

MUSIC macht Software fit für mobile Anwendungen

Ulrich Scholz
European Media Laboratory GmbH
Schloß-Wolfsbrunnenweg 33
69118 Heidelberg
ulrich.scholz@eml.org

Rolf Fricke
Condat AG
Alt-Moabit 91 d
10559 Berlin
rf@condat.de

Abstract: Das MUSIC Projekt unterstützt die Entwicklung von adaptiven Anwendungen für mobile Nutzer. Das Projekt baut eine Middleware und eine Open Source Entwicklungsumgebung für automatisch rekonfigurierbare Software, die sich an höchst dynamischen Nutzer- und Umgebungskontext anpasst und die für den Anwender über alle Adaptionen hinweg permanent nützlich bleibt. Über die Entwicklungsumgebung formalisiert der Applikationsentwickler die Adaptionmöglichkeiten seiner Software und die Eigenschaften der möglichen Varianten; die Erstellung der Applikationsfunktionalität selbst erfolgt in gewohnter Weise.

1 Mit MUSIC zu adaptiven, kontext-sensitiven Applikationen

Ziel des MUSIC Projekts (Self-Adapting Applications for Mobile USers In Ubiquitous Computing Environments) ist die Erstellung einer Middleware und einer Entwicklungsumgebung, die es Applikationsentwicklern ermöglicht, adaptive Anwendungen für mobile Nutzer zu schreiben. Idealerweise wird die Anwendungsfunktion herkömmlich programmiert und MUSIC übernimmt lediglich die Anpassung an die ubiquitäre Umgebung.

MUSIC ist eine für Software-Entwickler konzipierte offene, technische Plattform, die dem "jedes Gerät in jedem Netzwerk"-Paradigma folgt. MUSIC bietet eine Technologie zum Aufbau innovativer mobiler Anwendungen, die in Zukunft eine Rolle im Leben jedes einzelnen europäischen Bürgers spielen sollen. Je stärker sich der Benutzer an den Gebrauch mobiler Dienste (Services) gewöhnt, desto wichtiger, aber auch schwieriger wird es Annahmen darüber zu treffen, welche Dienste angeboten werden, wo diese verfügbar sein sollen und inwieweit sie den Erwartungen der Nutzer entsprechen.

Die große Anzahl mobiler Geräte (wie PDAs, Smart Phones, GPS, etc.) und deren funktionale Vielfalt erschweren zusätzlich den Aufbau und die Erhaltung dieser Dienste in einer benutzerfreundlichen Umgebung. Softwareentwickler müssen mit einer gewaltigen Anzahl von Fragen bezüglich Konfiguration, Betrieb, Instandhaltung und dem Management von Änderungen umgehen, um Systeme zu entwickeln, die sich dynamisch, sicher und automatisch den allgemeinen Erwartungen entsprechend an verschiedene Anwendungsszenarien anpassen können. MUSIC stellt eine eigens für diesen Fall entwickelte Softwremethodik sowie eine dezentralisierte Systemarchitektur für den Aufbau und die Implementierung adaptiver Anwendungen mobiler Nutzer zur Verfügung.

Im folgenden Abschnitt geben wir Beispiele für adaptive, mobile Applikationen in einer ubiquitären Umgebung und zeigen auf, welche Anforderungen eine solch flexible Nutzung an die Software und die Middleware stellt. Im Kapitel 3 erklären wir die Technologien, mit denen MUSIC diesen Anforderungen begegnet und stellen in den nächsten beiden Abschnitten die zwei Hauptergebnisse von MUSIC vor: die MUSIC Middleware und das MUSIC Studio. Kapitel 6 stellt zwei Pilotanwendungen vor, mit denen die Leistungsfähigkeit von MUSIC in großen, realen Szenarien demonstriert wird. Die Arbeit schließt mit der Vorstellung des MUSIC Projekts.

2 Anforderungen an adaptive Software in ubiquitären Umgebungen

Um die Anforderungen an eine komponentenbasierte, adaptive Plattform in einem ubiquitären- und dienstorientierten Umfeld zu verdeutlichen, verwenden wir das folgende, von einer MUSIC Pilotanwendung motivierte Szenario (siehe Abschnitt 6.1): Paul ist auf dem Weg von einem Konzert zu einer Freundin und wird dabei durch mobile Applikationen auf seinem Mobiltelefon geleitet. Er gerät in verschiedene Situationen, auf die er und sein Handy reagieren. Nach diesem Szenario erklären wir die besonderen Anforderungen, die ein solch flexibles Verhalten ermöglichen.

