

Abstrakte Dienstklassen zur Realisierung mobiler Prozesse

Christian P. Kunze, Sonja Zaplata und Winfried Lamersdorf

Universität Hamburg, Department Informatik
Vogt-Kölln-Straße 30, 22527 Hamburg, Deutschland

Zusammenfassung. Mobile Systeme sind in ihrer Leistungsfähigkeit beschränkt und können daher besonders von zusätzlichen Dienstangeboten in ihrer jeweiligen Umgebung profitieren. Das gilt insbesondere für die dynamische, kontextbasierte Zusammenarbeit verschiedener Geräte bei der Ausführung langlebiger und vielschichtiger (Geschäfts-) Prozesse. Deshalb wird in diesem Beitrag vorgeschlagen, abstrakte Prozessbeschreibungen zwischen Geräten weiterzugeben, um diese dann als „mobile Prozesse“ dezentral auszuführen. Dabei wird eine möglichst technologieunabhängige Integration von heterogenen Anwendungen und Diensten angestrebt, da zur Entwurfszeit der Prozesse in der Regel noch nicht bekannt ist, welche Art von Diensten und Anwendungen im Umfeld des ausführenden (mobilen) Geräts konkret zur Verfügung steht. Als Realisierungsarchitektur wird eine dreischichtige Dienstabstraktion eingeführt und gezeigt, wie durch die Nutzung „abstrakter Dienstklassen“ die Ausführung mobiler Prozesse flexibilisiert werden kann, während gleichzeitig die Komplexität der Prozessbeschreibung möglichst gering gehalten wird.

1 Einleitung

Dezentralisierung und ubiquitäre Verfügbarkeit von Informationen und Dienstleistungen sind durch den technischen Fortschritt in der Prozessor- und Netzwerktechnologie stetig gestiegen [3]. Eine zentrale Stelle nimmt dabei die Mobilität der Benutzer, der Geräte und des Anwendungscodes ein, wobei Benutzer mobiler Systeme meist nur dann optimal von diesen unterstützt werden können, wenn das häufig vorhandene Ungleichgewicht von benötigten und angebotenen Ressourcen überwunden wird. [5].

Um in einem solchen Umfeld komplexe und stark verteilte Anwendungen zu verwirklichen, können diese als anwendungsorientierte (Business-) Prozesse realisiert werden, die zwischen verschiedenen Geräten migrieren und so die Potentiale mehrerer Geräte nutzen können. Eine besondere Herausforderung liegt dabei in der Notwendigkeit einer möglichst technologieunabhängigen Beschreibung der Aktivitäten im Prozess sowie im Einbinden entsprechender a priori unbekannter Dienste zur Laufzeit. Dies ist deshalb wichtig, da der mobile Prozess zwischen Geräten mit sehr heterogenen Fähigkeiten migrieren kann und deshalb zum Zeitpunkt der Definition des Prozesses nicht festgelegt werden kann, welche Art der Implementierung einer Aktivität für das jeweils ausführende Gerät erforderlich ist. Zudem hat dieses Vorgehen zum Vorteil, dass durch eine späte Bindung sowohl der verwendeten Technologie als auch der Instanz des Dienstes eine flexible Ausführung der Aktivitäten erhalten bleibt.

Dementsprechend stellt diese Arbeit den im Rahmen des Projekts *Distributed Environment for Mobility-Aware Computing* (DEMAC) entwickelten Ansatz der *abstrakten Dienstklassen* und dessen Konkretisierung in einer Beschreibungssprache für mobile Prozesse vor. Zudem wird gezeigt, wie diese Dienstklassen während der Ausführung mit Hilfe von Kontextinformationen zu konkreten Diensten aufgelöst

und dann generisch angesprochen werden können. Zunächst aber wird das Konzept der „abstrakten Dienstklasse“ eingeführt und deren Einsatz in einer Middleware zur Unterstützung mobiler Prozesse aufgezeigt. Schließlich wird auch ein kurzer Einblick in laufende Implementierungsarbeiten vermittelt.

