in der Rohform für die Anwendung häufig ungeeignet. So ist eine Anwendung neben den reinen Koordinaten oft an Informationen über den aktuellen Standort interessiert. Häufig werden solche Informationen jedoch nicht zur Verfügung gestellt oder aufwändig durch die Anwendung selbst generiert.
- Viele wiederkehrende Operationen mit Ortsbezug werden noch innerhalb der Anwendung kodiert. Hier wäre eine Plattform sinnvoll, die der Anwendung einen Satz von Standard-Operationen zur Verfügung stellt.
- Dezentral vorliegende Informationen über Orte werden häufig noch zentral gespeichert, obwohl solche Lösungen schwer zu administrieren sind und schlecht skalieren.

Als Lösung für diese Probleme bietet sich eine dezentral organisierte Plattform an, die entsprechende Dienstleistungen zur Verfügung stellt.

2 Nimbus – eine Plattform für ortsbewusste Anwendungen

Um ortsbezogene Anwendungen und Dienste effizient entwickelt zu können, habe ich die *Nimbus*-Plattform vorgeschlagen [5, 6, 7, 8]. Ein mobiler Anteil der Plattform wird auf dem mobilen Endgerät installiert und bindet beliebige Systeme zur Positionsbestimmung transparent an. Ein Netzwerkanteil von Nimbus nimmt eine Erweiterung der Positionsangaben vor. Hinter dem Netzwerkanteil verbirgt sich ein dezentraler, selbstorganisierender Verbund logisch gekoppelter Server. Erweiterte Positionsangaben werden schließlich für Datenbankabfragen, Anfragen zu Webdiensten, entfernte Prozedur- bzw. Methodenaufrufe oder Anfragen an Benutzerregister verwendet. Die Nimbus-Plattform erbringt folgende Dienstleistungen:

- Es wird eine geeignete Auswahl unter den verfügbaren Positionsbestimmungssystemen getroffen und eine Sensordatenfusion durchgeführt. Liegen nur Positionsdaten mit lokalem Gültigkeitsbereich vor, werden diese zusätzlich in eine global gültige Repräsentation umgewandelt. Positionsdaten werden schließlich mit von der Anwendung vorgegebenen geodätischen Eigenschaften (z.B. in ebenen oder ellipsoidischen Koordinaten) zur Verfügung gestellt.
- Es werden so genannte *semantische Positionen* bereitgestellt (Positionen mit Bedeutung für Mensch oder Anwendung, z.B. "Campus der Fernuniversität"). Neben einer Liste von semantischen Positionsnamen (nach einem vordefinierten Namensschema) werden auch Meta-Informationen über den Ort angeboten. So wird beispielsweise die aktuelle Position in Klassen wie "Wald", "Parkplatz" oder "Gebäude" eingeteilt. Neben Informationen über den Ort werden auch Informationen über die nähere Umgebung in einem von der Anwendung definierten Radius bereitgestellt. Hierdurch können Suchanfragen der Art "Welche Hotels liegen im Umkreis von 5 km?" bearbeitet werden.
- Die Anwendung kann zusätzlich ortsbezogene Standard-Dienste wie Trigger-Dienste oder das Geocasting nutzen.

Die Architektur des Nimbus-Rahmenwerks ist in Abbildung 1 dargestellt. Nimbus kann in die drei Schichten *Basisschicht*, *Service-Schicht* und *Anwendungsschicht* eingeteilt werden.

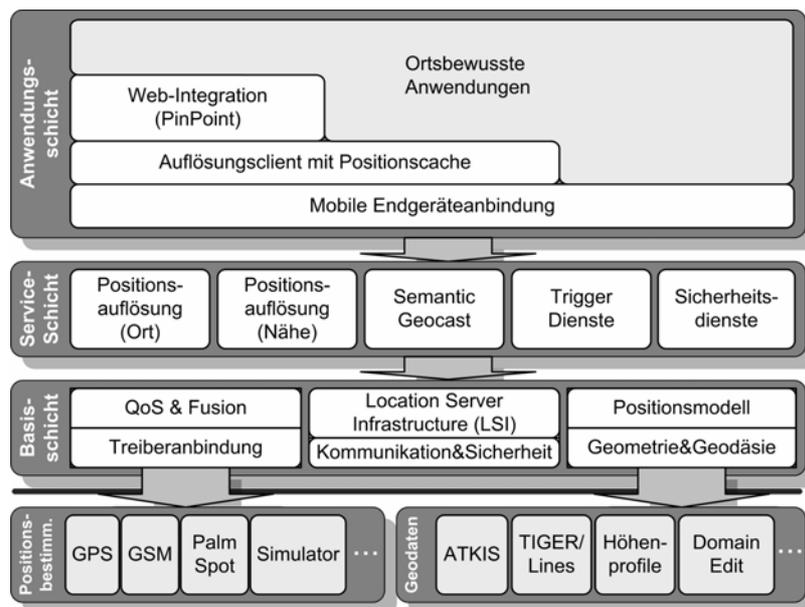


Abbildung 1: Das Nimbus Rahmenwerk

Die *Basisschicht* umfasst die *Anbindung an die Positionsbestimmung*, die Dienst-Infrastruktur *LSI* und das *Positionsmodell*.