2.1 Beispielszenario: Paul auf seinem Weg zu einer Freundin

Paul kommt von einem Konzert in Paris und nimmt nun die U-Bahn zu einer Freundin, um ihre neue Wohnung zu sehen und ihr von der Show zu erzählen. Sein Mobiltelefon mit MUSIC-Technologie versteht WiFi, UPnP und GPS. Es stellt verschiedene Applikationen zur Verfügung, unter anderem einen Routenplaner und eine Tauschplattform für Medien.

Der Routenplaner, genannt Travel Assistant, unterstützt Paul beim Planen seiner Reise und beim Ticketkauf, benachrichtigt ihn über Verspätungen und hilft ihm, darauf zu reagieren. Die Tauschplattform, genannt InstantSocial, erscheint dem Benutzer als Webseite, die allerdings nicht auf einem zentralen Server im Internet basiert, sondern aus einer Verbindung von Diensten besteht, die von in der Nähe befindlichen Geräten angeboten werden: Je mehr Benutzer an ihr teilnehmen, desto robuster und attraktiver wird sie; sobald eine kritische Masse von Benutzern die Plattform verlässt, wird sie beendet.

Szene 1. Am Anfang des Szenarios betritt Paul die U-Bahn. Er plant die Reise zu seiner Bekannten, wofür er einen elektronischen Fahrplan als auch einen Lokalisierungs- und einen Kartendienst für den letzten Teil der Reise braucht. RATP, die Betreibergesellschaft, bietet einen ÖPNV-Fahrplan und einen Kartendienst in zwei Qualitätsstufen (QoS, quality of service) an: einfach und premium. Via UMTS kann Paul auch kommerzielle Premium-Dienste erreichen, allerdings zu höheren Kosten. Paul legt Wert auf hohe Qualität und sein Handy wählt die günstigen Premium-Dienste von RATP.

Szene 2. Paul entscheidet sich für eine Verbindung und kauft ein Ticket, beides mit Hilfe seines Routenplaners. Als regelmäßiger Kunde besitzt er einen elektronischen Pass. Beim

Passieren der Zugangskontrolle wird diese durch sein Telefon entdeckt, Pauls Pass wird überprüft und die Schranke öffnet sich automatisch.

Szene 3. Im Zug möchte Paul noch nach weiteren Bildern des Konzertes suchen. Er startet die InstantSocial Applikation, die sich gemäß den anderen InstantSocial Instanzen in der Umgebung konfiguriert. Sein Handy informiert ihn über eine passende Gruppe, der er beitrifft, und einen Moment später zeigt das Display eine Auswahl von Bildern, von denen jedes für eine Sammlung von interessanten Schnappschüssen steht. Er blättert durch die Auswahl, wählt einige aus und beginnt sie herunterzuladen.

Szene 4. Derweilen blockiert ein Störfall seinen Reiseweg. Sein Travel Assistant benachrichtigt ihn und schlägt eine neue Route mit einer anderen Zielstation vor. Unglücklicherweise ist die Planung der verbleibenden Strecke nicht so einfach: RATP reserviert einen Großteil der Bandbreite für das Einsatzpersonal und schaltet seine hochwertigen Dienste ab. Auch kann Paul kein GPS benutzen, da in der U-Bahn die Satelliten nicht sichtbar sind. Als beste Lösung wählt Pauls Telefon die kommerziellen Premium-Dienste, trotz ihrer hohen Kosten.

Szene 5. Nun ist Paul in einer Bahn mit wenigen Konzertbesuchern. Wegen der verminderten Robustheit adaptiert sich InstantSocial vom Tauschen zum reinen Sammeln von Bildern. Diesen Fokuswechsel führen auch die anderen InstantSocial-Instanzen durch, so dass die gemeinsame Tauschplattform schwächer wird. Letztendlich wird Paul über die schlechte Qualität informiert und er beendet InstantSocial.

