Service Discovery

Automatisches Auffinden mobiler Dienste

Kai-Oliver Detken

Das World Wide Web (WWW) offeriert eine Fülle an Informationen und bietet den Zugang zu diversen Diensten. Die meisten von ihnen beschränken sich jedoch auf die Darstellung von Informationen, die vom Benutzer durch einen Web-Browser abgerufen werden können. Es sind größtenteils Informationsdienste mit einer statischen Adresse. Durch neue Techniken ist der Benutzer immer mehr in der Lage, von jedem Ort aus auf vorhandene Informationen zuzugreifen. Mobiltelefone oder PDAs eröffneten neue Wege für die Kommunikation und Informationsgewinnung. Durch neue Mobilfunksysteme wie z.B. GPRS oder UMTS können aufbereitete Informationen schneller und in größerem Umfang zum Benutzer gelangen.

Dr.-Ing. Kai-Oliver Detken ist Geschäftsführer der DECOIT GmbH sowie Dozent und freier Autor in Bremen

Solche Möglichkeiten bringen auch neue Randbedingungen mit sich. Mobile Benutzer befinden sich in einem ständig wechselnden Kontext. Daher ist es interessant, wie sich z.B. der Standort des Benutzers auf den Dienst auswirken kann:

- In welcher Umgebung wird der Dienst angeboten?
- Wird der Dienst in verschiedenen Umgebungen bemerkt?
- Wie greifen die Benutzer auf diesen Dienst zu?
- Für welche Benutzergruppe ist dieser Dienst interessant?

Dabei muß zwischen dem mobilen und dem stationären Benutzer am PC unterschieden werden. Letzterer hat mehr Möglichkeiten und wohl auch mehr Zeit, um nach gewünschten Diensten zu suchen. Außerdem kennt er seine Umgebung, und auch die Anforderungen an den Dienst werden sich über die Zeit nur geringfügig ändern. Der mobile Benutzer hingegen könnte an jedem Ort unterschiedliche Dienste nutzen, die sich z.B. auf seinen Stand-ort beziehen. Dabei kann ein automatisches Service Discovery zum Auffinden des Servers, der die Dienste vorhält, sehr hilfreich sein.

Service Discovery

Ist ein Dienst erfolgreich definiert, generiert und veröffentlicht worden, so muß dieser durch den Benutzer zu orten sein. Das Orten von Diensten wird Service Discovery genannt. Es soll dem Benutzer Methoden zur Verfügung stellen, mit deren Hilfe er in einem bestimmten Kontext nach einem Dienst suchen kann. Auch hier ist wieder der Unterschied zwischen stationärem und mobilem Benutzer besonders zu beachten.

Der mobile Benutzer kennt den neuen Kontext noch nicht. Nicht nur die angebotenen Dienste sind für ihn neu, sondern eventuell auch die Art und Weise, wie er auf sie zugreifen muß. Daher gilt es, die Service Discovery so zu gestalten, daß jeder Benutzer genau weiß, wie er nach einem Dienst suchen kann, daß immer gültige Dienste angeboten werden und daß der ganze Vorgang transparent und vor allem immer gleich abläuft. Die derzeitigen Protokolle des Service Discovery stellen Mechanismen für das Orten der verfügbaren Dienste und das Sammeln der notwendigen Informationen über das Suchen und Durchsehen aller vorhandenen Dienste, die Auswahl des richtigen Dienstes und für seine Benutzung zur Verfügung.

Der Benutzer kann durch verschiedene Protokolle aktiv nach einem Dienst suchen oder passiv auf das Angebot eines Anbieters reagieren. Es gibt vier grundlegend verschiedene Ansätze, einen Dienst zu orten, die sich entweder in der Art der Datenhaltung oder in der Kommunikation der an dem Vorgang des Service Discovery beteiligten Knoten unterscheiden:

- Der *zentralisierte Ansatz* setzt auf einer zentralen Instanz auf, der Service Repository. Alle Informationen werden hier abgelegt und nach Kriterien wie Themengebieten, Unternehmensbranchen oder andere Informationsstrukturen sortiert. Meist

Das Thema in Kürze

Der Erfolg neuer Mobilfunksysteme wie UMTS wird oft an der Einführung einer Vielzahl interessanter Datendienste festgemacht. Doch was nützen die schönsten Dienste, wenn der mobile Nutzer sie nicht finden kann. Genau hier setzt das Service Discovery zum Orten der Dienste in einem bestimmten Kontext an. Der Beitrag beschreibt die Funktionsweise des Ortungsvorgangs und die ihm zugrundeliegenden Protokolle.

ist das Service Repository öffentlich zugänglich. Eine Authentifizierung ist häufig erst dann erforderlich, wenn ein neuer Eintrag zum Service Repository hinzugefügt werden soll. Das beste Beispiel erfolgreicher Umsetzung eines Service Repository ist das Protokoll UDDI.