2 Ausführung komplexer Aufgaben durch mobile Prozesse

Um mobilen Geräten eine Kooperation zur Erledigung komplexer Aufgaben zu erlauben, ist zum einen eine genaue Beschreibung der auszuführenden Anwendungslogik, der Reihenfolge der Ausführung und der involvierten Daten nötig. Diese Elemente können gemeinsam als (Geschäfts-)Prozess modelliert werden, der um Möglichkeiten zur verteilten Ausführung erweitert und auf die speziellen Anforderungen mobiler Systeme und ihrer sich häufig ändernden Umwelt angepasst wird. So sollte zum einen die resultierende Beschreibung kompakt formulierbar und einfach ausführbar sein, um die Inanspruchnahme von Ressourcen möglichst gering zu halten. Zum anderen sollte die Integration von Benutzerinteressen in Form von nicht-funktionalen Parametern berücksichtigt werden. Die Beschreibung der Aktivitäten selbst sollte möglichst abstrakt gehalten werden, um durch die Möglichkeit einer späten Bindung eine flexible Zuweisung von Dienstinstanzen während der Prozessausführungszeit zu realisieren.

2.1 Mobile Prozesse

Grundidee mobiler Prozesse ist es, komplexe Aufgaben nicht allein auf einem einzelnen mobilen Gerät zu bearbeiten, sondern diese – zusammen mit dem aktuellen Ausführungszustand – an andere Geräte weitergeben zu können. Dies ist immer dann angebracht, wenn die Leistungsfähigkeit eines mobilen Geräts nicht ausreicht oder ihm keine Dienste mehr zur Verfügung stehen, um anstehende (Teil-) Aufgaben zu erledigen. Entsprechend ist ein mobiler Prozess wie folgt definiert:

Ein mobiler Prozess ist eine Abfolge zusammengehöriger (z.T. entfernter) Dienste, deren vollständige Bearbeitung i.a. eine längere Zeitspanne andauert und sich dabei auch über mehrere Geräte erstrecken kann. Die Ergebnisse eines derartigen Prozesses bestehen aus der Summe aller Effekte, die der Initiator im Zuge der Bearbeitung des Gesamtprozesses erwartet. [2, Seite 33]

Randbedingungen für die Art und Weise der Ausführung können in mobilen Prozessen durch die zusätzliche Spezifizierung von nicht-funktionalen Aspekten vorgegeben werden. Die Ausführung des Gesamtprozesses kann dabei prinzipiell beliebig heterogene Dienste, Prozessteilnehmer und Geräte umfassen und dynamisch einbinden, um so deren Potenziale jeweils optimal zu nutzen.

2.2 Integration von Diensten zur Ausführung von Prozessaktivitäten

Aus der Anforderung einer technologieunabhängigen Prozessbeschreibung, die für a priori unbekannte und heterogene Ausführungsumgebungen geeignet sein soll, ergibt sich die Notwendigkeit, für die Aktivitäten eines solchen Prozesses eine möglichst abstrakte und somit technologieunabhängige Beschreibung zu wählen. Insbesondere durch die mögliche Migration des mobilen Prozesses ist nicht vorhersagbar, welche Fähigkeiten und Technologien dem Gerät zur Verfügung stehen, das letztendlich den Dienst ausführen wird.

Abstrakte Dienstklassen Das Konzept der *abstrakten Dienstklasse* ist eine konsequente Fortsetzung der abstrakten (aber technologieabhängigen) Schnittstellenbeschreibungen von Middleware-Plattformen zum Dienstaufwurf in verteilten

Systemen. Es umfasst, ähnlich wie zum Beispiel die *tModel-Elemente* in einem UDDI-Verzeichnis, lediglich die gemeinsamen Meta-Daten, die benötigt werden, um später den gewünschten Dienst finden und ausführen zu können (vgl. [6]). Allgemein können abstrakte Dienstklassen folgendermaßen definiert werden:

Eine abstrakte Dienstklasse subsumiert alle technologieabhängigen Beschreibungen eines Dienstes unter einer gemeinsamen und technologieunabhängigen Referenz. Sie wird durch eine Signatur definiert, die aus einem eindeutigen Bezeichner und einer Parameterliste besteht.