Szene 6. Nachdem Paul die U-Bahn verlassen hat, beginnt das GPS-Modul zu arbeiten und sein Handy leitet ihn durch die Straßen. Minuten später erreicht er die Wohnung seiner Bekannten mit einem Telefon voller Impressionen.

2.2 Anforderungen an eine Middleware

Während Pauls Reise verwenden die Applikationen auf seinem Telefon flexibel verschiedene Dienste und Protokolle und funktionieren trotz zahlreicher Kontextwechsel. Insbesondere der Routenplaner und InstantSocial hängen von externen Diensten ab, die dynamisch gewählt und verwendet werden. Auch bietet jede Instanz von InstantSocial selbst Dienste für andere Instanzen an. All diese Beispiele von Flexibilität erfordern die Unterstützung einer Middleware wie MUSIC.

Die erste Szene demonstriert einen Auswahlprozess für Dienste, der von deren QoS abhängt. Die Verwendung eines UPnP-basierten Dienstes in Szene 2 verdeutlicht den Bedarf für alternative Verbindungsprotokolle und Dienste. Szene 3 zeigt die Anforderung, eine Vielzahl unabhängiger und paralleler Applikationen zu unterstützen und auch den dabei benötigten Grad an Flexibilität: Eine InstantSocial Instanz ist eine Kombination aus lokalen und externen Diensten, die in der Lage ist, selbst verschiedene Dienste zu verschiedenen QoS-Graden anzubieten. Über die tatsächliche Zusammensetzung der Instanz zu einem bestimmten Zeitpunkt muss hierbei zur Laufzeit entschieden werden. In Szene 4 wird die bewusste Verringerung des QoS eines externen Dienstes bei dessen Anbieter demonstriert. Daraus folgt eine Adaption zu einem alternativen Anbieter, obwohl der ursprüngliche den

Dienst aufrecht erhält. In Szene 5 hat das Handy die ungeplante Beendigung eines Dienstes durch das plötzliche Verschwinden von InstantSocial Instanzen zu verkraften. Zusätzlich zeigt sie die bewusste Beendigung eines Dienst-Angebots durch den Benutzer.

Um ein solches Szenario zu unterstützen, muss die Middleware eine Vielzahl von Fähigkeiten aufweisen: Neben Adaptivität und Kontext-Sensitivität sind dies auch SOA-Fähigkeiten, wie dynamische Entdeckung von Diensten, dynamisches Anbinden an einen Dienstanbieter und Änderung dieser Bindung, Verhandlung der Vereinbarung zur Dienstqualität und deren Kontrolle, sowie das Anbieten von eigenen Diensten.

3 Technologien für adaptive, mobile Software

Das Beispielszenario verdeutlicht verschiedene Anforderungen an eine adaptive Middleware. Im Folgenden zeigen wir, wie MUSIC diese Anforderungen erfüllt.

3.1 Komponentenbasierte Softwareentwicklung und Adaption

Adaptive Systeme sind in der Lage, sich während der Laufzeit an veränderte Situationen anzupassen. Adaptionen zu implementieren ist heutzutage sehr komplex und kostspielig. Dies wurde bisher nur in bestimmten Einsatzgebieten angewandt, in denen Systemstabilität gewährleistet sein musste, beispielsweise bei Telefon-Vermittlungsanlagen oder Geräten in der Raumfahrt. Dennoch ist Adaptivität in mehr und mehr Softwaresystemen mit mobilen Funktionen eine Voraussetzung geworden.

MUSIC vereinfacht für den Entwickler die Erstellung von adaptiven Anwendungen. Die Grundidee ist die komponentenbasierte Softwareentwicklung [AHE⁺06], bei der Applikationen hierarchisch aus Komponenten zusammengesetzt sind. Verschiedene Komponenten desselben Typs (z.B. Lokalisierung) können dieselbe Funktionalität implementieren (per GPS-Modul oder per WiFi) aber verschiedene nicht-funktionale Eigenschaften haben (Verfügbarkeit, Kosten). Jede konsistente Kombination der Komponenten ist dann eine mögliche Variante der Applikation. Der Entwickler gibt noch zusätzlich eine Funktion an, die jeder Variante in jeder Situation eine Eignung (Utility) zuweist, z.B. invers zu ihren Kosten.

