



SAMProc

Middleware für selbstadaptive mobile Prozesse in heterogenen ubiquitären Umgebungen

Holger Schmidt¹, Rüdiger Kapitza², Franz J. Hauck¹

¹ Institut für Verteilte Systeme
Universität Ulm

² Lehrstuhl für Informatik 4
Universität Erlangen-Nürnberg

1 Motivation

- Fokus: Anwendungen für dynamische und heterogene Umgebungen
 - ◆ Ubiquitous Computing

- Anwendungscharakteristiken:
 - ◆ Komplexität interner Aktivitäten
 - ◆ Interaktivität mit anderen Anwendungen
 - ◆ Mobilität
 - Mobile Benutzer, optimale Nutzung der Rechenleistung der Umgebung, Einsparung von Ressourcen, Heterogenität
 - ◆ Adaptivität / Adaptierbarkeit
 - Anwendungskontext, heterogene Infrastruktur, Geräte mit beschränkten Ressourcen, komplexe/ fehleranfällige Implementierung
 - ◆ Durchgängige Addressierbarkeit

1 Motivation

- Fokus: Anwendungen für dynamische und heterogene Umgebungen
 - ◆ Ubiquitous Computing

- Anwendungscharakteristiken:
 - ◆ Komplexität interner Aktivitäten
 - ◆ Interaktivität mit anderen Anwendungen
 - ◆ Mobilität
 - Mobile Benutzer, optimale Nutzung der Rechenleistung der Umgebung, Einsparung von Ressourcen, Heterogenität
 - ◆ Adaptivität / Adaptierbarkeit
 - Anwendungskontext, heterogene Infrastruktur, Geräte mit beschränkten Ressourcen, komplexe/ fehleranfällige Implementierung
 - ◆ Durchgängige Addressierbarkeit

- ★ **Middleware für mobile und adaptive Anwendungen mit Entwicklungsunterstützung**

2 Gliederung

- Teil 1: SAMProc und AWSM
 - ◆ Selbstadaptive mobile Prozesse
 - ◆ Szenario / Beispielanwendung
 - ◆ MDA-gestützte Anwendungsentwicklung
 - ◆ AWSM Middleware
 - ◆ Fazit und aktuelle Forschung

- Teil 2: Dynamisches Laden von Code
 - ◆ Zentrale oder dezentrale Infrastruktur?
 - ◆ Kriterien der Codeselektion
 - ◆ DCM-Plattform
 - ◆ OSGi Integration
 - ◆ Fazit und aktuelle Forschung

3 Selbstadaptive mobile Prozesse (SAMProc)

- Implementierung von Adaptierungsschritten ist komplex und fehleranfällig
 - ◆ Lösung: Selbstadaptive mobile Prozesse

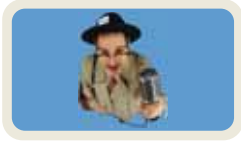
- Was ist ein selbstadaptiver mobiler Prozess? [*MAI'07, OT4AmI'07, MDS'07*]
 - Unsere Abstraktion für selbstadaptive und mobile Anwendungen
 - ◆ Eindeutige Identität (externe Sicht)
 - ◆ Geordnete Ausführung von Diensten (interne Sicht)
 - ◆ Selbstadaptierbarkeit (Kontext)
 - Zustand, Funktionalität, Implementierung
 - ◆ Mobilität
 - Lokale Ausführung von Diensten , Zugriff auf bestimmten Kontext

- Konzept
 - ◆ Anwendung wird als SAMProc beschrieben...
 - ◆ Abstrakte Spezifikation des Lebenszyklus und von Verteilungsaspekten