- Der *verteilte Ansatz* legt alle Dienste in einer vorgegebenen Struktur (z.B. Land, Stadt, Ort, Branche) an. Es werden unterschiedliche Server zur Datenhaltung verwendet. Genau nach diesem Prinzip funktioniert ein Verzeichnis. Ein Verzeichnis ist hierarchisch aufgebaut und bietet die Möglichkeit, auf andere Verzeichnisse gleicher Beschaffenheit zu verweisen. Protokolle, die hierauf basieren, sind X.500 und LDAP.
- Durch immer schnellere Client-Systeme ist man auf die Idee gekommen, deren Ressourcen in den Kommunikationsprozeß stärker zu integrieren. Bei diesem *Peer-to-Peer-Ansatz* wird ein Client gleichzeitig als Client und Server benutzt. Es gibt keine 1-zu-n-Beziehung mehr, sondern es bestehen beliebig viele Verbindungen zwischen den einzelnen Systemen. Peer-to-Peer-Netze wurden durch File-Sharing-Programme wie Napster oder eDonkey bekannt.
- *Broadcast* bedeutet, daß ein zentraler Netzknoten Werbung für die angebotenen Dienste macht, entweder als Diensteanbieter selbst oder als Vermittler für andere Knoten im Netz, die Dienste anbieten wollen. Hier kommt das Service Location Protocol (SLP) zur Anwendung.

Universal Description, Discovery and Integration (UDDI)

Bei UDDI handelt es sich um eine Spezifikation eines webbasierten Verzeichnisses für Webservices. Der Fokus eines UDDI-Verzeichnisses liegt hauptsächlich auf Service Discription und Service Discovery. Um auf ein UDDI-Verzeichnis zugreifen zu können, wurde eine UDDI-API veröffentlicht, die auf dem XML-Standard basiert. Anbieter von Webservices haben nun die Möglichkeit durch die Verwendung dieser UDDI-API einen Service zu registrieren. Danach können die Anwender über eine gewählte Schnittstelle nach einem Dienst suchen. Die Schnittstellen sind meist freie Programme, die über Java oder Visual Basic umgesetzt wurden. Allerdings setzen sich zur Zeit immer mehr webbasierte UDDI-Browser durch, die sogar auf beliebige UDDI-Verzeichnisse direkt zugreifen können.

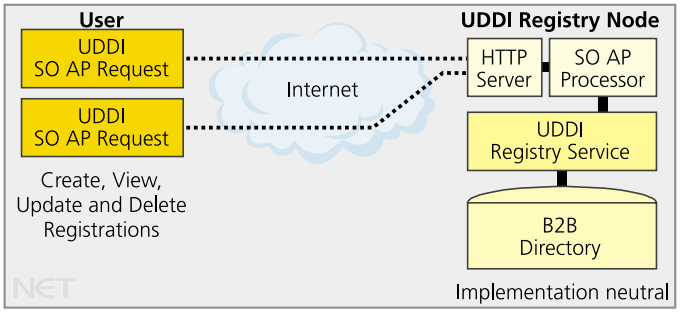

Bild 1: UDDI- und SOAP-Kommunikation

Webservices können mit normalen Internetprotokollen wie HTTP angesprochen werden und funktionieren vollkommen transparent für den Anwender. Die Implementierung des Webservices hat keinerlei Auswirkung auf die Bedienung bzw. auf die Rückgaben des Systems. Solange das Webservice-Interface die Spezifikation erfüllt, ist es unabhängig von der Programmiersprache.

Zum Transport der Informationen wird das auf XML basierende Simple Object Access Protocol (SOAP) verwendet, mit dessen Hilfe Informationen und Daten zwischen zwei verschiedenen Systemen ausgetauscht werden können. Da die einzelnen Nachrichten mit HTTP im Klartext übertragen werden, sollte für sensible Daten das HTTPS-Protokoll verwendet werden.

Damit ein neu geschaffener Webservice nun auch gefunden werden kann, wird dieser in einem UDDI-Verzeichnis registriert, in dem ausschließlich Webservices abgelegt werden können (*Bild 1*). Es gibt hierbei zwei Arten der Registrierung: Zum einen werden die Informationen eines Webservice-Anbieters übermittelt und gespeichert. Dazu gehören Kontakte und Adressen sowie Webdienste. Die andere Registrierung betrifft die Servicetypen. Sie helfen, alle gehalten Anbieter und deren Webdienste einem Typ zuordnen und in Kategorien aufteilen zu können.