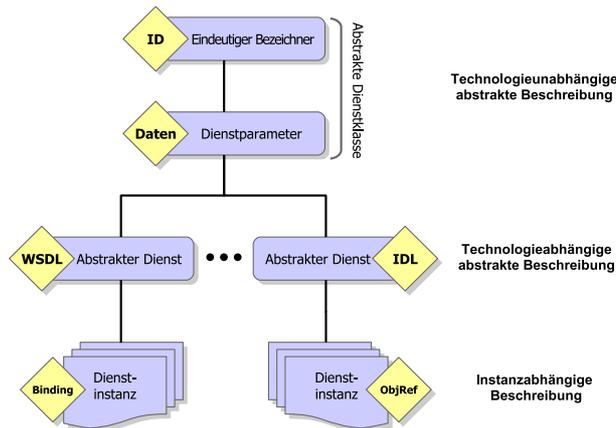


Abb. 1. Verwendung abstrakter Dienstklassen

Dienstklassen abstrahieren gleichartige Dienstinstanzen auf drei Ebenen, um sie so für mobile Prozesse nutzbar zu machen (siehe Abb. 1): Auf der untersten Ebene der *instanzabhängigen Beschreibungen* werden die Charakteristika und Referenzen beschrieben, die notwendig sind, um eine spezifische Dienstimplementierung anzusprechen (dies wären zum Beispiel die Binding-Informationen bei Verwendung von Web Services). Darüber liegt dann die Ebene der *technologieabhängigen abstrakten Beschreibungen*, welche die Schnittstellendefinition der Dienste in allgemeiner aber für die jeweilige Middleware spezifischer Form enthält (in Corba-Umgebungen befindet sich beispielsweise das zum Dienst gehörende IDL-Dokument auf dieser Ebene). Die Ebene der abstrakten Dienstklassen bzw. der *abstrakten und technologieunabhängigen Beschreibungen* fasst schließlich die Gemeinsamkeiten der darunter liegenden Schicht in einer einheitlichen Beschreibung zusammen. Dabei wird nicht nur ein eindeutiger Bezeichner vergeben, sondern auch die Liste der Parameter mit einbezogen. Dies ist deshalb notwendig, da in der Prozessbeschreibung im Zuge der Definition des Datenflusses angegeben werden muss, welche Variablen des mobilen Prozesses an welche Parameter der Dienstklasse gebunden werden müssen.

Auflösung abstrakter Dienstklassen zur Laufzeit Da abstrakte Dienstklassen eine mehrfache Abstraktion einzelner Dienstinstanzen bilden, muss auch das Auflösen der Dienstklasse zur Laufzeit mehrere Schritte umfassen. Dabei werden im Auswahlprozess neben der funktionalen Referenz der Dienstklasse auch nicht-funktionale Parameter des mobilen Prozesses und des ausführenden Gerätes berücksichtigt. So wird im ersten Schritt durch die vom Gerät unterstützten Middleware-Plattformen die Menge der in Frage kommenden technologieabhängigen Dienstbeschreibungen bestimmt. Danach wird unter Verwendung der im Prozess beschriebenen nicht-funktionalen Aspekte, wie zum Beispiel Kosten- oder Qualitätsbedingungen (QoS), eine konkrete Instanz zur Ausführung der Aktivität ausgewählt. Diese wird anschließend an die durch die Parameterliste der Dienstklassenbeschreibung definierten Variablen des mobilen Prozesses gebunden und ausgeführt.

3 Integration abstrakter Dienstklassen in eine Middleware für mobile Prozesse

Die Integration abstrakter Dienstklassen in eine Middleware für mobile Prozesse geschieht auf zwei Ebenen: Zum einen auf der *Ebene der Prozessbeschreibung*, auf der die Referenz der enthaltenen Aktivitäten auf die dahinter stehende Dienstleistung durch abstrakte Dienstklassen realisiert wird, zum anderen auf *Ebene der Prozessausführung*, auf der diese dann während der Ausführung des mobilen Prozesses in konkrete Dienstinstanzen aufgelöst werden.