3 Szenario / Beispielanwendung



- Mobile Reporteranwendung
 - ◆ Mobiler Anwendungs-Workflow als SAMProc



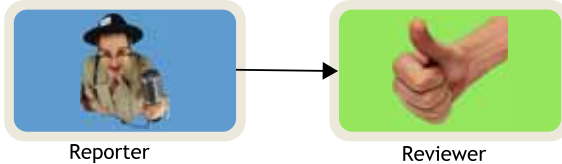
Reporter

3 Szenario / Beispielanwendung



- Mobile Reporteranwendung

 - ◆ Mobiler Anwendungs-Workflow als SAMProc



3 Szenario / Beispielanwendung

■ Mobile Reporteranwendung

◆ Mobiler Anwendungs-Workflow als SAMProc



3 Szenario / Beispielanwendung

■ Mobile Reporteranwendung

◆ Mobiler Anwendungs-Workflow als SAMProc



■ Dynamische Infrastruktur

◆ Reporter werden zu Reviewer nach bestimmter Anzahl erfolgreicher Berichte

■ Initiales Deployment

◆ Rettungsleitstellen starten Berichtsprozesse mit initialen Code (z.B., MMS)

■ Feedback

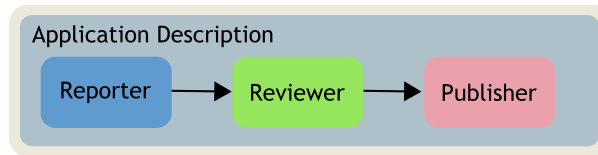
◆ Reviewer und Publisher können weitere Berichtsprozesse als SAMProc starten

◆ Reporter können Workflow "überwachen"

4 MDA-gestützte Anwendungsentwicklung

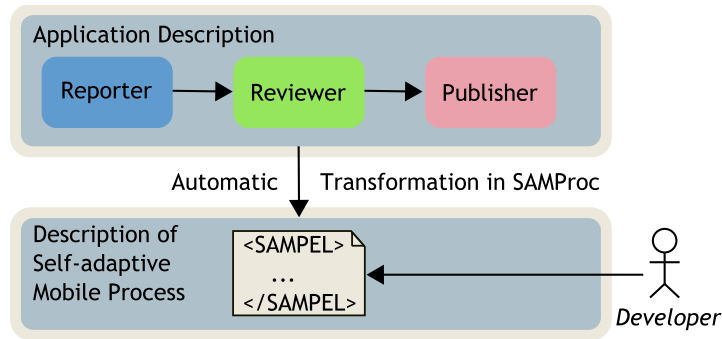


- Transformation eines high-level models (PIM) in Implementierung (PSM)



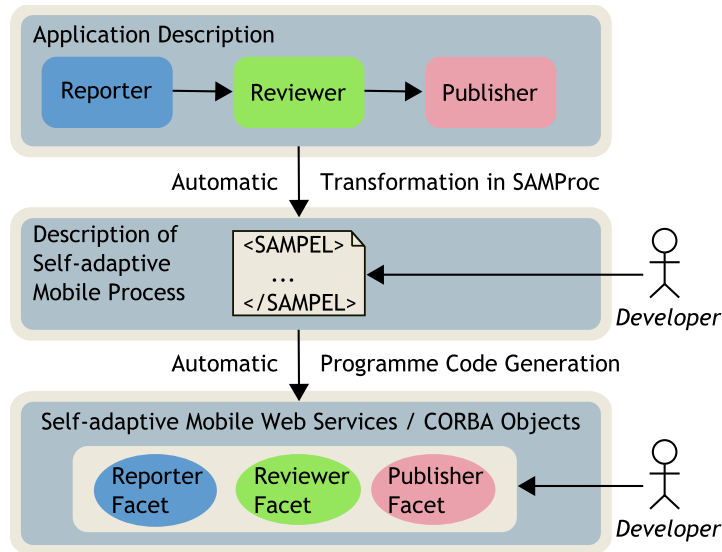
4 MDA-gestützte Anwendungsentwicklung

- Transformation eines high-level models (PIM) in Implementierung (PSM)