- Durch die Treiberanbindungskomponente [7] können Positionsbestimmungssysteme später verändert und hinzugefügt werden, ohne dass die aufsetzenden Komponenten modifiziert werden müssen. Erreicht wird dies über Treiber, die wie in Betriebssystemen die verwendete Hardware abstrahieren. Eine Komponente *QoS & Fusion* führt eine Auswahl unter den Positionsbestimmungssystemen anhand von Qualitätsanforderungen der Anwendung durch und verknüpft die Positionsdaten durch eine Sensordatenfusion.

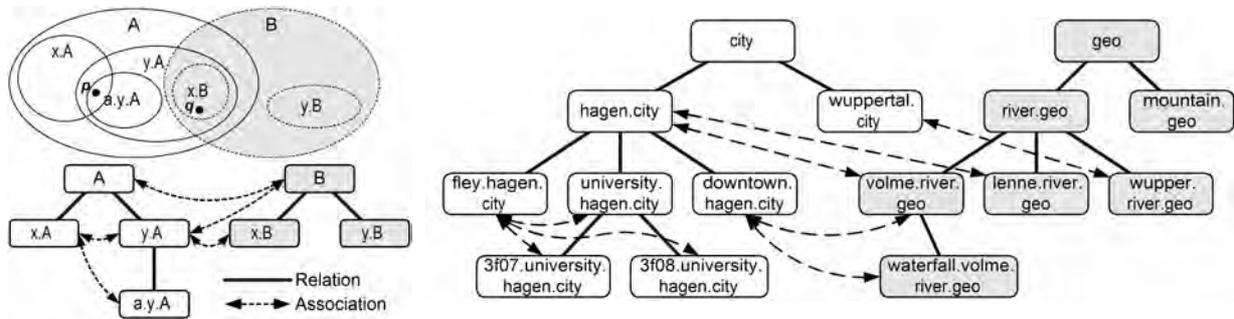


Abbildung 2: Das Nimbus-Positionsmodell

- Das *Positionsmodell* [4, 6, 8] stellt einen Formalismus zur Instanziierung der Positionsdaten zur Verfügung, die später in der Plattform gespeichert werden. Grundobjekte sind dabei die *Domänen*, die im Wesentlichen aus der semantischen Positionsbezeichnung gemäß einem vorgegebenen Namensschema und der entsprechenden physikalischen Ausdehnung bestehen. Das Positionsmodell definiert losgelöst von der späteren Speicherung den Zusammenhang zwischen den einzelnen Domänen anhand der Beziehungen *Association* und *Relation* (Abbildung 2) [6]. Inhaltlich zusammenhängende Domänen werden zu *Hierarchien* zusammengefasst. Um zwischen verschiedenen Koordinatensystemen umzurechnen und notwendige Polygonoperationen effizient durchzuführen, steht eine Bibliothek mit Geometrie- und Geodäsie-Funktionen zur Verfügung.

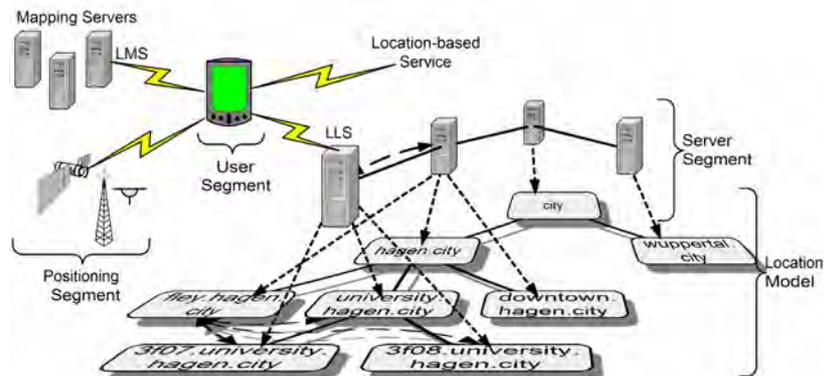


Abbildung 3: Zusammenhang zwischen Positionsmodell und Infrastruktur

- Die Dienstinfrastuktur *LSI* [3, 7] besteht aus einem selbstorganisierenden Verbund von *Location Servern*, die jeweils einen bestimmten Anteil der verfügbaren Positionsdaten verwalten. Abbildung 3 zeigt, wie das Positionsmodell und die Infrastruktur zusammenhängen. Die Infrastruktur speichert Domänen dezentral: ein Location Server ist immer für eine komplette Sub-Hierarchie verantwortlich bis zu den Domänen, die wiederum durch eigene Location Server repräsentiert werden. Die logischen Beziehungen zwischen Domänen werden dabei auf die entsprechenden Location Server übertragen. Neben den Location Servern stellt die Infrastruktur *Mapping Server* zur Verfügung, die jeweils einen physikalischen Bereich mit den dort eingesetzten Positionsbestimmungssystemen betreuen und insbesondere die Umrechnung von lokalen in globale Positionen vornehmen.

