

Java TV

Seminar Medientechnik

Kristin Doppler

23.06.2003

Übersicht

- Einleitung
- Umgebungen
- Java TV API - Kategorien
 - Service- und Selektions-APIs
 - Definitionen
 - Packages
 - Service Selection API
 - Application Lifecycle APIs (Xlets)

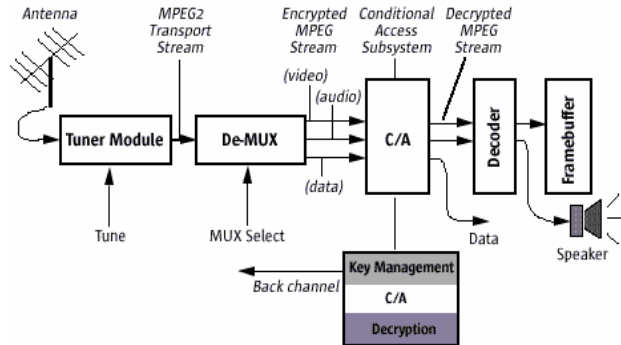
Einleitung

- Erweiterung der Java Plattform
- Vier Hauptbereiche der Technologie
 - Electronic Program Guides (EPGs)
 - Enhanced Video Broadcast
 - Interactive Video
 - Video-on-Demand

Hardware Umgebung

- Hardware- Kontext eines TV-Receiver
- Bestimmende Eigenschaft: Medien- und Daten-Übertragungsleitung
- Abstraktion: Java TV API unabhängig von Details der Hardware Umgebung

Übertragungsleitung



23.06.2003

Seminar Medientechnik
Kristin Doppler

5

Software Umgebung

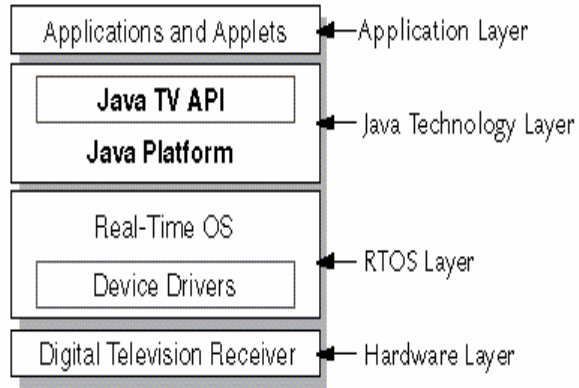
- Java-Plattform und Java TV API
- Beinhaltet Real-time Operating System (RTOS)
- Höchste Stufe: Zugriff auf darunterliegende Stufe (Java TV API und packages)
- Applikationen in Virtueller Maschine (VM) ausgeführt
- Niedrigste Stufe: Hardware-Kontrolle durch RTOS und zugehörige Bibliotheken anhand Sammlung von Gerätetreibern
- RTOS ermöglicht VM und Klassen-Bibliotheken zu implementieren
- Java TV API abstrahiert Funktionalität der System-Bibliotheken

23.06.2003

Seminar Medientechnik
Kristin Doppler

6

Java TV API Umgebung



23.06.2003

Seminar Medientechnik
Kristin Doppler

7

Java TV API Kategorien

- Service- und Selektions-APIs
- Broadcast Pipeline APIs und JMF
- Broadcast Daten APIs
- Application Lifecycle APIs (Xlet)

23.06.2003

Seminar Medientechnik
Kristin Doppler

8

Services und Service Informations - Definitionen (1)

➤ Service:

- Sammlung von Service-Komponenten
- können als Einheit kontrolliert werden
- besitzt Service-Information (SI)
- SI für Receiver verfügbar
- SI Eintrag eine von den bestimmenden Eigenschaften

Services und SI - Definitionen (2)

➤ SI:

- Information, die Inhalt des Service beschreibt
- Basis-Information, um Komponenten als kohärentes Ganzes zu präsentieren
- Meta-Information

➤ SI-Element:

- Objekt, das einen Teil der SI repräsentiert

Services und SI - Definitionen (3)

- Service-Komponente:
 - „mono-media“ Element, präsentiert ohne zusätzliche Information zu benötigen
 - Service besitzt eine oder mehrere Komponenten
 - kann von mehr als einem Service benutzt werden

- Service Locator:
 - Information, um Service in physikalische Adresse für Präsentation umzuwandeln
 - Locator-Objekte benutzt, um SI-Elemente zu referenzieren

Services und SI - Definitionen (4)

- SI-Datenbank:
 - speichert SI, zugänglich für TV-Applikationen

- SI-Manager:
 - Hauptzugangspunkt zur SI-Datenbank
 - berichtet Veränderung bezüglich der SI-Elemente
 - wandelt Service Locator in Meta-Daten um

SI Packages

- SI-Datenbank-Objekt-Modell: verschiedene Ansichten der SI
- Jede Ansicht als package in Java TV SI APIs repräsentiert:
 - Service
 - Navigation
 - Guide
 - Transport

SI Packages (2)

- Service package: javax.tv.service
 - Hauptzugangspunkt SI-Datenbank
 - Paket enthält Basisklassen für die anderen Pakete, wie die Schnittstellen Service und SIElement
- Navigation package: javax.tv.service.navigation
 - Klassen zur Navigation in existierenden Services