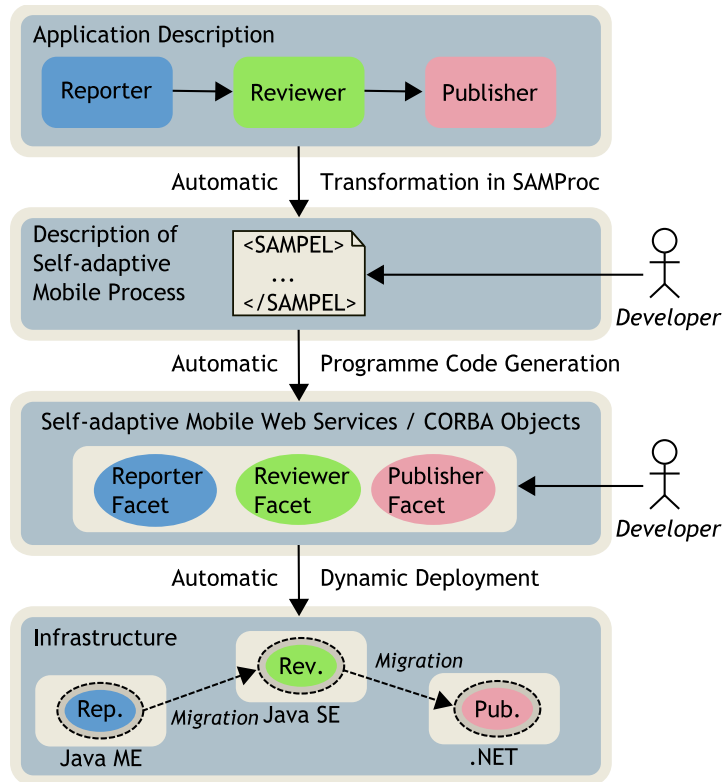
4 MDA-gestützte Anwendungsentwicklung

- Transformation eines high-level models (PIM) in Implementierung (PSM)



4 MDA-gestützte Anwendungsentwicklung

■ Transformation eines high-level models (PIM) in Implementierung (PSM)



5 AWSM Middleware

- Selbstadaptive mobile Web Services und CORBA Objekte
[*DOA'06, OT4AmI'06, DOA'07, DAIS'07*]

- Facetten-Konzept
 - Dynamische Rekonfiguration zur Laufzeit
 - ◆ Schnittstelle (Funktionalität)
 - ◆ Zustand
 - ◆ Implementierung

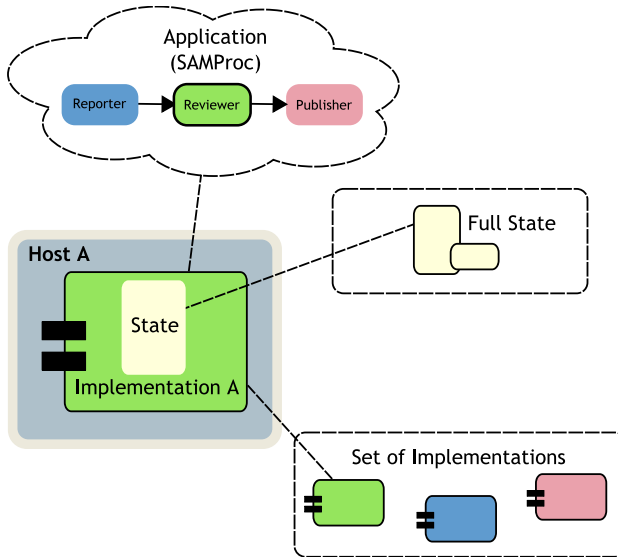
- Konzept des aktiven and passiven Zustands
 - ◆ Gesamtzustand = aktiver + passiver Zustand

- Migration: Übertragung implementierungsunabhängiger Zustand

- Generisches Konzept (unabhängig von genutzer Technologie)