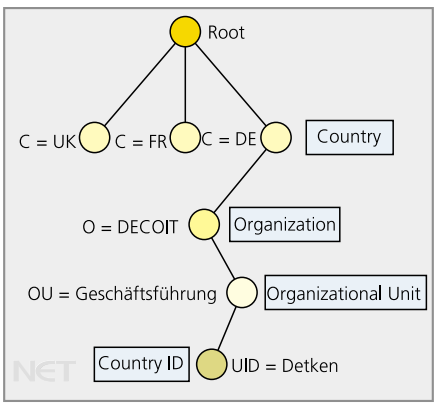

Bild 2: LDAP Directory Information Tree (DIT)

Der Webserver eines UDDI Registry Nodes wertet nun die eingehenden Nachrichten mittels eines XML/SOAP-Prozessors aus. Aus dessen Ergebnis können nun UDDI-konforme Abfragen generiert und andere Funktionen ausgeführt werden. Ein UDDI Registry Service führt die gewünschten Abfragen durch und gibt das Ergebnis über den Webserver zurück. Dazu werden die Daten wieder in SOAP-Nachrichten verpackt und via HTTP an den anderen Knoten zurückgeschickt.

Lightweight Directory Access Protocol (LDAP)

LDAP ist ein Abfrageprotokoll für herkömmliche X.500-Verzeichnisse, die nicht mehr über den kompletten OSI-Stack, sondern über den IP-Stack angesprochen werden. Mit Hilfe des X.500-Standards können große Datenmengen in einem hierarchischen Baumgebilde organisiert werden.

Um auf Verzeichnisse dieser Art zuzugreifen, wird das Directory Access Protocol (DAP) benötigt. Das bedeutet, daß jede Anwendung den gesamten OSI-Stack implementieren muß, was eine sehr komplexe und aufwendige Aufgabe darstellt. Aus diesem Grund wurde LDAP entwickelt, das auf den einfacheren TCP/IP-Stack aufbaut.

LDAP in der Version 3 wurde durch den Internet Standard RFC-2251 definiert. Die folgenden verwandten RFC-Standards der IETF wurden an die LDAP-Version 3 angepaßt bzw. sind neu hinzugekommen:

- Lightweight Directory Access Proto-

col (v3): Attribute Syntax Definitions (RFC-2252);
- Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names (RFC-2253);
- The String Representation of LDAP Search Filters (RFC-2254);
- The LDAP URL Format (RFC-2255);
- A Summary of the X.500(96) User Schema for Use with LDAPv3 (RFC-2256).

Alle Informationen, die von einem LDAP-Verzeichnis gehalten werden, basieren auf einzelnen Einträgen, sog. LDAP-Einträgen. Ein LDAP Entry hat immer einen eindeutigen Namen, der auch als Distinguished Name (DN) bezeichnet wird. Jeder Eintrag kann dabei ein oder mehrere Attribute haben. Diese sind von einem bestimmten Typ, haben einen Namen und meist einen oder mehrere Werte.

Damit in einem LDAP-Verzeichnis neue Objekte erzeugt werden können, müssen diese vorerst in einem Schema definiert werden. Daher werden alle Objekte in Klassen organisiert, die eine eindeutige ID und einen eindeutigen Namen haben. Auch müssen die Attribute des Objektes angegeben werden. Damit diese Attribute verwendete werden können, müssen sie vorab deklariert werden.

Alle Einträge werden in einer hierarchischen, baumartigen Struktur gehalten, die meist durch die geografischen oder organisatorischen Gegebenheiten des Umfeldes vorgegeben ist. Ganze Unternehmensstrukturen können so abgebildet werden. *Bild 2* zeigt einen typischen LDAP-Baum.

Der LDAP-Baum kann von einem oder mehreren LDAP-Servern umgesetzt werden. Ein LDAP-Client verbindet sich mit einem LDAP-Server und sucht nach einem bestimmten LDAP-Eintrag. Sollte der Eintrag beim befragten LDAP-Server vorhanden sein, bekommt der Client eine direkte Antwort mit allen Informationen zum Eintrag. Es kann aber auch sein, daß der LDAP-Eintrag nicht von dem befragten Server verwaltet wird, dieser aber genau weiß, welcher Server für den Eintrag zuständig ist. Daher sind LDAP-Server über sog. LDAP-Referrals miteinander verbunden.

