Selbst-adaptive Software

INF-M2 - Anwendungen 1 - Sommersemester 2008

Tobias Hutzler

Department Informatik
HAW Hamburg

20. Mai 2008

Agenda

Agenda

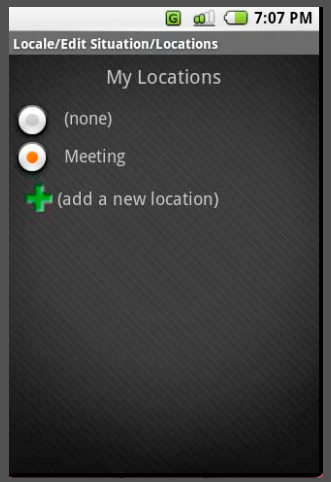
Locale - Android Developer Challenge - One of 50 Winners

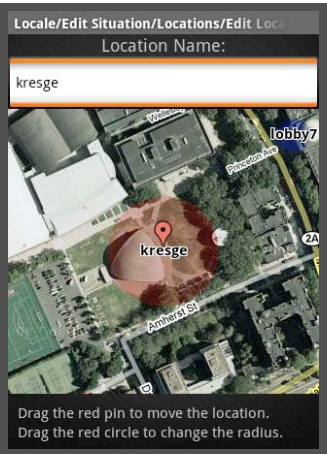
Locale

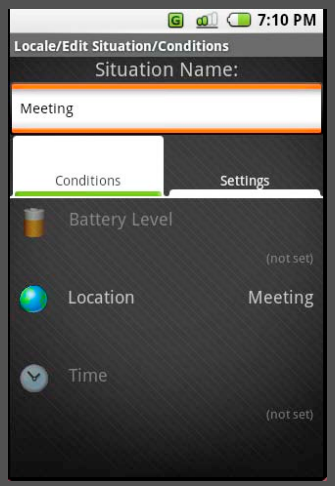
Developer: Clare Bayley, Carter Jernigan, Christina Wright, Jasper Lin

Dynamic settings manager that automatically changes your settings based on conditions, such as your current location.

- Locale ensures your ringer will never go off at an inopportune time again
- Third parties can leverage the Locale platform to reuse locations or add custom settings and conditions
- So easy to use, you'll wonder how you ever lived without it







Set It and Forget It!

Abbildung: Locale entwickelt von Studenten am MIT [http://code.google.com/android/adc.html]

Ideale Anwendungszenarien für Adaption - Notwendigkeit

- Mobile and pervasive computing, ...
  - Variabilität der Anwendungen
  - Contextabhängige Benutzer Vorlieben

Selbst-adaptive Software

Definition:

- Self-adaptive software modifies its own behavior in response to changes in its operating environment [P. Oreizy et al, 1999]
- Self-adaptive software evaluates its own behavior and changes behavior when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible. [DARPA BAA]

Selbst-adaptive Software Systeme

- Herausforderungen an Systemarchitekten und Softwarearchitekten
  - Wie kann man Komponenten organisieren, damit Überwachung und Anpassung möglich wird?
  - Wie, wann und wo soll man die Adaption ausführen?
- Bekannter Ansatz: **feed-back control systems**, welcher die automatische Adaption ermöglicht!
  - Bekannt unter dem Namen: autonomic computing or self-* systems
    - Schon einige Vortäge im Rahmen AW1 und AW2
    - self-*: capturing automatic adaptation in a variety of ways
    - self-managing, self-healing, self-configuring, self-optimizing, etc

Agenda

Wie kann man dynamische Adaption realisieren?

Dynamische Adaption

Dynamische Adaption findet zur Laufzeit der Anwendung aufgrund von Änderungen des Kontexts oder Ressourcenzustandes statt.[Geihs]

- Parametrische Adaption
- Kompositionelle Adaption

Parametrische Adaption I

- Einfache Form der dynamischen Adaption
- Längst bekannt! Beispiele:
  - Helligkeitssensor eines Laptops, der die Bildschirmhelligkeit anpasst
  - TCP - Anpassung der Größe des Übertragungsfensters an die Last- und Fehlersituation im Netzwerk

Parametrische Adaption II

- Der Zustand der Umgebung beinflusst interne Parameter der Anwendung
- Die Anwendungsstruktur wird nicht verändert
- Die Adaptionsmöglichkeit wird zur Entwurfszeit durch Parameter vom Entwickler festgelegt
- Die Adaption erreicht meist eine gute Performanz

Kompositionelle Adaption

Compositional adaptation

Compositional adaptation results in the exchange of algorithmic or structural parts of the system, in order to improve a program's fitness to its current environment [McKinley, P.K., et al.]