3.1 Beschreibung abstrakter Dienstklassen in mobilen Prozessen

In Anlehnung an XPDL [4] wurde im Rahmen der hier dargestellten Arbeiten die Prozessbeschreibungssprache *DPDL (DEMAC Process Description Language)* zur Beschreibung mobiler Prozesse entwickelt (vgl. [2]).

Durch die beschriebene Entkopplung von konkreten Technologie- und Implementierungsdetails kann die Einbindung der aufzurufenden Dienste innerhalb der Prozessbeschreibung auf die Definition der auszuführenden Aufgabe und die Bindung der aktuellen Parameter beschränkt werden. Listing 1 zeigt ein Beispiel hierfür: Der funktionale Inhalt einer generischen Anwendung (*Drucken*) wird durch eine Referenz auf einen eindeutigen Bezeichner vom Typ UUID und die formalen Parameter der abstrakten Dienstklasse identifiziert, welche alle zur Verfügung stehenden Anwendungen umfasst, die zur Erledigung der anstehenden Aufgabe ausgelegt sind. Der syntaktische Teil wird dabei durch die Spezifikation der Ein- und Ausgabeparameter sowie deren Datentypen festgelegt – in diesem Fall ein Dokument vom Typ *Text*. Durch eine Referenz auf die in der Prozessbeschreibung enthaltenen Variablen werden die entsprechenden Prozessdaten an die generische Anwendung gebunden.

Listing 1. Prozessbeschreibung einer generischen Aktivität

```
<Application Id="Drucken">
  <UUID>12345678901234567890123456789012</UUID>
  <FormalParameters>
    <FormalParameter Id="Dokument" Index="1" Mode="IN">
      <DataType>
        <DeclaredType Id="Text"/>
      </DataType>
    </FormalParameter>
  </FormalParameters>
</Application>
```

3.2 Realisierung abstrakter Dienstklassen

Die im Projekt DEMAC entwickelte Middleware-Plattform zur Integration mobiler Prozesse besteht, grob betrachtet, aus Basisdiensten zur ereignis- und nachrichtenorientierten Kommunikation, einer Komponente zur Interpretation und Ausführung mobiler Prozesse (*Prozessdienst*) und einem *Kontextdienst*, welcher ein *föderiertes Datenmodell* zur Beschreibung und Nutzung von Kontextinformationen enthält [2].

Der Prozessdienst ist modular aufgebaut, um sich den Fähigkeiten des jeweiligen mobilen Gerätes anpassen zu können und enthält als zentrale Komponente eine Workflow-Engine zur Interpretation und Ausführung mobiler Prozesse. Ein um zusätzliche Zustände erweiterter Prozesslebenszyklus erlaubt dem Prozessdienst, einen laufenden Prozess sicher an andere Teilnehmer zu migrieren (vgl. [2]). Zusätzliche Plug-In-Module ermöglichen es zudem, weitere Funktionalitäten, wie die Unterstützung von Transaktionen oder die Integration von Sicherheitsmechanismen, einzufügen.

Mobile Prozesse können entweder von lokalen Anwendungen an den Prozessdienst übergeben werden oder über die Kommunikationsschnittstelle von anderen Geräten

in die Middleware migrieren. Die Ausführung des Prozesses, also das sukzessive Auflösen und Ausführen der Aktivitäten, wird dabei durch die im Prozessdienst enthaltene Workflow-Engine gesteuert. Die eigentliche Suche nach konkreten Dienstinstanzen wird vom Kontextdienst mit Hilfe der abstrakten Dienstklassen sowie den nicht-funktionalen Parametern des Prozesses und des Gerätes vorgenommen. Der Kontextdienst enthält hierzu neben dem föderierten Datenmodell auch ein *verteiltes Dienstverzeichnis* (Registry) und eine Komponente zum Erzeugen generischer Wrapper-Objekte für Dienstinstanzen (Fabrik). Damit stellt der Kontextdienst die zentrale Instanz dar, um Informationen über den Kontext des Gerätes zu sammeln und zu verwalten (vgl. auch [2]).