SI Packages (3)

- Guide package: [javax.tv.service.guide](#)
 - Klassen für EPG's
 - Programmabläufe, individuelle Programm-Ereignisse und Programm-Einschaltquoten
- Transport package: [javax.tv.service.transport](#)
 - MPEG-2-Lieferungs-Mechanismus

SI APIs

- bieten asynchronen Zugang zu Information, die nicht gecached wurde
- bieten flexiblen Mechanismus für zukünftige Erweiterung des SIElement

Service- und Selektions-APIs

➤ Definitionen:

- Service Context:
 - Umgebung, in der Service präsentiert wird
 - Receiver unterstützt einen oder mehrere Service Contexts
 - Innerhalb kann ein Service ausgewählt werden, für eine Präsentation zu einer Zeit

Service- und Selektions-APIs (2)

➤ Definitionen:

- Service Content handler:
 - verantwortlich für Darstellung bestimmter Teile eines Service
 - kann mehrere Service-Komponenten darstellen (dieselbe time clock)
 - Lebenszyklus gebunden an Lebenszyklus der Darstellung des Service

Service Selection API (1)

- Zweck: Mechanismus, um Services zu selektieren
- Klasse: ServiceContext-class
- High-level Repräsentation eines Tuners, seiner zugehörigen dekodierenden Hardware und seinem Zustand
- Kontrolle der Komponentendarstellung

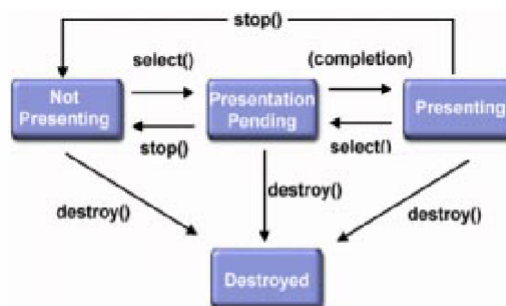
Service Selection API (2)

- Select ()-Methode: Versuch, Service zu präsentieren
 - Selektion ist asynchron
 - Fertigstellung angezeigt durch ein event-listener Mechanismus
 - Fehler angezeigt durch eine Ausnahme (Zeit von Select ()), sonst durch Event
- Während Präsentation können verschiedene Informationen über Service eingeholt werden

Service Selection API (3)

- 4 Zustände des Service Context
 - not presenting
 - presentation pending
 - presenting
 - destroyed
- Von jedem Zustand kann Select ()-Methode aufgerufen werden

Service-Kontext-Zustands-Modell



Xlets

➤ Xlet:

- Java-Applikation, die auf digitalen Fernsehgeräten läuft

➤ Applikations-Manager:

- Kontrolliert den Lifecycle eines Xlets, indem er die Zustandsänderungen signalisiert
- Kann Xlet zu jeder Zeit löschen

Xlets (2)

➤ Zustände:

- Loaded, Active, Paused, Destroyed
- Änderungen von Xlet selbst durchgeführt
- Xlet muss Manager mitteilen, wenn Zustand geändert wurde

➤ Xlet Context:

- Objekt für Zugang zu anderen Einrichtungen im System
- jedes Xlet hat ein XletContext Objekt, kann an spezifische Umgebung angepaßt werden

Xlet API (1)

- besteht aus zwei Interfaces: Xlet und XletContext
- Zustandsänderung entweder durch Xlet:
 - benachrichtigt Manager über XletContext Objekt
- oder durch Manager, indem er eine Methode aufruft

Xlet API (2)

- Halten sich an folgende Prinzipien:
 - Signalisieren Xlet, wenn Zustandsänderung erforderlich ist
 - Context wird Xlet gegeben, wenn es initialisiert wird
 - Xlet kann signalisieren, wenn es seinen Zustand geändert hat
 - Xlet kann signalisieren, wenn es fertig ist

Xlet Interface

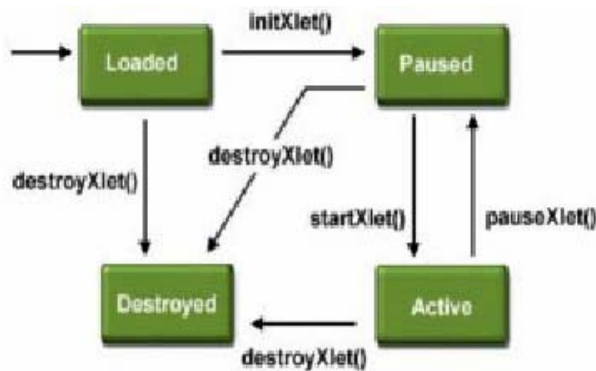
➤ Bietet Manager vier Methoden, um Xlet Zustandsänderungen zu signalisieren:

- `public void initXlet (XletContext ctx)`
- `public void startXlet ()`
- `public void pauseXlet ()`
- `public void destroyXlet ()`

➤ falls Änderung nicht möglich:

- `XletStateChangeException`

Zustands-Diagramm



Quellen

- www.java.sun.com/products/javatv
- www.developer.java.sun.com/developer/technicalArticles/javatv