- Kontextabhängig zur Laufzeit können Komponenten der Anwendung ausgetauscht werden
- Neue Komponenten können hinzukommen, alte Komponenten können wegfallen
- Komponenten sind unter Umständen zur Entwurfszeit noch gar nicht bekannt
- Sehr mächtige Art der Adaption, aber auch sehr komplex

Adaptions Mechanismen

- Parameter tuning
- Code (agent or component) migration
- Compositional adaptation
  - aspect weaving
  - reflection
  - component-based

Kompositionelle Adaption - bekannte Techniken

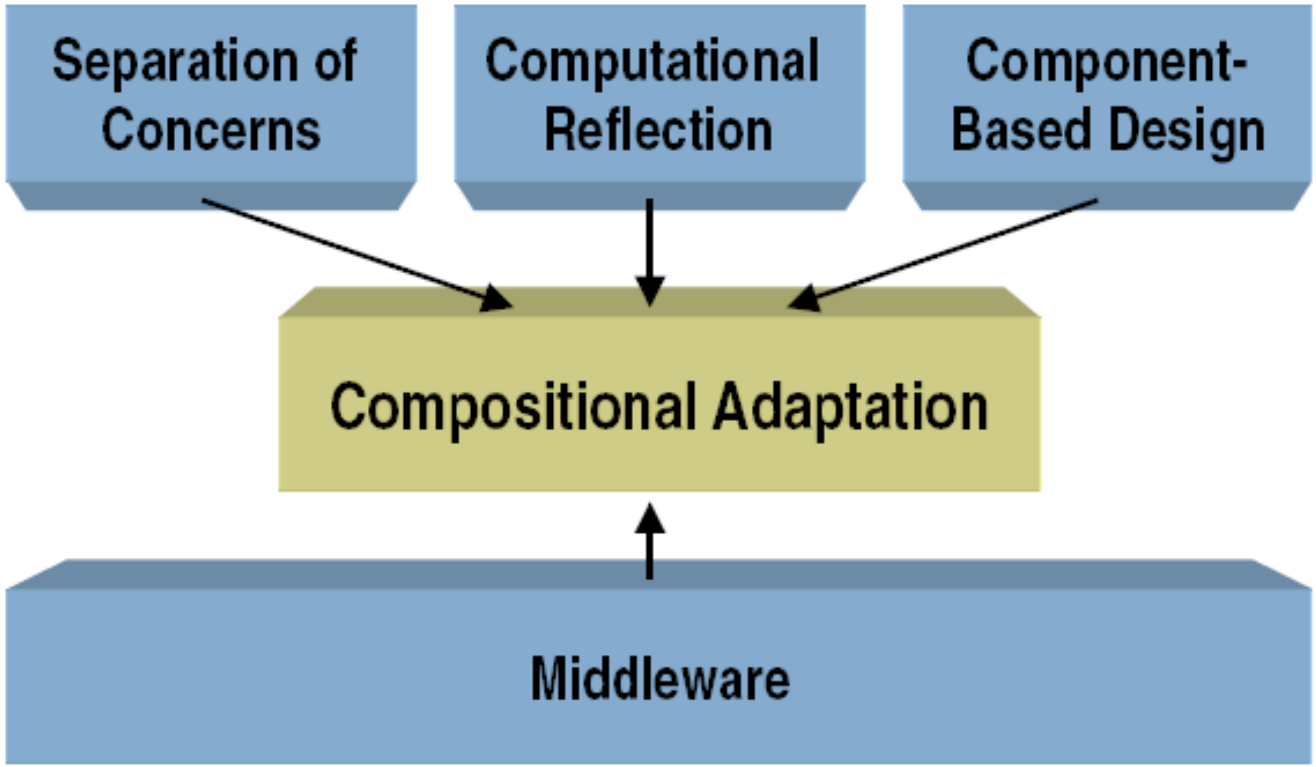


Abbildung: Main technologies supporting compositional adaptation [McKinley, P.K., et al.]