Ändert sich die Umgebung zur Laufzeit und ist damit die aktuelle Variante möglicherweise nicht mehr optimal, so sucht die MUSIC Middleware durch alle Varianten und wählt diejenige aus, die momentan optimal für den Benutzer ist.

3.2 Dienstorientierte Architekturen (SOA)

Applikationen mit MUSIC-Technologie können im Sinne einer dienstorientierten Architektur (service-oriented architecture, SOA) Dienste verwenden und anbieten. Die Unter-

stützung von Diensten durch MUSIC ist durch die Möglichkeit motiviert, die Benutzbarkeit, die Nützlichkeit und die Robustheit von Applikationen zu kontrollieren, indem man sie an die sich ändernde Dienstelandschaft anpasst.

Hier muss man anmerken, dass es keine universell akzeptierte, standardisierte Definition von dienstorientierten Architekturen gibt; es ist nach [BGO04] wohl eines der am meisten überbeanspruchten Wörter der Softwareindustrie. Typischerweise werden Dienste als Funktionalitäten und Fähigkeiten definiert, die von einem Softwaresystem einem anderen Softwaresystem oder einem Benutzer angeboten werden. Dienste im Rahmen von SOA werden von unabhängigen Anbietern erstellt, im Netz installiert und durch standardisierte Mechanismen angeboten, so dass sie dynamisch entdeckt und verwendet werden können.

Ein fundamentales Konzept der Dienstorientierung ist der standardisierte Dienstkontrakt. Die Qualität eines Dienstes wird zwischen dem Anbieter und Nutzer verhandelt und als Teil des Kontrakts durch eine Qualitätsvereinbarung (Service Level Agreement, SLA) beschrieben. Mit einem SLA werden erwartete Leistungen (z.B. Reaktionszeiten und Verfügbarkeit) als auch Eigenschaften wie Abrechnung, Kündigung und Folgen einer SLA-Verletzung beschrieben.

MUSIC bindet Dienste in den Adaptationsmechanismus durch Dienst-Proxies ein: Entdeckt die MUSIC Middleware einen Dienst, so handelt sie mit dem Anbieter ein SLA aus und erstellt für den Dienst eine Proxy-Komponente. Die Eigenschaften des Proxies ergeben sich aus den Bedingungen des ausgehandelten SLA. Für die Applikation erscheint nun dieser Dienst wie eine normale Komponente, die die Funktionalität des Dienstes implementiert. Zum Beispiel haben die GPS-Komponente von Pauls Handy und die Proxy-Komponente des externen Lokalisationsdienstes denselben Typ; welche der beiden sich in einem konkreten Fall am besten eignet kann also in jeder Situation erneut entschieden werden.

3.3 Kontext-Sensitivität

Mehr und mehr Anwendungen sind heutzutage kontext-sensitiv. Das Öffnen der Schranke in Szene 2 ist hierfür ein Beispiel: Die Schranke ist ein UPnP-fähiges Gerät. Sobald Pauls Telefon erkennt, dass UPnP-Geräte verfügbar sind, adaptiert die Middleware (sie verwendet selbst MUSIC Technologien) und richtet eine UPnP-Schnittstelle ein. Der Routenplaner erkennt dann den Schrankendienst des Portals und verwendet ihn, um Paul den Einlass zu ermöglichen.

Kontext-Sensitivität ist eine der wichtigsten Konzepte, um Nutzer vor unüberschaubaren Bedienungsvorgängen zu bewahren. Sie erlaubt es, die erforderlichen Interaktionen sowie das Augenmerk des Benutzers auf diesen Bereich zu minimieren. MUSIC ermöglicht dies, indem es kontinuierlich und automatisch Informationen über den Status des Endnutzers und dessen zugehöriges Verhalten sammelt. Ziel ist es, durch das Filtern wichtiger Informationen über den Nutzerkontext und der Anwendungsumgebung eine intelligente Adaption der Anwendung ohne direkte Intervention des Endnutzers abzuleiten.