Der Kontextdienst erstellt Wrapper-Objekte auf Basis des lokalen Kontextwissens oder, wenn nötig, aus den Ergebnissen einer Suche nach der Dienstklasse auf Geräten in seiner Umgebung. Dazu werden die nicht-funktionalen Parameter um die Informationen über die vom Gerät beherrschten Technologien ergänzt und eine entsprechende Anfrage an das verteilte Dienstverzeichnis gestellt. Aufgrund der hohen Dynamik mobiler Umgebungen und der beschränkten Ressourcen mobiler Geräte werden – sollte kein lokaler Dienst zur Verfügung stehen – die Anfragen direkt an die Verzeichnisdienste anderer Geräte weitergereicht und dort ausgewertet. Damit wird erreicht, dass die Suche nach passenden Diensten einen möglichst geringen lokalen Ressourcenverbrauch verursacht, da keine Detaildaten, die zur Auswahl nötig wären, zwischen den Geräten ausgetauscht werden müssen. Zudem ergibt sich aus der Parallelisierung der Anfragebearbeitung ein zeitlicher Vorteil bei der Suche nach passenden Dienstbringern. Die Auswahl auf Seiten des Dienstanfragenden beschränkt sich danach nur noch auf die Selektion eines Dienstes aus der Menge der prinzipiell passenden Dienstbringer. Hierbei können recht einfache Strategien, wie zum Beispiel *first-come-first-served*, Verwendung finden, da die grundsätzliche Eignung ja schon durch die nicht-funktionalen Parameter innerhalb der Anfrage festgelegt wurde.

Die Antworten auf die Suchanfragen enthalten im Kopf der Nachricht eine eindeutige Referenz auf die Anfrage und die Bezeichnung der Technologie, mit dem der Dienst realisiert wurde. Im Nachrichtenkörper ist zusätzlich ein Dokument mit den technologiespezifischen Details zum Dienstaufwurf enthalten. Bei einem Web Service ist dies beispielsweise ein WSDL-Dokument mit entsprechenden Binding-Informationen. Diese Angaben werden dann von der Fabrik-Komponente in ein entsprechendes Wrapper-Objekt überführt und als Ergebnis der vom Prozessdienst initiierten Dienstauflösung zurückgegeben.

An dem Wrapper-Objekt wird anschließend eine generische Schnittstelle mit den an die Aktivität gebundenen Datenfeldern als Parameter aufgerufen. Damit ist die Auflösung und Ausführung der abstrakten Dienstklasse unter Berücksichtigung der nicht-funktionalen Parameter des Prozesses und des Gerätes beendet. Sollten während der Auflösung oder Ausführung Fehler aufgetreten sein, so kann eine erneute Auflösung der abstrakten Dienstklasse versucht werden oder aber der gesamte Prozess wird zu einem Gerät mit anderen Fähigkeiten (z.B. alternative Technologien, mit denen weitere Dienste angesprochen werden können) migriert.

4 Analyse bestehender Prozessmodelle und Infrastrukturen

Bei der Betrachtung bestehender Konzepte müssen sowohl Sprachen zur Beschreibung als auch Techniken zur Ausführung langlebiger benutzerzentrischer Prozesse berücksichtigt werden. Zudem müssen Arbeiten im Bereich der Middleware-Plattformen für mobile Systeme untersucht werden, welche sich mit der Verarbeitung von Kontextdaten und der Ausführung von mobilen Anwendungen beschäftigen (eine detaillierte Betrachtung verwandter Arbeiten findet sich in [2]).

Zusammengefasst kann man sagen, dass die meisten Prozessbeschreibungssprachen, wie beispielsweise *BPEL4WS*, *WSCI* oder *JPDL*, nicht abstrakt und technologieunabhängig genug sind, um mobile Prozesse beschreiben zu können. Jedoch bietet die Beschreibungssprache *XPDL* [4] als Meta-Prozessbeschreibungssprache eine gute Basis zur Formulierung mobiler Prozesse, muss jedoch um zusätzliche Aspekte, wie die Beschreibung des Prozesszustandes, oder um Möglichkeiten zur Definition von nicht-funktionalen Aspekten erweitert werden.