Vorhandene Technologien - Aspect Weaving

Aspektorientierte Programmierung(AOP)

- Verbreiteter Ansatz um mit **cross cutting concerns** umzugehen
- cross cutting concerns (Security, QoS, Fault tolerant, etc)
- **AOP** ermöglicht separation of cross cutting concerns als **Aspects**
- Aspects werden getrennt entwickelt und zur Kompilierungszeit eingewoben (aktuell: Einwebung zur Laufzeit)
- **Pointcuts** definieren Stellen, an denen Aspects eingewoben werden können

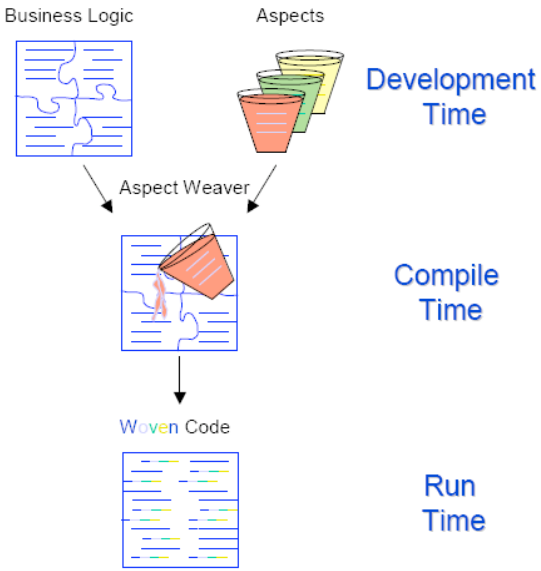


Abbildung: Aspect Weaving [McKinley, P.K., et al.]

Vorhandene Technologien - Reflection

Introspection

Ermöglicht einer Anwendung sich selbst zu beobachten.

Intercession

Ermöglicht einer Anwendung ihr selbst beobachtetes Verhalten zu ändern.

Base level

Anwendung selbst (code)

Meta level

Selbstrepräsentation (model) der Anwendung

Meta-object Protocol (MOP)

Systematische Introspection und Intercession

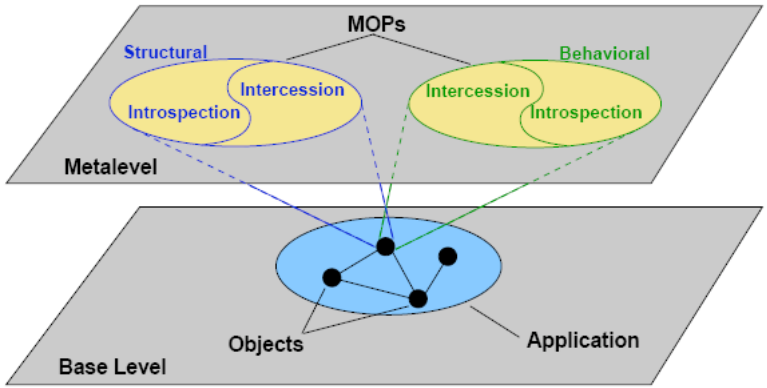


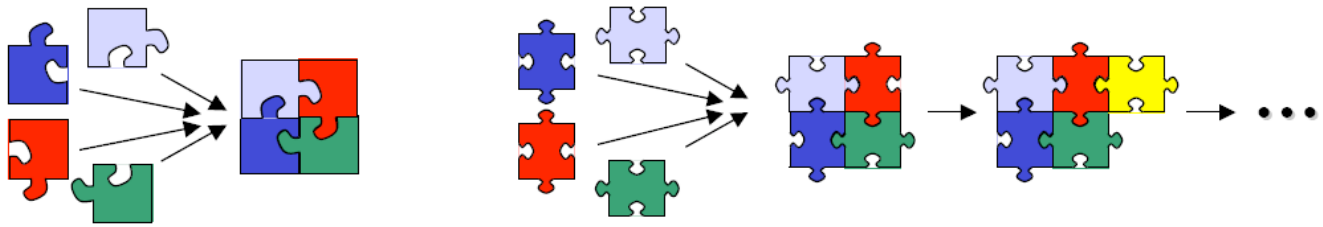
Abbildung: Metalevel understanding collected into metaobject protocols (MOPs) [McKinley, P.K., et al.]

Vorhandene Technologien - Komponenten-basiertes Design

Idee: Software ist ein Netzwerk aus konkurrierenden Komponenten, die über Konnektoren verbunden sind

dynamic recomposition

Nötig (nicht hinreichend) für Selbst-Adaption!



(a) Static recomposition. (b) Dynamic recomposition.

Abbildung: Component- based design enables static and dynamic recomposition [McKinley, P.K., et al.]

When to compose?

