

## Service Brokerage - Ein benutzerorientierter Ansatz

### Service Brokerage

Service Brokerage ist ein zentraler Dienst in Ambient Intelligence Umgebungen, bei dem verschiedene Dienste von unterschiedlichen Software-Komponenten innerhalb der intelligenten Umgebung angeboten werden. Zusätzlich zu der Möglichkeit, einen bestimmten Dienst aus den verfügbaren Diensten auszusuchen ist es dabei ebenso wichtig, dass Dienste pro-aktiv vom System angeboten werden, die auf die jeweilige Situation des Benutzers zugeschnitten sind. Das Zentrum für Graphische Datenverarbeitung ZGDV e.V. hat innerhalb der Projekte MAP und servingo einen Service Broker entwickelt. In beiden Projekten basierte die Lösung auf dem Semantic-Web-Ansatz zur Beschreibung von Diensten. Im Folgenden geben wir einen kurzen Überblick über die in diesen Projekten zur Verfügung gestellte Komponente.

### Der Broker-Agent in MAP

Das Projekt MAP stellt eine Agentenplattform zur Verfügung, die entweder als persönliches Portal, die sogenannte Homepage, oder als Firmenportal konfiguriert werden kann. Der Broker-Agent ist eine in die Agentenplattform

integrierte Komponente, die für das Brokering der Dienste, die von anderen Agenten der Plattform angeboten werden, zuständig ist. Abbildung 1 fasst die Situation mit einem Broker-Agenten auf einer Homepage zusammen.

Damit der Broker-Agent nahtlos Agenten lokalisieren und aufrufen kann, wurde ein generisches Modell zur Verfügung gestellt, welches als Basis zur Beschreibung von Agenten und ihrer Merkmale benutzt werden konnte. Das Modell unterstützte den Broker-Agenten bei der Bestimmung der zur Ausführung benötigten Umgebung und Ressourcen, beim Erwerb des Source Codes und schließlich bei der Erzeugung einer persönlichen Instanz des Agenten. Weiterhin beschafften solche Beschreibungen Informationen über die von den Agenten angebotenen Dienste in OWL-S, nachverfolgte Aufgaben, die vom Benutzer an bestimmte Agenteninstanzen delegiert wurden und definierte Zugangsrechte zur Überwachung der Dienste. Die bereitgestellte Benutzerschnittstelle besteht im Wesentlichen aus drei Flächen, damit der Nutzer schnell und ohne viel Aufwand durch das System navigieren kann (siehe Abbildung 2). Sie

ist webbasiert, so dass der Nutzer Dienste unabhängig vom verwendeten Gerät über das Web aufrufen kann. Dazu muss lediglich ein Browser installiert sein.

Das Hauptbild erlaubt dem Benutzer einen Überblick über die delegierten Aufgaben und deren Fortgang. Gleichzeitig dient es als Haupt-Bedienfeld, das dem Benutzer erlaubt, das Erzeugen, Löschen oder Abändern von Aufgaben zu initiieren. Die tatsächliche Erzeugung und Abänderung von Aufgaben wird in Nebenbildern ausgeführt, so dass die Informationen, die auf dem Hauptbild angezeigt werden, leserlich bleiben. Laufende oder beendete Tasks werden mit einem Icon aufgelistet, das den jeweiligen Status anzeigt. Die Oberfläche auf der neue Aufgaben erzeugt werden, erlaubt es dem Benutzer einen gewünschten Dienst auszuwählen und eine Kennzeichnung einzugeben, die als persönliche Kennung für die Aufgabe dient. Eine Kombi-Box führt die dem Nutzer zu Verfügung stehenden Dienste auf. Jedem Dienst ist eine kurze Zusammenfassung seiner Möglichkeiten zugeordnet, die der Nutzer ansehen kann, wenn er auf den Lupen-Icon drückt.