Sind die entsprechenden Informationen in ausreichender Form gesammelt, dann kann das

System regelmäßig eigenständige Entscheidungen treffen, die den Nutzer unterstützen, ohne dass dieser explizit eingreifen muss. MUSIC speichert eine große Anzahl an Informationen, die im Folgenden in drei Kategorien klassifiziert sind:

Ressourcenkontext – beschreibt die Informationen über Geräte im System und deren Abhängigkeiten untereinander. Er dient der Kontrolle der aktuellen Zustände (z.B. Gewohnheiten und Verbrauch), Ressourcen und Konfigurationsparameter. Daten des Ressourcenkontexts können sich auf verfügbare Netzwerke und Services, die Bildschirmgröße, Ausrichtung des Gerätes, verfügbare Speicher- und Batteriekapazitäten, etc. beziehen.

Umgebungskontext – beschreibt die Umgebung, in der sich Nutzer und Gerät befinden und bewegen. Kontext dieser Art wird u.A. von Sensoren gemessen. So kann zum Beispiel in Szene 4 durch Informationen über den Zug und die Linie, die Paul nimmt, Lokalität, Zeitpunkt und Ursache über ihn betreffende Störungen ermittelt werden.

Nutzer-“Kontext” – wie wichtige persönliche Daten (z.B. Name, Adresse), Informationen über Unternehmen, Arbeitsplatz, Hobbies oder Bedürfnisse etc., aber auch Vorlieben für Eigenschaften von Geräten. MUSIC unterstützt verschiedene Ebenen des Nutzerkontexts, die sich in ihrer Sichtbarkeit und Privatheit unterscheiden.

4 Die MUSIC Middleware

Die Entwicklung der MUSIC Middleware ist der Kern des Projektes. Die Middleware dient dazu, den Umgebungskontext einer Anwendung zu bewerten und die bestmögliche Konfiguration für den Nutzer automatisch auszuwählen. Dieser Prozess ist für den Anwender transparent. Die Middleware besteht aus der Kontext- und der Adaption-Middleware.

Die **Kontext-Middleware** ist dafür verantwortlich, Kontext zu überwachen, Kontext-Informationen zu verwalten und einen Kontextwechsel, der eine adaptive Veränderung verursacht, zu erkennen. Damit alle beteiligten Kontext-Quellen, -Vermittler und -Nutzer reibungslos interagieren können, verwendet MUSIC eine übergreifende Ontologie zur Kontext-Beschreibung. Zur Einbindung neuer Anwendungen kann die standardmäßig verfügbare MUSIC-Ontologie vom Entwickler um weitere Ontologien mit spezifischen Kontextdomänen erweitert werden. Die Kontext-Middleware nutzt einen Mechanismus, der durch logisches Ableiten – dem Kontext reasoning – den Anwendungen eine ausgereifte Darstellung des Kontextes bereitstellt. Darüber hinaus werden Kontextinformationen vor nicht autorisiertem Datenzugriff geschützt. Die Sicherheitsmechanismen für die Gewährleistung der Privatsphäre sind Teil der Middleware, die in alle MUSIC Komponenten eingebunden werden, die Kontextdaten erzeugen und übermitteln.

Die **Adaptions-Middleware** ist in der Lage, eine Anwendung dynamisch an den Kontext, in dem sie ausgeführt wird, anzupassen. Um für eine Anwendung die optimale Konfiguration zu ermitteln, ist es notwendig, die Abhängigkeiten zwischen den einzelnen Anwendungs-Parametern, den verfügbaren Ressourcen und den Rechenleistungen der Geräte zu kennen. Die konkreten Werte für diese Parameter werden bei jedem Kontextwechsel während der Laufzeit neu bestimmt.

5 Das MUSIC Studio

Um den Entwicklern die Arbeit zu erleichtern, gibt es eine spezielle Entwicklungsumgebung: das MUSIC Studio. Neben der Möglichkeit, Anwendungen zu erstellen, die die MUSIC Middleware nutzen, unterstützt diese Plattform auch die Evaluation und Wiederverwendung adaptiver Softwarelösungen durch mehrfach nutzbare Komponenten und Strukturen.