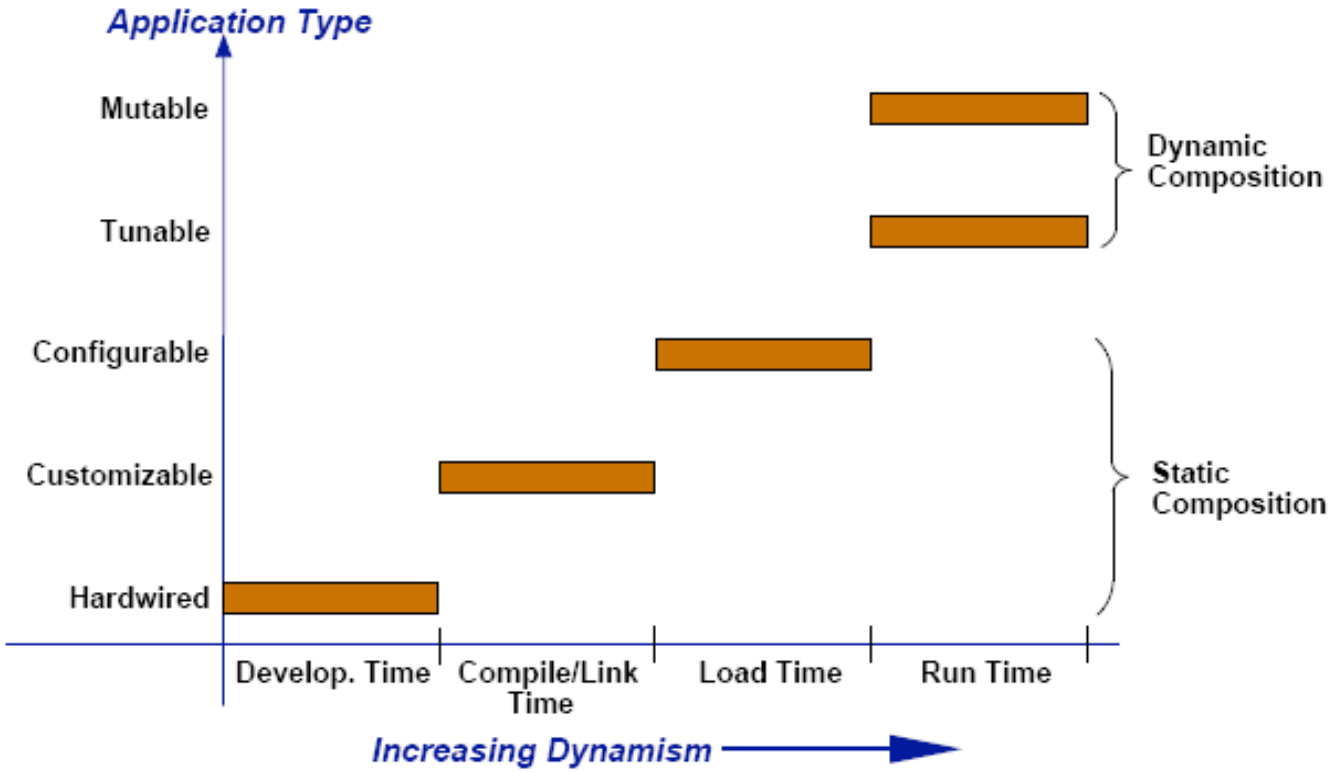


Abbildung: Classification for software composition using the time of composition or recomposition [McKinley, P.K., et al.]

Where to compose?

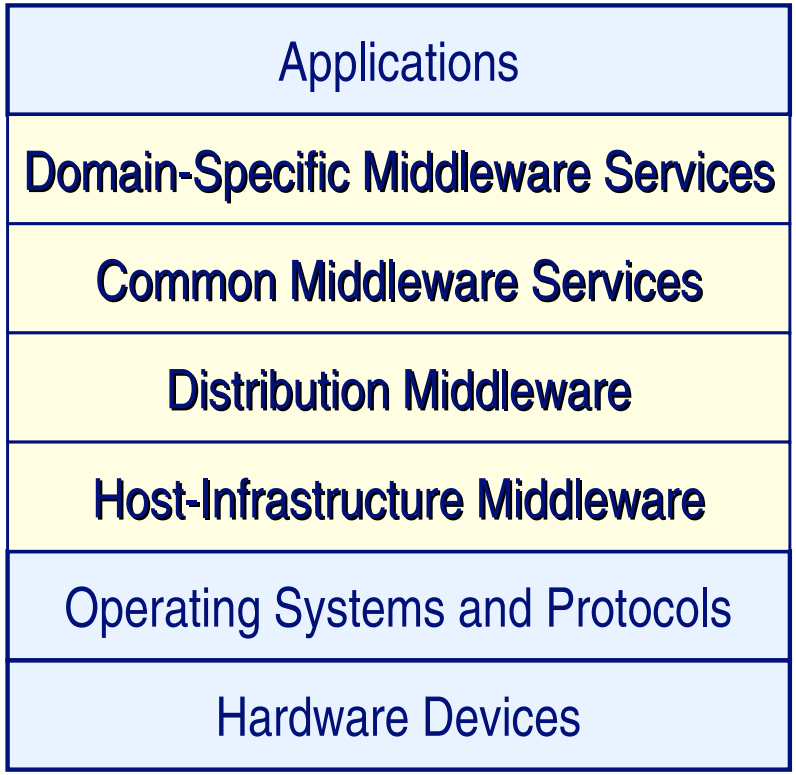


Abbildung: Schmidt's Middleware Layers [Douglas C. Schmidt]

Adaptation policies

Techniken, für Auswahl, Berechnung oder Ableitung einer neuen Konfiguration, die den aktuellen Status oder Kontext des Systems erfüllt.