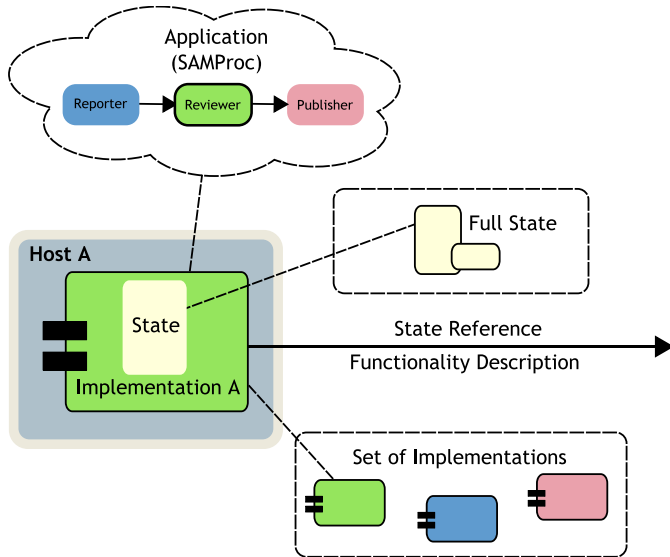
6 AWSM Middleware

■ Exemplarische Implementierung eines Anwendungsprozesses



6 AWSM Middleware

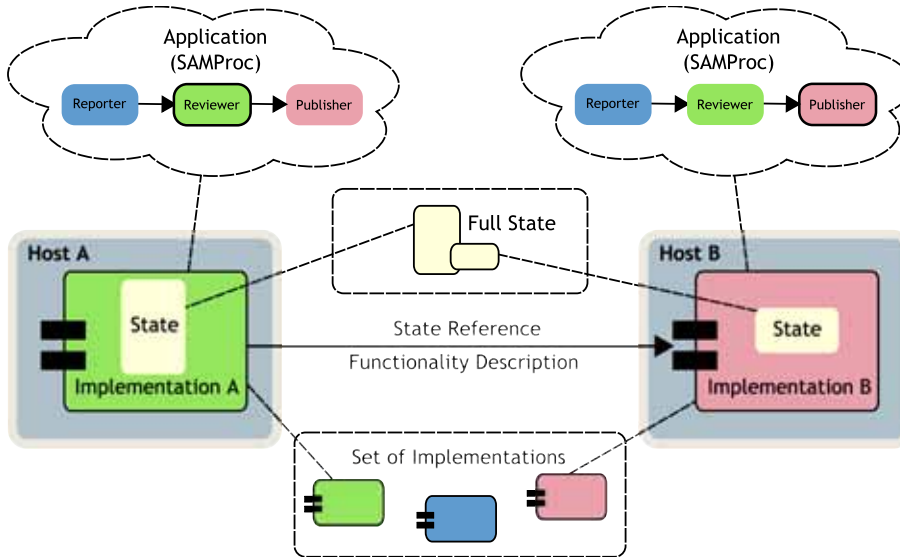
■ Exemplarische Implementierung eines Anwendungsprozesses



6 AWSM Middleware



■ Exemplarische Implementierung eines Anwendungsprozesses



7 Fazit und aktuelle Forschung

- SAMProc bietet Lösungen für wichtige Fragestellungen im UbiComp
 - ◆ Unterstützung für Migration in heterogenen Umgebungen
 - Abstrakte Zustandsbeschreibung
 - ◆ Reduzierte Komplexität für die Implementierung von Anpassungsschritten
 - Trennung Adaptions-/Migrations- von Anwendungslogik (MDA-Ansatz)
 - ◆ Angepasste Anwendungen für ressourcenbeschränkten Geräten
 - Minimale Daten zur Ausführung, Laden maßgeschneiderter Code bei Bedarf

7 Fazit und aktuelle Forschung

- SAMProc bietet Lösungen für wichtige Fragestellungen im UbiComp
 - ◆ Unterstützung für Migration in heterogenen Umgebungen
 - Abstrakte Zustandsbeschreibung
 - ◆ Reduzierte Komplexität für die Implementierung von Anpassungsschritten
 - Trennung Adaptions-/Migrations- von Anwendungslogik (MDA-Ansatz)
 - ◆ Angepasste Anwendungen für ressourcenbeschränkten Geräten
 - Minimale Daten zur Ausführung, Laden maßgeschneiderter Code bei Bedarf

- Aktuelle Forschung
 - ◆ Optimierung der AWSM Infrastruktur
 - ◆ Beschreibungssprache und Codeerzeugung -> Modellierungswerkzeug
 - ◆ Implementierung eines Web Service Containers für Java ME
 - ◆ Weiterführende Mechanismen für Feedback, Klonen und Mergen
 - ◆ Weitere Anwendungsfelder und Szenarien