Die *Service-Schicht* umfasst Service-Primitive, die durch die Anwendung direkt angesprochen werden können.

- Die zwei Komponenten zur *Positionsauflösung* [6, 8] dienen der Erweiterung von Positionsinformationen. Es wird eine flächendeckende Versorgung von semantischen Positionen zur Verfügung gestellt, auch wenn diese durch die Positionsbestimmungssysteme selbst nicht ermittelt werden können. Zusätzlich werden Metadaten zu den Positionen bereitgestellt.
- Die Komponente *Semantic Geocast* [5] erlaubt, Netzwerknachrichten an eine Gruppe mobiler Benutzer zu versenden, die eine bestimmte Position einnehmen. So können beispielsweise mit einer einzelnen Operation Nachrichten an alle Personen auf dem Campus versendet werden.
- Durch *Trigger-Dienste* kann die Anwendung informiert werden, wenn ein bestimmtes positionsrelevantes Ereignis eintritt. So kann beispielsweise die Anwendung informiert werden, wenn der mobile Benutzer den Campus verlässt oder betritt.
- Die Sicherheitskomponente behandelt die Frage nach authentischen Positionsdaten. Ohne Vorkehrungen könnte ein Angreifer eigene Location Server einrichten, die irreführende oder falsche Positionsinformationen verbreiten. Eine Sicherheitsinfrastruktur mit Zertifizierungsstellen verhindert dies, indem der mobile Benutzer zu jeder Dienstnutzung die Zertifikate der beteiligten Location Server abrufen kann.

Die *Anwendungsschicht* enthält die ortsbewussten Anwendungen, die *Endgeräteeinbindung* durch die Kommunikations-Middleware *NKF (Network Kernel Framework)* [2], die Clientkomponente zur Positionsauflösung und die Komponente zur Web-Integration *PinPoint* [3]. Letztere erlaubt, ortsbezogene Webdienste auf der Basis von Standard-Webkomponenten zu entwickeln. Das System setzt auf den Komponenten *Positionsbestimmung* und *Geodaten* auf. Diese gehören formal nicht mehr zum Nimbus-Rahmenwerk, dennoch sind hierzu Eigenentwicklungen durchgeführt worden. So wurde eine Simulationsumgebung für Positionsbestimmungssysteme und Werkzeuge für das Einlesen von Geodaten aus den Tiger/Line- und ATKIS-Datenbanken entwickelt.

Referenzen

1. Jörg Roth: *Mobile Computing*, dpunkt-Verlag, 2002
2. Jörg Roth: *A Communication Middleware for Mobile and Ad-hoc Scenarios*, International Conference on Internet Computing (IC'02), Las Vegas (USA), 24.-27. Juni 2002, Vol. I, CSREA Press, 77-84
3. Jörg Roth: *Context-aware Web Applications Using the PinPoint Infrastructure*, IADIS International Conference WWW/Internet 2002, Lissabon (Portugal), 13.-15. Nov. 2002, IADIS Press, 3-10
4. Jörg Roth: *Flexible Positioning for Location-based Services*, IADIS International Conference e-Society, Lissabon (Portugal), 3.-6. Juni 2003, Vol. I, IADIS Press, 296-304
5. Jörg Roth: *Semantic Geocast Using a Self-organizing Infrastructure*, Innovative Internet Community Systems (I2CS), Leipzig, 19.-21. Juni 2003, LNCS 2877, Springer-Verlag, 216-228
6. Jörg Roth: *Accessing Location Data in Mobile Environments – the Nimbus Location Model*, Mobile HCI 03 Workshop on Mobile and Ubiquitous Information Access, Udine (Italien), 8. Sept. 2003, LNCS 2954, Springer-Verlag, 256-270
7. Jörg Roth: *Flexible Positioning for Location-based Services*, IADIS Journal on WWW/Internet, Vol. I, Nr. 2, Dez. 2003, IADIS Press, 18-32
8. Thomas Hadig, Jörg Roth: *Proximity Services with the Nimbus Framework*, IADIS International Conference Applied Computing 2004, Lissabon (Portugal), 23.-26. März 2004, IADIS Press, 437-444
9. Jörg Roth: *Data Collection*, in Jochen Schiller, Agnès Voisard (eds), *Location-Based Services*, Morgan Kaufmann Publishers, erscheint im Juni 2004