

DVB-HTML
MIDP 2.0
Graphics Architectures for Non-Desktop
Devices

Pablo Cesar

pcesar@tml.hut.fi

<http://www.tml.hut.fi/~pcesar>

Outline

- Part I
 - DVB-HTML
- Part II
 - MIDP 2.0
- Part III
 - Graphics Systems in Embedded Devices

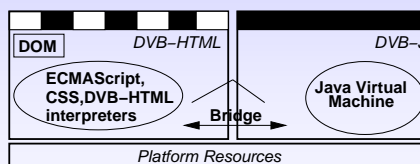
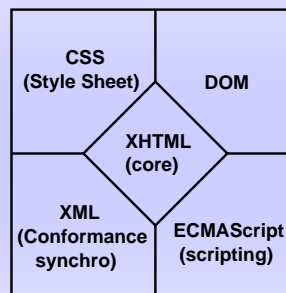
Part I: DVB-HTML

- Outline
 - Overview
 - Valid Tags: XHTML
 - Look: CSS
 - Accessing the Tree: DOM
 - Relationship with Java

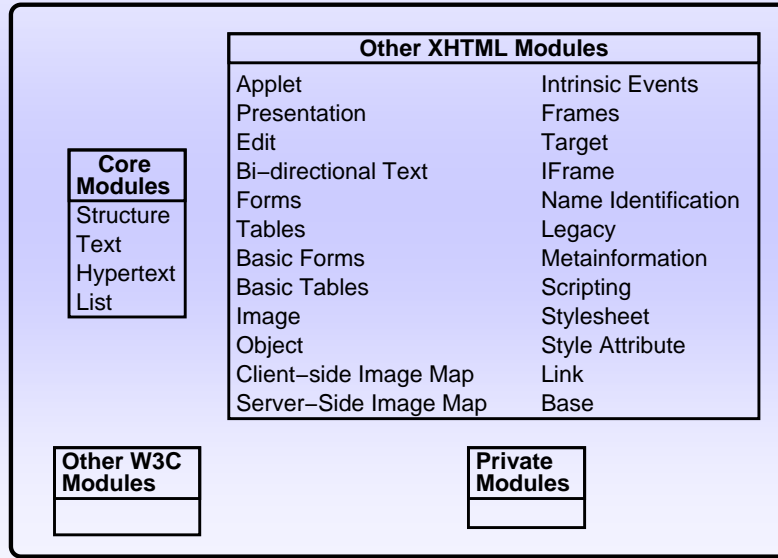
Overview

Modularisation of XHTML

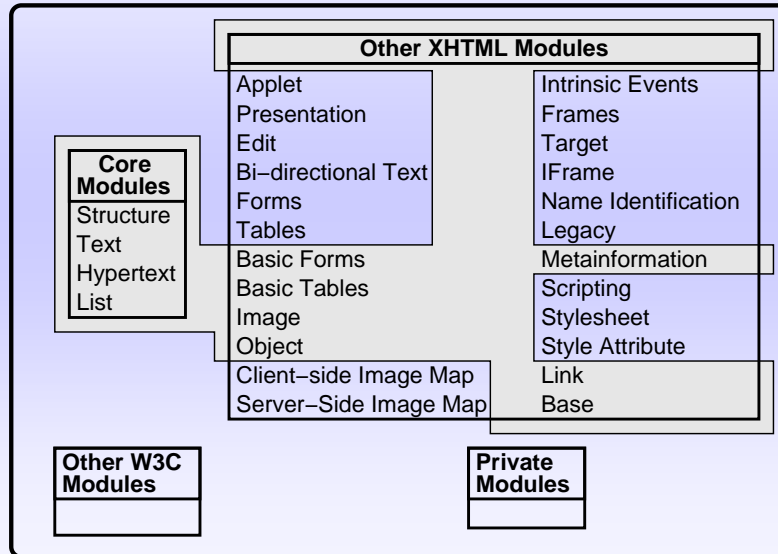
- Superset of XHTML Basic
- Comparable to XHTML 1.1
- CSS Support
 - CSS2 Compatible
 - DVB Extensions
- Media Types Support
 - DVB-J Applications
 - Video and Audio (MPEG-2)
 - ECMAScript
 - Images (png, gif, mpeg)
- DOM
 - Level 2 Modules
 - DVB Extensions