8 Dynamisches Laden von Code

■ Rahmenbedingungen selbstadaptiver mobiler Prozess

- ◆ Selbstadaptierbarkeit (Kontext)
 - Zustand, Funktionalität, **Implementierung**
- ◆ Mobilität in einem heterogenen Umfeld
 - Welches System zur Laufzeit
 - Systeme unterschiedlicher Leistung

→ **Dynamisches Laden von spezifischem Code!**

■ Zentrale oder dezentrale Infrastruktur

- ◆ Keine einzelne zentrale Einheit gegeben
- ◆ Standardisierte Komponenten unterschiedlicher Entwickler
- ◆ Skalierbarkeit und Fehlertoleranz

→ **Dezentrale Infrastruktur!**

9 Kriterien der Code-Selektion

- Funktionale Anforderungen
 - Schnittstellen

- Anforderungen der Plattform
 - Betriebssystem, CPU, etc.

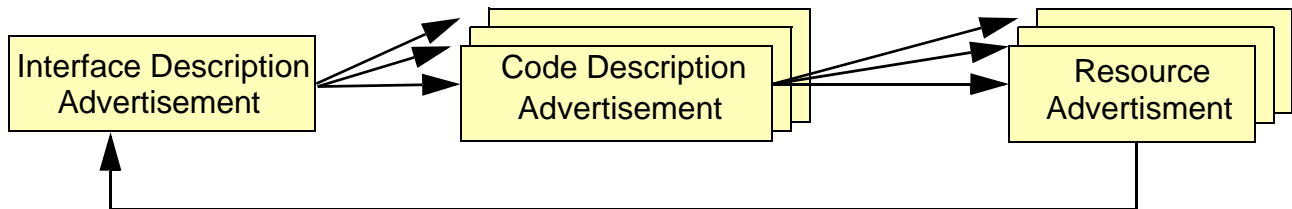
- Nicht-funktionale Anforderungen
 - Ressourcenverbrauch, Performance, etc.

10 Dezentrale Infrastruktur zum Laden von Code



- FOXTA [DAIS'07] und DCM [MiNEMA'08]
 - ◆ JXTA: P2P Infrastruktur zum dezentralen Suchen/Publizieren von Ressourcen
 - ◆ OSGi: Komponentenmiddleware als Basis und zur Evaluierung

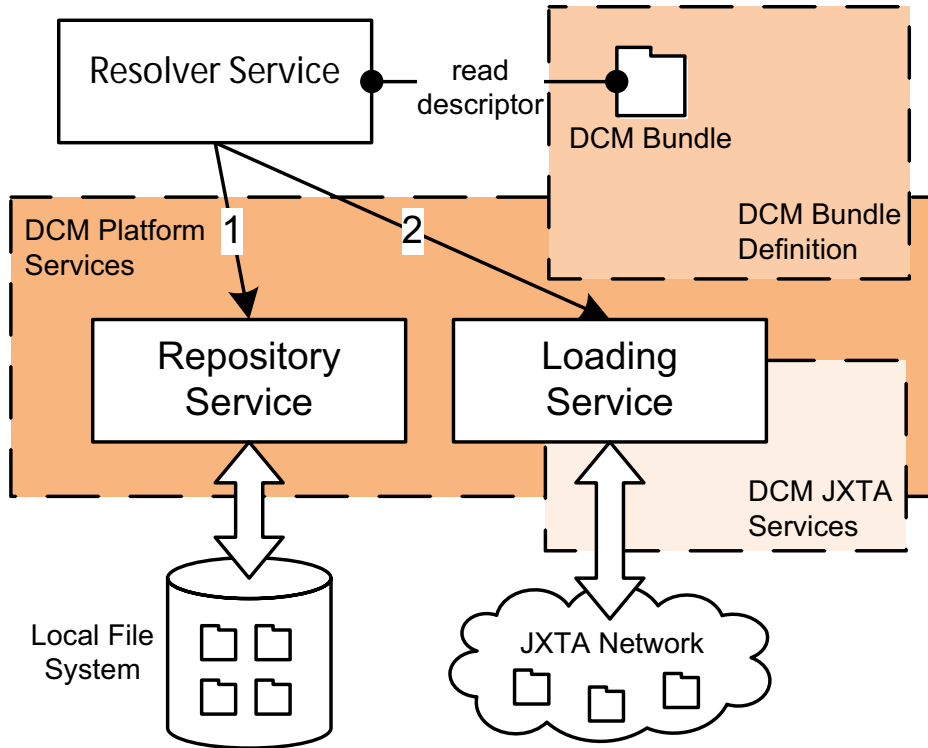
- Code/Ressource wird durch folgende Advertisements repräsentiert
 - ◆ Interface Description Advertisement (IDA)
 - ◆ Code Description Advertisement (CDA)
 - ◆ Resource Advertisement (RA)



