Stefan Kost

# GITK - Eine generische Architektur für multimodale Interfaces

## 1. Einleitung und Motivation

Ein Großteil der heutigen Anwendungen auf Rechnersystemen sorgt selbst für die mediale Umsetzung ihrer Interaktionskomponenten. Am Beispiel der Nutzerschnittstellen bedeutet dies in der Regel, dass eine graphische Schnittstelle (GUI) zur Verfügung gestellt wird. Eine solche Schnittstelle ist zudem oft auch noch fest in das Produkt integriert. Damit ist der Anwenderkreis auf sehende Nutzer am (stationären) Bildschirmarbeitsplatz eingeschränkt. Als Interaktionsmedium werden dabei hauptsächlich visuelle Darstellungen und damit verknüpfte Eingabegeräte (Maus und Tastatur) verwendet. Damit werden unter anderem die Möglichkeiten mobiler Systeme vernachlässigt und Nutzergruppen mit Behinderungen ausgeschlossen.

Aus diesen und vielen weiteren Gründen ist es notwendig, eine in dieser Hinsicht komplexere Softwarearchitektur zu wählen um diese Nachteile auszuschließen. UIMS stellen eine verbesserte Architektur nach dem Seeheim Modell zur Verfügung [GEPfaff, 1985]. Aktuellen Implementierungen fehlt allerdings eine mächtige Präsentationskomponente, welche in der Lage ist, multimodal zu präsentieren.

Verschiedene existierende Ansätze (siehe [UIML, 2000], [Dubinko et al., 2003]) realisieren eine multimodale

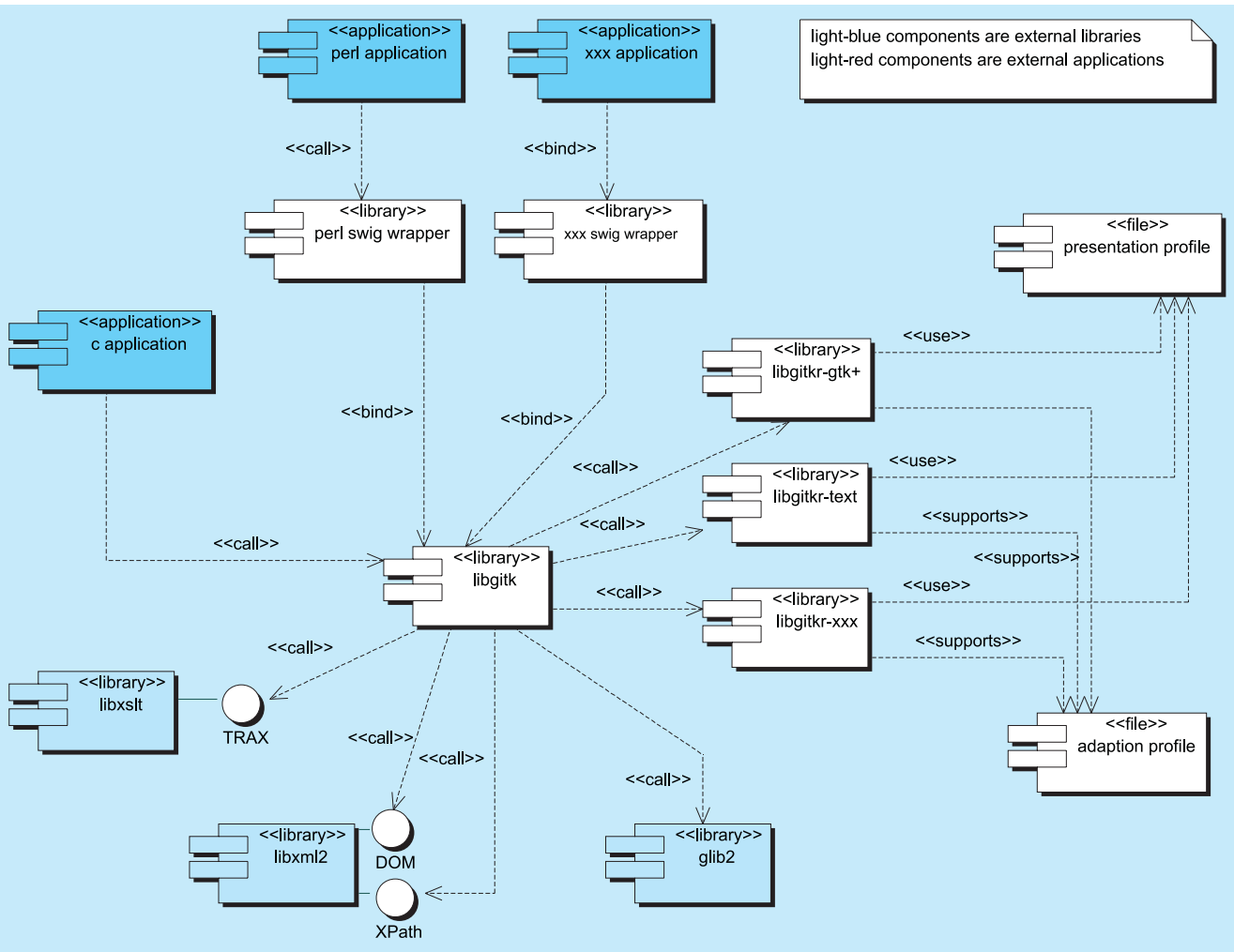


**Bild 1:** The gitk components

Adaption, benötigen jedoch Adaptionsprofile für jede Zielplattform und jede Anwendung.