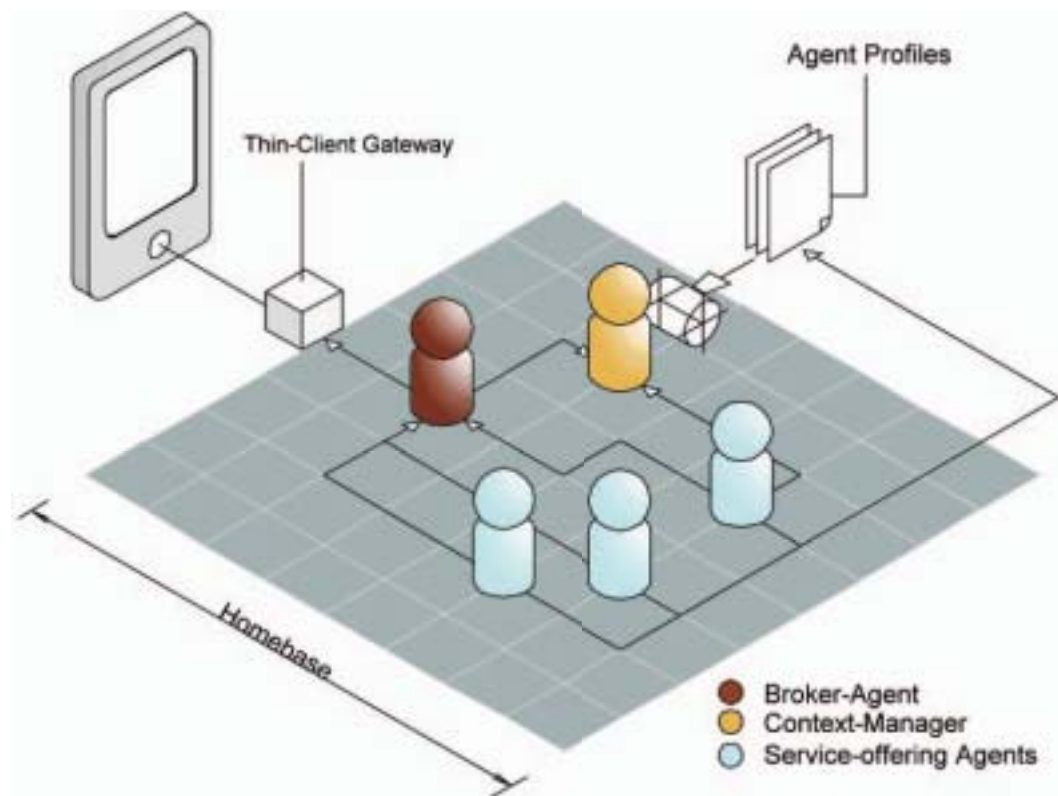


Abb.1: Der Broker Agent als Dienste-Schnittstelle.  
Fig.1: The broker agent as the service gateway.

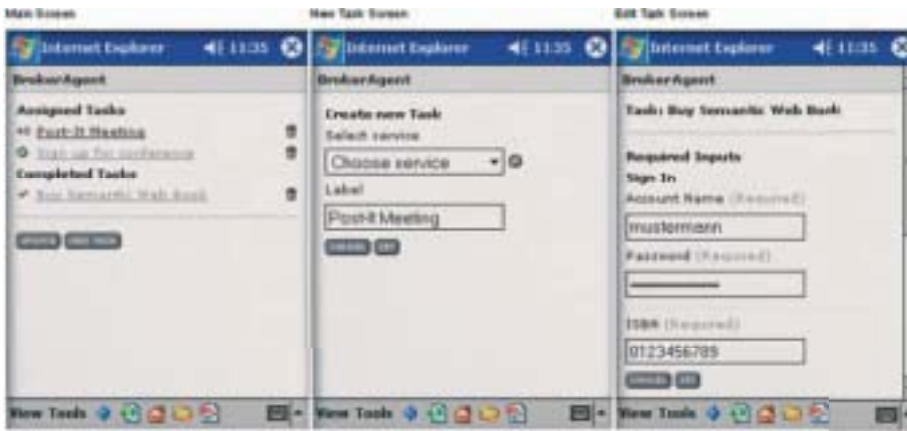


Abb.2: Screenshots der Broker-Agent-Schnittstelle.

Fig.2: Screenshots of the broker agent user interface.

Möchte der Benutzer Aufgaben editieren oder Zwischenergebnisse ansehen, benutzt er das Edit-Task-Feld. Auf Basis des OWL-S Dienstmodells werden die bis zu diesem Zeitpunkt berechneten Ergebnisse und die erforderlichen Input-Parameter in einem Formular dargestellt. Danach kann der Nutzer das Formular versenden und die Ergebnisse dieses Schritts ansehen, sobald der entsprechende Agent das Formular verarbeitet hat.

Die bereitgestellte Lösung erleichtert die Interaktion mit der Homebase, da sie Aufgaben an Agenten delegiert. Sie verringert die Unkosten für die Anschaffung von User Interfaces für mobile Agenten. Der Nutzer wird durch eine Standardmethode, mit verschiedenen Agenten zu interagieren, entlastet. Er kann sofort den Stand seiner Arbeit erkennen und die entsprechenden Dienste verwenden, ohne sich mit der Verwaltung und Konfigurierung der diensteanbietenden Software-Komponenten beschäftigen zu müssen. Die Hauptinnovationen bestanden aus:

- Der kontextbasierten Wissensvermittlung über die erhältlichen Dienste,
- einem neuen OWL-S-Fundament zur Agentenkommunikation und einer konkreten Syntax für den Gebrauch von Diensten und der die Lieferung der Ergebnisse und
- einen ersten Ansatz für einen mobilen Zugang zu Diensten zum Zweck der Aufgabendelegierung.