Service Location Protocol (SLP)

Wenn ein Benutzer auf herkömmliche Weise auf einen Dienst im Netz zugreifen möchte, muß er die URL oder die Netzadresse des Servers kennen. Kommt ein Benutzer neu hinzu, so weiß dieser nichts über die Existenz eines möglichen Servers und dessen Dienste. Dank Service Location Protocol (SLP), das derzeit in der Version 2 spezifiziert ist, hat er die Möglichkeit, nach den fehlenden Informationen „zu fragen". Die SLP-Architektur besteht aus drei Agenten, die miteinander kommunizieren (*Bild 3*):

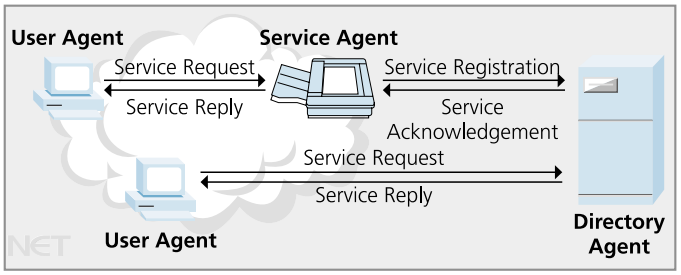

Bild 3: SLP-Architektur

- Der *User Agent* (UA) übernimmt auf der Client-Seite das Service Discovery und ist für Anfrage und Auswertung der Antworten zuständig.
- Der *Service Agent* (SA) vertritt den Dienst und macht „Werbung". Er kann direkt auf Anfragen eines UA antworten und ihm die nötigen Informationen zukommen lassen.
- Der *Directory Agent* (DA) sammelt alle Informationen von allen vorhandenen SAs und speichert diese in einer Datenbank. Auch er kann die Anfragen eines UA bearbeiten.

Sollte nun ein neuer Service zu den vorhandenen hinzukommen, muß der SA diesen beim DA anmelden (Service Registration) und wird bei erfolgreicher Registrierung von diesem bestätigt. Sucht ein Benutzer nach einem bestimmten Service, fragt er den DA und durchsucht dessen Services nach bestimmten Kriterien.

Zusammenführen der Ansätze

Das neue Dienste entwickelt werden müssen, die von den Teilnehmern akzeptiert werden, ist jedem Netzbetreiber klar. Allerdings ist das Auffinden der neuen Dienste in unbekannten Netzen problematisch. Hiermit beschäftigen sich in Europa unterschiedliche Initiativen und Projekte. Das Nomad-Projekt (Integrated Networks for Seamless and Transparent Service Discovery) beschäftigt sich speziell mit der mobilen Unterstützung neuer Dienste. Im Vordergrund steht der Begriff des Nomadic User, der zum einen die unterbrechungsfreie Kommunikation bei der Verwendung von Sprache und Video benötigt und zum anderen das verfügbare Netz sowie die darin angebotenen Dienste erkennen muß. Ein User Profiling zum Erkennen der Anforderungen des Benutzers ist ebenfalls in der Entwicklung.

Erste öffentliche UDDI-Registries wurden von HP, IBM, Microsoft und SAP bereits umgesetzt. Auf diesen Servern können derzeit Webservices registriert und öffentlich bekannt gemacht werden. Aufgrund der zahlreichen UDDI-Server muß der entsprechende Webdienst auf alle UDDI-Server übertragen werden. Dies geschieht durch eine Datenbankreplikation.

Alle herkömmlichen Verfahren sind nicht nur für den Anbieter sehr aufwendig, sondern auch für den Benutzer verwirrend. Derzeit kommt es sogar vor, daß nicht alle UDDI-Repositories dieselben Webservices anbieten. Das kann nur vermieden werden, wenn jeder Anbieter seine Webdienste auf einem zentralen Server registriert und einrichtet. Von einer guten Skalierbarkeit kann dann aber nicht mehr die Rede sein.

Das Nomad-Projekt hat deshalb die Erweiterung eines UDDI-Repositories um ein LDAP-Backend entwickelt, wodurch die Defizite hinsichtlich der Skalierbarkeit ausgeglichen und auch die Performance sowie die Sicherheit erhöht wurden. Gerade das Problem der verteilten Datenhaltung konnte so behoben werden. Alle Daten sind über einen zentralen Zugangspunkt adressierbar und können dennoch verteilt vorliegen. Sollte ein Benutzer nach einem bestimmten Dienst suchen, so weiß er, wie er diesen Dienst adressieren kann. *(bk)*