## 2. Die Idee

Interfaces (ob für Menschen oder zu anderen Rechensystemen) müssen so abstrakt und vor allem medien-neutral beschrieben werden, dass eine konkrete Realisierung daraus generierbar ist. Dazu bedarf es einer konsequenten Trennung von Funktionalität und Präsentation der Schnittstellen. Für die Erzeugung eines Interfaces aus einer abstrakten Beschreibung und den Profilen von Anwender und Zielplattform werden Generatoren benötigt. Solche sollten jeweils eine gute Umsetzung für die entsprechenden Kommunikationsmittel (z. B. Ein- und Ausgabehardware oder Protokolle) erzeugen. Hierbei soll eine Lösung angestrebt werden, bei der die Anwendungen keine Adaptionsprofile für die jeweiligen Plattformen mitbringen müssen.

Weiterhin ist zu überlegen, inwieweit die mögliche Multimodalität durch Beschränkungen der Komplexität der Interfaces (in Hinsicht auf die mediale Abhängigkeiten) gesteigert werden kann. Diese Überlegung erscheint sinnvoll, da eine Vielzahl von Anwendungen mit einfach strukturierten Daten arbeitet.

## 3. Lösung

Im Rahmen meiner laufenden Dissertation habe ich zu Beginn das multi-dimensionale Adaptionsproblem untersucht und in die Teilprobleme: technische –, kulturelle – und persönliche Anpassung separiert. Diese Untergliederung kann auch als Rangfolge der Adaptionen verstanden werden (eine weitere Gliederungsmöglichkeit wird in [Sturm, 2002] vorgestellt). Außerdem habe ich die Merkmale unserer Sinnesorgane und die damit verbundenen Möglichkeiten der Informationsverarbeitung analysiert. Aus diesem Teil ergibt sich eine Aufstellung, welche z. B. kognitiven Einschränkungen bei der Anpassung an bestimmte Plattformen zu beachten sind.

Im zweiten Teil der Arbeit entwickle ich eine abstrakte XML basierende Sprache (GIML – Generalized Interface Markup Language) zur Schnittstellenbeschreibung, sowie eine Softwarearchitektur für deren Anwendung. Dabei wird auf etablierte Technologien (wie z. B. XSLT und XML Namespaces) gesetzt. Die Architektur basiert auf einer Gliederung in mehrere Schichten mit modularen Komponenten (siehe Bild 1). Aus diesen Komponenten wird eine Transformationskette aufgebaut, welche eine schrittweise Adaption durchführt. Adaption bedeutet hier, dass die zu Beginn medienneutrale Schnittstellenbeschreibung in jedem Adaptionsschritt mit Informationen aus Profilen (Nutzer- oder Plattformprofile) angereichert wird. Dabei wird jeweils eine Anpassung hinsichtlich eines Aspektes vorgenommen (z. B. Sprache oder Layout). Am Ende dieser Kette übernehmen domainspezifische Rendering-Plugins die Generierung des eigentlichen Interfaces. Im Endeffekt wird somit die eingangs genannte multidimensionale Anpassung durchgeführt.

## 4. Implementation

Parallel zur Entwicklung der Architektur, prüfe ich die Tragfähigkeit meines Ansatzes anhand einer konkreten Implementierung. Diese wird als Open-Source Projekt auf der Sourceforge Plattform entwickelt und ist unter http: //gitk.sf.net verfügbar. Die Implementation beinhaltet die Kernbibliothek, derzeit zwei verschiedene Renderer (einen graphischen via GTK+ und einen Text-basierten mit optionaler Sprachausgabe) und ein paar Beispiele. Weitere Renderer sind als Diplomarbeiten in Entwicklung (Telefon, Web via HTML und CSS und 3D via OpenGL).

Die Software ist sehr portabel, da wenig Abhängigkeiten zu anderen Bibliotheken existieren und die einzelnen Komponenten in der Programmiersprache C geschrieben sind. Derzeit wird sie unter Solaris und Linux entwickelt. [Kost, 2003]

## 5. Ergebnise und Ausblick

Auch wenn die aktuelle Implementation noch nicht alle konzipierten Funktionalitäten beinhaltet, können bereits einige Aussagen über die möglichen Anwendungsgebiete und der Qualität der Anpassungen getroffen werden:

- gut nutzbare Anwendungen: Informationsabfragen (Wetter, Bahnverbindungen, Adressen), Kommunikationsanwendungen (eMail, Chat), Zeitplaner (Kalender).
- inhärent nicht sinnvoll umsetzbare Anwendungen: stark medien-abhängige Anwendungen (Grafikbearbeitung, Videoschnitt, Soundbearbeitung).
- Qualität der Adaption: auch wenn erst wenige Beispiele existieren, sieht der Ansatz vielversprechend aus - vielfältige Transformationen sind möglich.

Derzeit wird viel an der Referenzimplementation gearbeitet. Das Ziel ist, aussagekräftigere Beispiele zu erhalten, damit Nutzertests möglich sind.

**Dipl. Inf. Stefan Kost,** HTWK Leipzig
Kost@imn.htwk-leipzig.de

### Literatur

Dubinko, M.; Klotz, L. L.; Merrick, R.; Raman, T. V.: XForms 1.0. http: //www.w3.org/TR/xforms/ (14 October 2003).

Pfaff, G. E.: User Interface Management Systems. Springer Verlag, Berlin Heidelberg New York Tokyo, 1985.

Kost, S.: GITK – Generalized Interface ToolKit. (2000-2003). http: //gitk.sf.net.

Sturm, C.: TLCC – Towards a framework for systematic and successful product internationalization. In: IWIPS 2002: *Proceedings of the 4th Annual International Workshop on Internationalization of Products and Systems.* Austin/Texas, 2002.

UIML: UIML - User Interface Markup Language (1999,2000). http: //uiml.org/