### Der Service Broker in servingo

Das Projekt servingo, ein Forschungs- und Entwicklungs-Projekt, welches zum Teil von Bundesministerium für Wirtschaft und Arbeit finanziert wird, hat zum Ziel, einen moder-

nen, informativen und unterhaltsamen Führer durch die Fußballweltmeisterschaft 2006 auf mobilen Geräten zur Verfügung zu stellen. Das Projekt servingo unterstützt die Mobilität von Besuchern und Teilnehmern bei der Fußballweltmeisterschaft indem es gezielte und unterhaltsame Informations- und Kommunikationsmöglichkeiten über ein Web-Portal anbietet. Dadurch trägt es zu einem effizienteren und reicheren Erleben der Gäste bei.

Das servingo-Portal besteht aus mehreren dienst anbietenden Portlets. Die Rolle des Service-Brokers als wesentlichen Bestandteil des Portals ist es, die Integration der Portlets ins Portal zu erleichtern und dem Benutzer Mittel zur Verfügung zu stellen, mit denen er die entsprechenden Dienste suchen und Zugang zu ihnen bekommen kann. Um dies zu ermöglichen, benutzt der Service-Broker die Beschreibungen der Dienste in OWL-S. Die Dienstbeschreibungen können mit Hilfe eines speziellen Editors erzeugt werden, der außerdem Mechanismen zur Verfügung stellt, mit denen Portlets spezifiziert und implementiert werden und die Dienste an eine erweiterbare Diensthierarchie angebinden werden können. Der Editor kann auch verwendet werden, um aus den momentan verfügbaren Diensten neue komplexe Dienste zu bilden.

Die Diensthierarchie wird innerhalb des Portals benutzt, um die Navigationshierarchie zu erstellen. Indem ein Dienst aus der Navigationshierarchie ausgewählt wird, wird der Broker veranlasst das zuständige Portlet zu finden und einen Ausführungsvorgang des entsprechenden Dienstes, nämlich einen Prozess, anzustoßen. Wählt der Nutzer einen anderen

Dienst aus der Navigationshierarchie, so wird der laufende Prozess minimiert und der Liste der offenen Prozesse hinzugefügt, mit der Option ihn zu reaktivieren und fortzuführen. Ein einmaliges Merkmal des Service Brokers in servingo ist die kontextbezogene Dienstbereitstellung je nach Benutzer, Sitzung und Prozesskontext. Wenn beispielsweise der laufende Prozess einen Point of Interest als Zwischenergebnis bringt, liefert der Broker einige schnelle Links, sogenannte »quinks« (quick links), um diesen Output als Input an die Dienste weiterzuleiten, die einen Ort benötigen, um starten zu können. Der Service Broker stellt so einen automatischen Mechanismus zur Verfügung, um verfügbare Dienste zu verketten.

In Abbildung 3 geben die horizontale Navigation im grünen Feld und das Menü links die Navigationshierarchie, die vom Broker erstellt wurde, wieder. Die Liste 'last visited' enthält die Liste der offenen Prozesse. Im mittleren Bereich, unterhalb des Menüpfads des momentan aktiven Prozesses, folgt auf den vom Portlet kommenden Inhalt, der im laufenden Prozess aktiviert wurde, die Liste der vorgeschlagenen quinks.

### Service Brokerage

- Bereitstellung verschiedener Dienste und Software-Komponenten innerhalb einer intelligenten Umgebung
- Anpassung auf die individuelle Situation des Nutzers durch vom System pro-aktiv angebotene Dienste
- Einsatz unter anderem in den Projekten MAP und servingo

### Kontakt

**Dr. Dirk Balfanz**  
Zentrum für Graphische Datenverarbeitung  
ZGDV, Darmstadt  
E-Mail: [dirk.balfanz@zgdv.de](mailto:dirk.balfanz@zgdv.de)

**Saied Tazari**  
Zentrum für Graphische Datenverarbeitung  
ZGDV, Darmstadt  
E-Mail: [saied.tazari@zgdv.de](mailto:saied.tazari@zgdv.de)

## Service Brokerage - A User-Oriented Approach

Service brokerage is a central issue in Ambient Intelligence environments, where several services are offered by different software components in the intelligent environment. In addition to the possibility for the user to choose a certain service from the available and accessible services, it is also important to be able to offer services proactively depending on the situation in which the user finds himself. The Computer Graphics Center ZGDV e.V has developed a service broker within the framework of the projects MAP and servingo. In both projects, the solution was based on the Semantic Web approach for describing services. In the following, we give a short overview of the provided component in these projects.