11 DCM Plattform



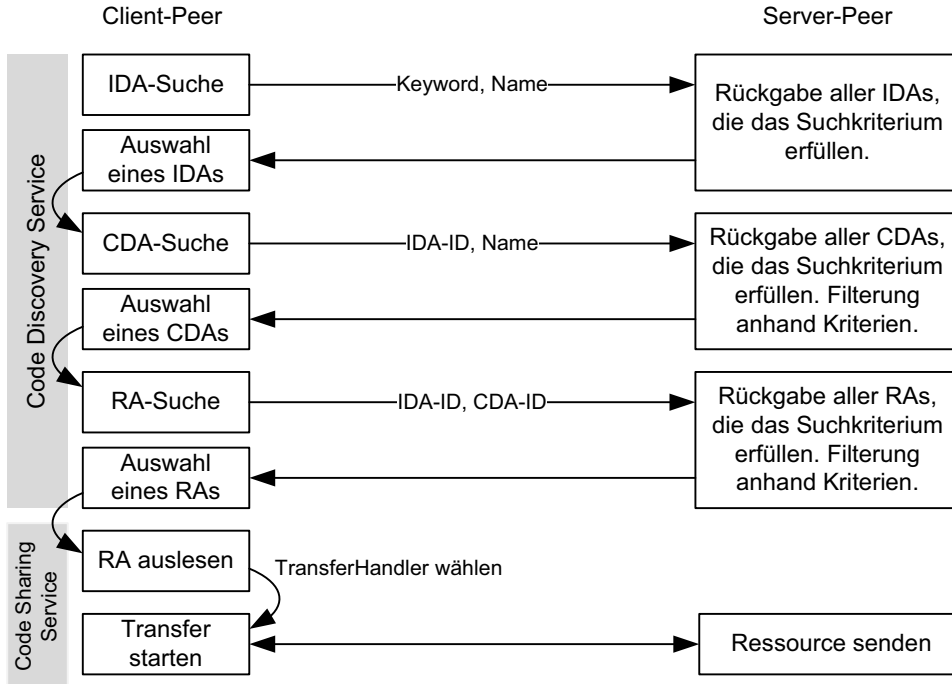
■ Architektur



12 DCM Plattform



■ Laden von Code: Ablauf



13 OSGi Integration



- Bisher ermöglicht OSGi die Integration und Verwendung von Code in unabhängigen Arbeitsschritten
- Transparente Integration der Ladeinfrastruktur in den Bindevorgang
 - ◆ Binden/Laden kann durch die Applikation gesteuert werden
 - ◆ Nicht-funktionale Eigenschaften können durch die Anwendung/Plattform ausgewertet werden
- Beispiel: OSGi HTTP Service
 - ◆ Fast alle aktuellen OSGi Plattformen implementieren diesen Dienst
 - ◆ Je nach Anwendung und System ist Performance oder Ressourcenverbrauch wichtiger

14 Fazit und aktuelle Forschung

- Merkmale der entwickelten Plattformen
 - ◆ Dezentralisierung der Bereitstellung und Suche von Code
 - ◆ Berücksichtigung nicht-funktionaler Eigenschaften
 - ◆ Automatische Auflösung von Abhängigkeiten (z.B. bei OSGi)

- Aktuelle Arbeiten
 - ◆ Evaluierung von nicht-funktionalen Eigenschaften durch Beispielszenarien
 - ◆ Erweiterung und Ergänzung der OSGi Plattform durch DCM

Danke für Ihre Aufmerksamkeit!



Fragen?

<http://www-vs.informatik.uni-ulm.de/proj/samproc>