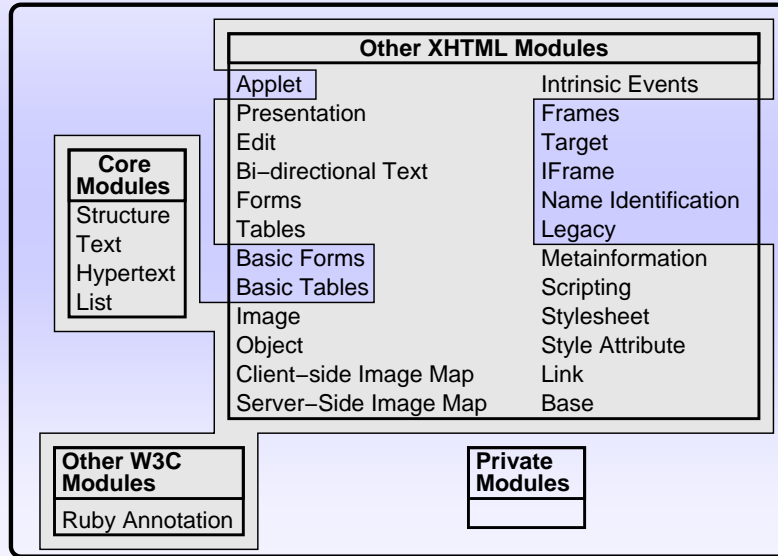
XHTML Modularisation



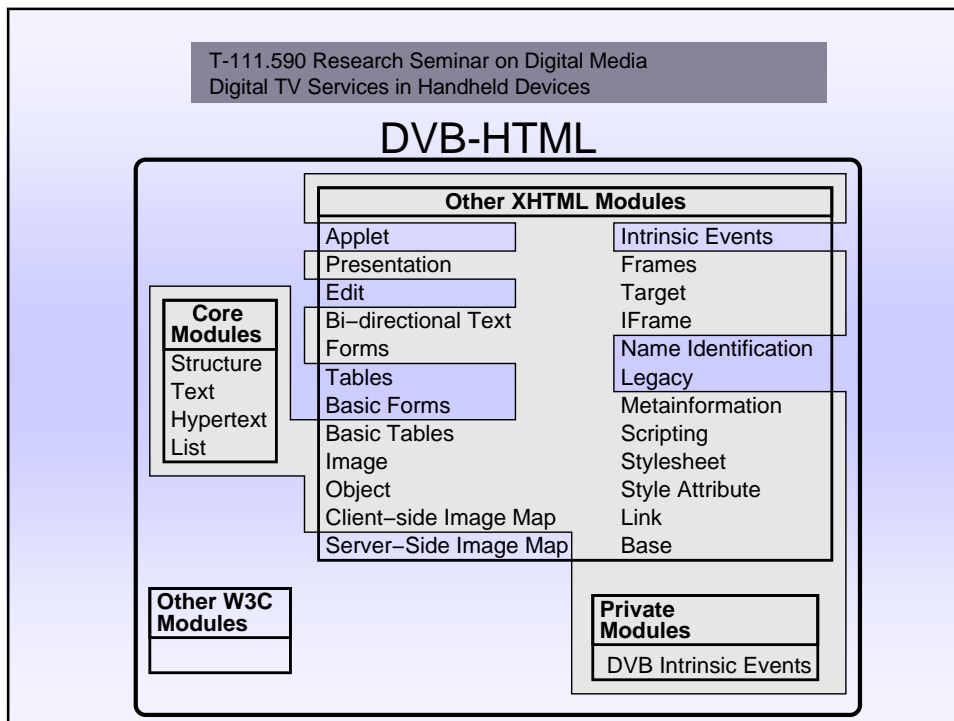
XHTML Basic



XHTML 1.1



DVB-HTML



CSS

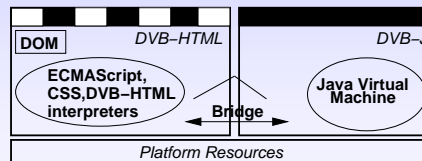
- **Compliant with CSS2:**
 - Properties: background, border, font, text, table...
 - Layout (all CSS2): absolute positioning, fixed positioning...
 - Selectors (all CSS2): hover, focus, active, lang...
- **CSS3:**
 - User Interface Module (directional navigation): nav-up, nav-down, nav-left, nav-right
 - Colour Module: opacity (transparency: alpha channel support)
- **DVB Extensions:**
 - New media-type: "dvh-tv", an extension of screen
 - Compose-rule: how to apply transparency
 - Clip-video: place a video (or a part of it) in a given location
 - Viewport: MHP graphics architecture integration (background + video + applications)