- Situation-action rules
  - Spezifizert genau was zu tun ist
  - z.B. IF (RT>100msec) THEN (increase CPU by 5 %)
- Goal-based
  - Spezifizert den gewünschten Zustand: RT $<$ 100 msec
  - System bleibt es selbst überlassen das gewünschte Ziel zu erreichen (Planungsproblem)
- Utility-based
  - utility-function: Bewertet alle machbaren/möglichen Systemzustände (Vorhersage)
  - U_R(CPU)=U(RT(CPU))
    - zugrunde liegendes Modell RT(CPU)
  - Adaption wird zum Optimierungsproblem: Ermittle brauchbare Werte für die CPU, damit U maximal wird!

Agenda

Fazit und offene Fragen

- Software-Adaptivität (in allen Disziplinen) ist und bleibt sehr aktuell
  - Ausschreibungen und Tagungen zu Forschungsprojekten
- Kompositionelle Adaption klingt sehr vielversprechend!
- Benutzbarkeit
  - Ist Adaption vom Benutzer gewünscht?
  - Wie empfindet der Benutzer die Adaption?
  - Wie oft darf sich eine Anwendung anpassen, ohne dass es als störend empfunden wird?
- Test und Validierung
- Sicherheit
  - Wie schützt man potenziell persönliche Kontextinformationen?
  - Schutz vor böswilliger, ungünstiger Adaption

Agenda

Ausblick

- Interessant für mobile und pervasive Computing
- Es gibt noch viele andere Technologien
- Agenten klingen sehr vielversprechend!

Agenda

Literatur I

McKinley, P.K., et al.

A taxonomy of Composational Adaptation.

Tech report, Software Engineering and Network Systems Laboratory, Michigan state university. 2004.

McKinley, P.K., et al.

Composing Adaptive Software.

IEEE Computer Volume 37, Issue 7, July 2004 Page(s):56 - 64.

Douglas C. Schmidt

Middleware for real-time and embedded systems.

Comm. ACM, June 2002, pp. 43-48.

Jeffrey O. Kephart, Rajarshi Das (IBM T.J.Watson Research Center)

Achieving Self-Management via Utility Functions.

Internet Computing, IEEE, Volume: 11 Issue: 1 Jan.-Feb. 2007 Page(s): 40-48.

Literatur II

Jacqueline Floch (SINTEF)

D2.2 Theory of adaptation.

http://www.intermedia.uio.no/display/madam/Home , Dec 2006, referenziert 18.05.2008.

Kurt Geihs, et al.

D3.3 UML Modelling Elements and Approach for Application Adaptation (Final).

http://www.intermedia.uio.no/display/madam/Home , Dec 2006, referenziert 18.05.2008.

Alessandro Mamelli (Hewlett-Packard)

D4.2 MADAM Core Middleware Platform and Middleware Services.

http://www.intermedia.uio.no/display/madam/Home , Dec 2006, referenziert 18.05.2008.

IntegraSys (Pedro Ruiz, José Manuel Sanchez) Condat (Rolf Fricke)

MADAM development framework evaluation (Final).

http://www.intermedia.uio.no/display/madam/Home , Apr 2007, referenziert 18.05.2008.

Literatur III

Geihs, Kurt

Selbst-Adaptive Software

Informatik-Spektrum, Volume 31, Number 2, page 133–145 - April 2008.

Mourad Alia, et al.

A Utility-based Adaptivity Model for Mobile Applications.

IEEE Computer Society, AINAW '07, 2007

Holger Mügge

Integrating aspect-orientation and structural annotations to support adaptive middleware.

ACM, Mai 2007

Christine R. Hofmeister

Dynamic Reconfiguration of Distributed Applications.

PhD thesis, Computer Science Department, University of Maryland

Robert Laddaga (DARPA BBN)

Self Adaptive Software - Problems and Projects.

IEEE Computer Society, Second International IEEE Workshop on Software Evolvability (SE'06), 2006

Diskussion

Vielen Dank für Eure Aufmerksamkeit!

Noch Fragen?