Das MUSIC Studio, das auf die Eclipse Plattform aufsetzt, ist als ein Open Source Projekt öffentlich zugänglich. Ziel der Werkzeuge ist es, die Entwicklung der adaptiven Anwendungen für universelle Anwendungsbereiche effektiver und praktischer zu gestalten. Dazu bietet das MUSIC Studio folgende Hilfsmittel:

Die **Modellierungssprachen** erlauben auf Basis von UML2 die Spezifikation von Adaptionsvarianten und deren Abhängigkeit von Kontextzuständen.

Die **Modellierungswerkzeuge** unterstützen die Gestaltung adaptiver Modelle durch die Modellierungssprachen. Ein visuelles Gestaltungswerkzeug ermöglicht dem Entwickler eine interaktive Modellierung der adaptionsfähigen Anwendung.

Die **Transformationswerkzeuge** dienen dazu, plattform-unabhängige adaptive Modelle in plattform-abhängige Modelle zu transformieren. Damit eine modellierte adaptionsfähige Anwendung ablauffähig wird, ist es notwendig, das Modell auf die vorgesehene spezifische Zielplattform abzubilden. Im Rahmen des Projektes ist zunächst die Realisierung einer Transformation für die Java Plattform vorgesehen.

Die **Simulationswerkzeuge** dienen zum Test der Anwendung in einer simulierten Labor-Umgebung mit fest vorgegebenen Kontextwerten. Ein modellgesteuerter Ansatz unterstützt die Validierung und Erprobung der Entwurfsentscheidungen durch die Simulation des Adaptionsverhaltens in frühen Stadien des Entwicklungsprozesses. Damit hilft das Werkzeug Fehler im Aufbau und der Funktionalität zu identifizieren. Basierend auf den gewonnenen Erkenntnissen können durch die MUSIC Konfigurationswerkzeuge widersprüchliche adaptive Strategien, Konflikte im Gebrauch von Ressourcen, widersprüchliche Sicherheitspolitiken oder andere Schwachstellen aufgedeckt und repariert werden. Dabei werden auch allgemein zugängliche Werkzeuge für die Verwendung im MUSIC Projekt angepasst und erweitert (z.B. die Eclipse Test- und Performance Tool Plattform, www.eclipse.org/tptp).

6 Die Pilotanwendungen

Im MUSIC Projekt werden zwei Pilotanwendungen entwickelt und in einer realen Umgebung getestet. Sie demonstrieren die Vorteile, die private und berufliche Anwender im täglichen Gebrauch ihrer Applikationen mit MUSIC-Technologie haben.

6.1 Der Pariser Metro Pilot

Die Pariser Metro Pilotanwendung versorgt einen Reisenden auf seinem Weg durch die Metro stets mit aktuellen Informationen für die besten Verbindungen über seinen PDA. Weitere Interaktionsmöglichkeiten erlauben ihm den Austausch von Dokumenten oder Bildern mit Freunden während des Wartens an einem Bahnhof oder im Zug. Zudem wird dem Reisenden ermöglicht, bei unerwarteten Zwischenfällen, wie z.B. einem Unfall, Unterstützung leisten. Zur umfassenden Demonstration der adaptiven Fähigkeiten des PDAs erhält der Reisende die Anwendungen Travel Assistent, InstantSocial und Irregular Operations. Diese Pilotanwendung wird von der Betreibergesellschaft der Pariser Metro RATP (www.ratp.fr) erstellt und in einer U-Bahn Station validiert; sie wird im Beispielszenario in Kapitel 2 eingesetzt.

Der Travel Assistent unterstützt den Reisenden von einem Ausgangs- zu einem Zielpunkt durch einen Routenplaner, einer Darstellung von Routen und einer Touristen-Information. Der Routenplaner beschreibt den genauen Wegverlauf mit den verschiedenen Verkehrsmitteln und Umsteigepunkten, so dass ihm umständliches Suchen und Blättern in gedruckten Metro- und Stadtplänen erspart bleibt. Sobald der Weg ausgewählt ist, wird der Reisende mit der Routen-Anzeige durch die Stadt geführt, wobei die Route, dem kleinen Display entsprechend, in mehrere Schritte aufgeteilt ist.