DOM

- **DOM Level 2 Modules**
 - Core, Views, Stylesheets
 - HTML not required: does not need to support HTML DTD
- **DOM Level 2 Events**
 - Events, UIEvents, MutationEvents required
 - HTMLEvents not required
- **MHP Extensions**
 - DVB-HTML: replacement of DOM Level 2 HTML (DVB-HTML DTD)
 - DVB Events and KeyEvents: binding remote control events and lifecycle, synchronisation with external video files
 - DVB CSS: for CSS Extensions
 - DVB Environment: platform characteristics

Relationship With Java

- application/dvbj media type
 - Similar to Applet
- EcmaScript
 - Using the bridge
 - Developer can access to MHP packages



MHP Web Page

“Is it true that DVB-HTML may be replaced in the future within theMHP1.1 specification by another TV-centric HTML-based language? In the spirit of global harmonization, it has been suggested that DVB-HTML should be modified or replaced by the resulting work of the CableLabs-OpenCable and ATSC-DASE harmonization. The discussions are still ongoing”

CENELEC Technical Report

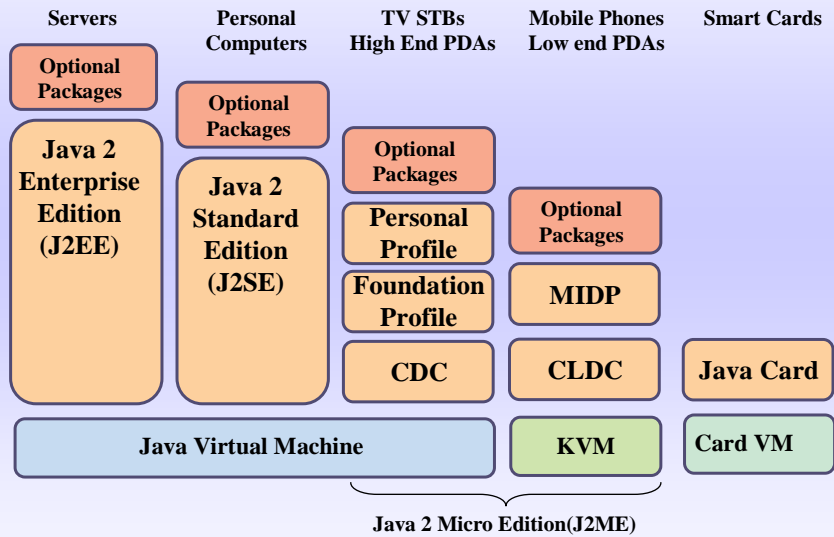
“The most likely candidate [...] is ACAP. ACAP [...] will be made available to DVB in order to for it to evaluate the possible replacement of its DVB-HTML presentation engine in MHP 1.1 specification. It should however be noted that ARIB has also proposed BML specification for consideration in this harmonization process. Therefore, there is insufficient certainty as this point in time to determine whether harmonization efforts with respect presentation engines will actually lead to a positive and useful result within a reasonable amount of time.”

Cesar's Doctoral Thesis

“As a final conclusion we will like to propose the following questions: What organization should be in charge of defining a standard for the declarative environment, DVB, CableLabs, ATSC, ARIB, W3C, SMPTE? Would not be better if all these groups joins, and harmonize, so the number of defined tags for the same purpose stop increasing unnecessarily?”

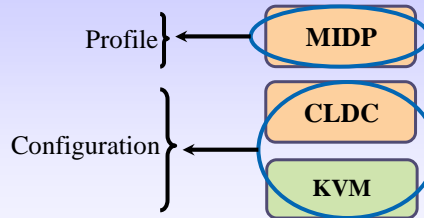
Part II: MIDP

- Java Overview (Editions/Configurations/Profiles)
- Java 2 Micro Edition (J2ME)
 - Connected Device Configuration (CDC)
 - Connected, Limited Device configuration (CLDC)
- Mobile Information Device Profile (MIDP)
 - Architecture
 - User Interface
 - Multimedia
 - Problems



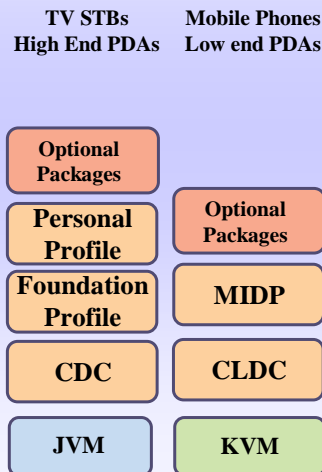
Java Overview

- Nowadays, trying to target all kind of computer devices
- Editions:
 - Java 2 Enterprise Edition (J2EE): for servers and enterprise computers
 - Java 2 Standard Edition (J2SE): for servers and personal computers
 - Java 2 Micro Edition (J2ME): for embedded devices, PDAs, mobile phones, and Digital television set-top boxes
 - Java Card: for smart cards
- Profile
 - Requirements for a specific vertical market of devices (set of APIs)
- Configuration
 - Minimum platform for a horizontal grouping of devices (VM + core APIs)