Zur Unterstützung des Mobile Computings existieren eine Reihe von Middleware-Ansätzen, wie zum Beispiel *Nexus*, das Projekt *m3*, *Gaia* oder *PCOM*. Da diese jedoch nicht für mobile Prozesse und kontextbasierter Kooperation entworfen wurden, sind sie nur bedingt in einer entsprechenden Middleware einsetzbar. Sie konzentrieren sich eher auf die Unterstützung geschlossener Anwendungen in mobilen Umgebungen. Beim Einsatz von mobilen Prozessen hingegen diffundiert die Anwendungslogik zu einem gewissen Maß in die Middleware und benötigt hierfür eine eigene und neuartige Unterstützung.

Andere vergleichbare Ansätze zu der in diesem Beitrag beschriebenen Infrastruktur können im Bereich der mobilen Agenten identifiziert werden [1]. Im Gegensatz zu Agenten enthalten mobile Prozesse jedoch lediglich Metadaten und keinen ausführbaren Quelltext. Zudem ist bei mobilen Prozessen weder soziales noch autonomes Verhalten erforderlich oder unbedingt erwünscht.

5 Zusammenfassung

Die in diesem Beitrag vorgestellte Klasse von Anwendungen des Mobile Computing mit kontextbasierter Zusammenarbeit stellt besondere Anforderungen an eine entsprechend erweiterte Middleware-Unterstützung. Hierzu werden vor allem abstrakte und technologieunabhängige Methoden und Techniken benötigt, die es ermöglichen, mit der Heterogenität und Dynamik fortschrittlicher mobiler Anwendungsumgebungen möglichst gut umzugehen. Zu diesem Zweck wurden die Konzepte des mobilen Prozesses und der abstrakten Dienstklassen eingeführt, um so eine Kooperation zwischen a priori unbekanntem und mit unterschiedlichen Fähigkeiten ausgestatteten Geräten angemessen modellieren und geräteunabhängig unterstützen zu können. Dabei haben abstrakte Dienstklassen den Vorteil, nicht nur von einzelnen Dienstanstanzen zu abstrahieren, sondern auch von der bei deren Ausführung zu verwendenden Technologie. Dies erlaubt eine möglichst späte Bindung – nicht nur auf Instanz-, sondern auch auf technologischer Ebene – und ermöglicht damit ein Höchstmaß an Flexibilität in der Prozessausführung auch in vorab unbekanntem bzw. dynamisch wechselnden Umgebungen.

Literaturverzeichnis

1. BRAUN, P. und W. ROSSAK: *Mobile Agents - Basic Concepts, Mobility Models, and the Tracy Toolkit*. Elsevier and Morgan Kaufmann and dpunkt.verlag, 2005.
2. KUNZE, C. P., S. ZAPLATA und W. LAMERSDORF: *Mobile Process Description and Execution*. In: *Proceedings of the International Conference on Distributed Applications and Interoperable Systems*, S. 32 – 47. Springer, 2006.
3. MATTERN, F.: *Acht Thesen zur Informatisierung des Alltags*. In: *Tagungsband zum Acatech-Symposium Computer in der Alltagswelt*, Berlin, Juni 2005.
4. NORIN, R., M. MARIN und R. SHAPIRO: *Workflow Process Definition Interface - XML Process Definition Language Version 2.0*. Spezifikation, WfMC, 2005.
5. SATYANARAYANAN, M.: *The Many Faces of Adaption*. IEEE Pervasive Computing, 3(3):4–5, 2004.
6. SCHMELZER, R., T. VANDERSYPEN, J. BLOOMBERG, M. SIDDALINGAIAH, S. HUNTING, M. QUALLS, C. DARBY, D. HOULDING und D. KENNEDY: *XML and Web Services Unleashed*. Sams Publishing, 2002.