Der PDA versucht, dem Reisenden immer den höchsten Nutzen zu bieten, wobei er auf Grundlage von Nutzerprofil, Tagesplan, Displayeigenschaften, Licht, Lärmpegel, Speicher, verfügbaren Netzwerken und Geräten sowie seiner über GPS ermittelten Position entscheidet. So zeigt der PDA z.B. nur Straßen und Objekte an, die für ihn relevant sind. Innerhalb einer Station wird eine andere Darstellung als auf der Straße gewählt. Erkennt der PDA eine Informationssäule der Pariser Metro via RFID, passt er sich so an, dass das Bild auf dem größeren Display verfeinert angezeigt und ausgedruckt wird. Ist der Reisende als Besucher in der Stadt, wird er von der Touristen-Informationsfunktion auf weitere Sehenswürdigkeiten entlang des Weges hingewiesen. Der PDA entscheidet auf Basis von verfügbarer Bandbreite, Speicher und Kosten, ob lediglich Text oder auch Bilder und Videos angezeigt werden. Erkennt der PDA einen Ticket-Automaten, bietet er den Kauf einer elektronischen Fahrkarte an, so dass das Gerät selbst zum Ticket wird. Die Verbindung vom PDA zu anderen Nutzern oder Geräten erfolgt durch WLAN, Bluetooth oder GSM.

Die InstantSocial Anwendung [FHS08] ist motiviert durch die wachsende Verfügbarkeit von Internetzugängen, digitalen Kameras und dem sozialen Bedürfnis, Kontakte zu Freunden oder Kollegen aufrecht zu halten, auch wenn sie weit verstreut leben. MUSIC unterstützt diese Peer-2-Peer Anwendung zum Austausch von Daten, Bildern und Nachrichten zwischen mehreren Nutzern. Z.B. hilft sie bei der Identifikation von anderen Nutzern mit gleichen Interessen und der Herzustellen eines Kontaktes. Die für eine Nutzergruppe interessanten, verteilten Daten wie z.B. Bilder oder Videos werden effizient nach dem Peer-2-Peer Konzept auf den Nutzer-PDAs verteilt abgelegt. Zudem können externe Geräte identifiziert und zur Ablage von Daten oder zum Ausdrucken verwendet werden.

Die Irregular Operation Anwendung wird bei unvorhergesehenen Ereignissen aktiviert, wie bei einem Unfall, Feuer oder technischem Defekt eines Zuges: Hier bietet der PDA

Funktionen an, um einen neuen Weg zu finden, Hilfe zu holen oder erste Hilfe zu leisten. Andere Reisende werden umgeleitet, während Sicherheitskräfte zum Unfallpunkt dirigiert werden. Da in solchen Fällen jeder erst einmal per Handy seine Verspätung mitteilt, tritt meist auch eine Netzwerküberlastung ein. Mit Hilfe von MUSIC erfolgt eine Priorisierung der Funktionen nach der Relevanz für die kritische Situation: z.B. haben Funktionen für die Ambulanz oder das Sicherheitspersonal Vorrang vor Touristen-Hinweisen, Spielen oder News. Zur Unterstützung der Ersten Hilfe werden vom PDA entsprechend dem vorliegenden Notfall geeignete Informationsquellen oder Experten ausgesucht; die Auswahl erfolgt auf Grundlage von Meta-Daten und Profilen zu den verfügbaren Diensten und Personen.

6.2 Der SatCom Installation Pilot

Der SatCom Installation Pilot der Firma IntegraSys S.A. (www.integrasys-sa.com) demonstriert die Unterstützung der Installation und Ausrichtung von VSAT Antennen. Dies erfolgt über ein Mess- und Einrichtungsinstrument, das Empfangsleistung und Signalparameter am Installationsort misst und drahtlos über einen PDA von einem Monteur kontrolliert wird. Der PDA adaptiert entsprechend dem Licht, dem Lärmpegel und der Qualität der drahtlosen Verbindung. Z.B. schaltet er bei schlechter Verbindung in einen reinen Empfangsmodus oder er reduziert die Übertragung von Bild- und Messdaten, je nach dem welche Funktionen in welchem Arbeitsschritt benötigt werden. Weiterhin sind Module auf entfernte Laptops auslagerbar, um Speicher zu sparen oder die Performance zu verbessern. Generelles Ziel ist es, den Monteur bei allen seinen Aufgaben durch den PDA zu unterstützen, wie z.B. bei der Routenplanung, der Aktualisierung von Aufträgen aus der Zentrale sowie der Kommunikation mit anderen Monteuren.