### The Broker Agent in MAP

The project MAP provided an agent platform that could be configured either as a personal portal, the so-called homebase, or as the corporate portal. The broker agent was an integral component of the agent platform responsible for brokering the services offered by the other agents available on the platform. Figure 1 summarizes the situation on a homebase with the broker agent.

In order for the broker agent to seamlessly locate and invoke agents, a generic model was provided that could be used as the basis for describing agents and their characteristics. The model supported the broker agent in

determining the environment and resources required for execution, acquiring the source code, and finally creating a personal instance of the agent. Furthermore, such descriptions provided information about the services offered by the agent in OWL-S, tracked tasks delegated by the user to concrete agent instances, and defined access rights necessary to control service usage.

The user interface provided consists mainly of three screens. So the user can navigate through the system quickly and effortlessly (see figure 2). It is web-based so that the user can access services via the web regardless of the device used – with the minimal requirement of a preinstalled browser.

The main screen allows the user to have an overview of delegated tasks and their progress. At the same time, it functions as the main control panel allowing the user to initiate the creation, deletion or modification of tasks. The actual creation and modification of tasks is handled in sub-screens so that the information displayed on the main screen remains legible. Ongoing and completed tasks are listed with an icon depicting the respective state.

The screen for creating new tasks lets the user select a service he or she wishes to use and enter a label, which serves as a personal identifier for the task. A combo box lists the available services from which the user can choose. Each service is associated with a short résumé of its capabilities, which can be viewed by clicking on the magnifier icon. To edit tasks and view intermediate results, the user consults the edit task screen. Based on the OWL-S service model, the results computed thus far and the required input parameters will be displayed in a form. When done, the user can submit the form and view the result of the execution of that step once the respective agent has processed the form. The solution provided facilitates the interaction with the homebase in terms of delegating tasks to software agents. It mitigates the overhead of providing UIs for mobile agents. The user is relieved by a standard way of interacting with different agents, can recognize the status of his work at a glance, and utilize the services without dealing with the administration and configuration of the service-offering software components.



Abb.3: Eine Momentaufnahme des servingo-Portals auf Desktop-Computern.  
Fig.3: A snapshot of the servingo portal on desktop computers.

The major innovations consisted of:

- The context-based knowledge imparting about the available services,
- a new OWL-S grounding for agent communication and concrete syntax for the utilization of services and delivery of the results, and
- a first approach for mobile access to services in the light of task delegation.

### The Service Broker in servingo

The aim of servingo, a research and development project, partially funded by the German Federal Ministry of Economics and Labor, is to provide a modern, informative, and entertaining guide on mobile devices through the FIFA World Cup 2006. By supporting the mobility of visitors and participants at the FIFA World Cup 2006 and offering targeted and entertaining information and communication options through a web portal, servingo will promote a more efficient and richer experience to all guests.

The servingo portal is comprised from several service-offering portlets. The role of the service broker as an integral part of the portal is to facilitate the integration of the portlets into the portal and to provide means for the users to find and access the appropriate services. To do so, the service broker uses the descriptions of the services in OWL-S. The service descriptions can be generated using a special editor that additionally provides mechanisms to specify the portlets implementing them and to bind the services to an extensible service hierarchy. The editor can also be used in order to compose new complex services using the currently available services.

The service hierarchy is used within the portal to provide the navigation hierarchy. Choosing a service from the navigation hierarchy will cause the broker to find the responsible portlet and initiate an execution instance of the corresponding service, called a process. By choosing another service from the navigation hierarchy, the current process will be minimized and added to the list of open processes with the possibility to again activate and proceed with it.

A unique feature of the service broker in servingo is the context-sensitive service provision, based on the user, session, and process context. For instance, if the current process returns a point of interest as an intermediate result, the broker provides some quick links,

the so-called quinks, to pass that output as an input to those known services that need a location in order to start. This way, the service broker provides an automatic mechanism for interlacing available services.

In figure 3, the horizontal navigation in the green area and the menu on the left side display the navigation hierarchy provided by the broker. The list titled 'last visited' contains the list of open processes. In the middle area, under the menu path of the currently active process, the content delivered by the portlet, activated within the current process, is followed by the list of suggested quinks.



Abb.4: Dienste in servingo werden kontextbezogen bereitgestellt.

Fig.4: Services in servingo are provided context-sensitive.

### Service Brokerage

- Several services offered by different software components in an intelligent environment
- Support of the user depending on his individual situation via proactively offered services
- Already implemented in the projects MAP and servingo

### Contact

**Dr. Dirk Balfanz**  
**Computer Graphics Center ZGDV,**  
**Darmstadt, Germany**  
**E-mail: dirk.balfanz@zgdv.de**

**Saied Tazari**  
**Computer Graphics Center ZGDV,**  
**Darmstadt, Germany**  
**E-mail: saied.tazari@zgdv.de**