7 Das MUSIC Projekt

MUSIC (www.ist-music.eu) ist ein EU Projekt des sechsten Rahmenprogramms und läuft dreieinhalb Jahre von Oktober 2006 bis März 2010. Es ist forschungsorientiert und verfolgt eine Open Source Philosophie. Das Interesse der Wirtschaft an den Ergebnissen soll durch große, praxisnahe Pilotanwendungen demonstriert werden. Dieser Fokus sowohl auf die Forschung als auch auf die Realisierung spiegelt sich in der Auswahl der Projektpartner wider.

An MUSIC arbeiten vierzehn Partner aus neun europäischen Ländern. An der Entwicklung der Pilotanwendungen beteiligen sich u.a. die Condat AG (Deutschland), Hewlett-Packard (Italien) und RATP (Frankreich). Auf der Forschungsseite engagieren sich die Universitäten von Leuven (Belgien), Oslo (Norwegen) und Kassel. Als Beispiel für kleine und mittlere Unternehmen im Projekt – und damit als Mittler zwischen Forschung und Entwicklung – findet man die European Media Laboratory GmbH (EML). Die internationale und breite Ausrichtung des Projekts bildet eine gute Basis für innovative Lösungen.

Das Projekt MUSIC entstand auf der Basis des MADAM Projektes (Mobility and Adaptation enAbling Middleware), das im Bereich Information Society Technologie (IST) angesiedelt war. Ziel dieses Projektes war es, Softwareentwickler mit erweiterten Modellierungssprachen, Werkzeugen und Middleware-Komponenten zu unterstützen. Diese sollten gemeinsam die Gestaltung, Implementierung und Handhabung innovativer Anwendungen und Dienste für den mobilen Anwender verbessern. Das MADAM Projekt wurde im Frühjahr 2007 erfolgreich abgeschlossen.

Das MUSIC Projekt ist daran interessiert, die erstellten Technologien und Softwaresysteme für weitere mobile Anwendungen im produktivem Betrieb einzusetzen. MUSIC wendet sich an Unternehmen, die komplexe Applikationen für tragbare Geräte entwickeln und die MUSIC Middleware zum Einsatz in ubiquitären Umgebungen mit geringem Aufwand integrieren wollen. Die Bereitstellung von Entwicklungswerkzeugen und Verwendung einer Open Source Lizenz macht dies auch für kleinere und mittlere Unternehmen möglich und bezahlbar. Über die MUSIC Webseite und eine geplante Distributionsplattform kann man zudem an der Weiterentwicklung von MUSIC teilnehmen und die Entwickler und Nutzer von MUSIC erreichen. Treten Sie dem Orchester bei!

8 Danksagung

Das MUSIC Projekt wird von der Europäischen Kommission unter der Nummer 035166 unterstützt. Kapitel 2 und 3.2 orientieren sich an einer Arbeit von Rouvoy *et. al.* [RBD⁺08]. Unser Dank gilt Peter Saueressig und allen Mitgliedern des MUSIC Projektes für ihre Unterstützung.

Literatur

- [AHE⁺06] Mourad Alia, Geir Horn, Frank Eliassen, Mohammad Ullah Khan, Rolf Fricke und Roland Reichle. A Component-based Planning Framework for Adaptive Systems. In *8th International Symposium on Distributed Objects and Applications (DOA)*, Oktober 2006.
- [BGO04] Ziv Baida, Jaap Gordijn und Borys Omelayenko. A shared service terminology for online service provisioning. In *Proceedings of the 6th international conference on Electronic commerce*, Seiten 1–10, 2004.
- [FHS08] Luís Fraga, Svein Hallseinsen und Ulrich Scholz. “InstantSocial” - Implementing a Distributed Mobile Multi-user Application with Adaptation Middleware. *Communications of the EASST*, 11, 2008.
- [RBD⁺08] Romain Rouvoy, Paolo Barone, Yun Ding, Frank Eliassen, Svein Hallsteinsen, Jorge Lorenzo, Alessandro Mamelli, und Ulrich Scholz. MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments. Eingereicht bei LNCS Hot Topics on Software Engineering for Self-Adaptive Software Systems, 2008.