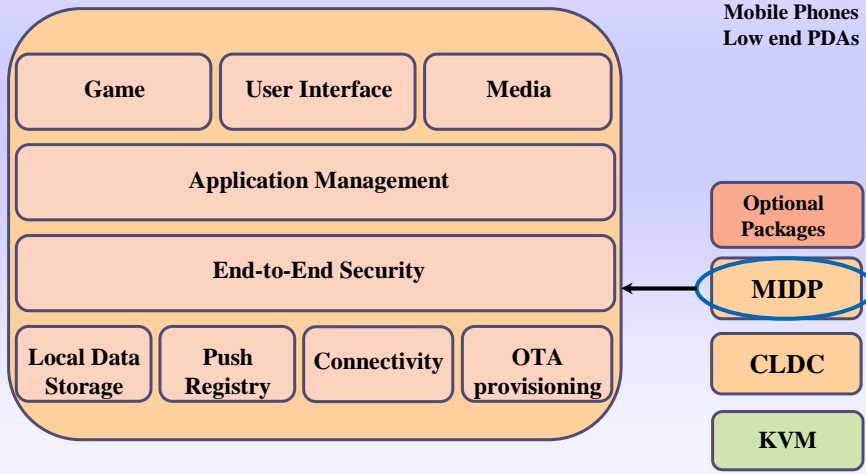


J2ME

- Defines two Configurations:
 - CDC: High end consumer devices
 - Processor: 32 bits
 - RAM Java Memory: around 2MB
 - ROM Java Memory: around 2.5MB
 - CLDC: Low end consumer devices
 - Processor: 16 bit/16 MHz or higher
 - Java total memory: 160-512 KB
 - Power: Limited power
- CDC (Connected Device)
 - Personal Profile
 - Adds support for lightweight AWT
 - Foundation Profile
 - Basic application APIs (no GUI)
- CLDC (Connected Limited Device)
 - Mobile Information Device Profile (MDIP)
 - Application APIs + GUI APIs



MIDP Architecture



MIDP Architecture

- Basic Layer
 - Local data storage
 - Persistent storage of data
 - Push Registry
 - Allows MIDlets to be launched in response to incoming network connections (e.g., alerts)
 - Connectivity
 - Connection for datagrams, sockets, and server sockets
 - OTA provisioning
 - Simplifies the way applications are delivered to consumers
 - Ability to dynamically deploy and update applications over-the-air (OTA). How applications are discovered, installed, updated...
- Second Layer
 - End-to-End security
 - MIDP provides a robust security model: http and https connections, and public key management

MIDP Architecture

- Third Layer
 - Application Management
 - Applications are called MIDlets, manager in charge of controlling their state
- Higher Layer
 - Game
 - Specific game API for developers
 - User Interface
 - Both High Level (ready made widgets), and high level API (developer can paint on the screen)
 - Media
 - Audio utilities API

MIDP User Interface

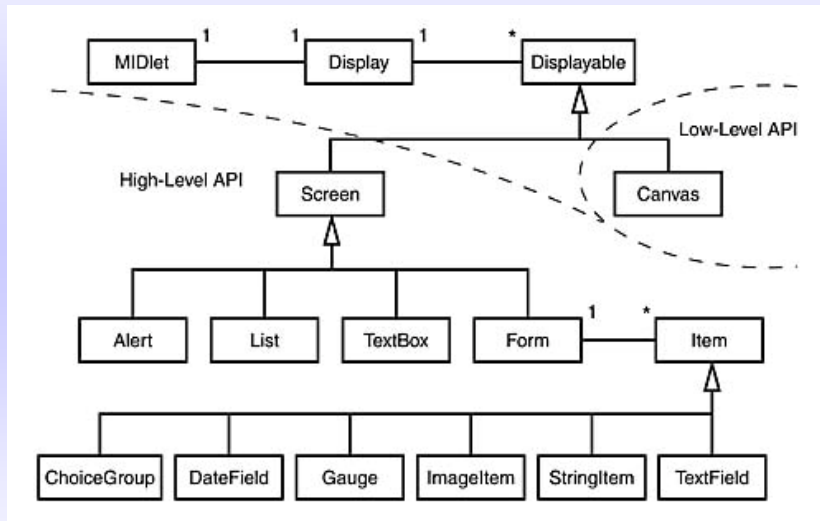
- User interface for handheld devices are different from PCs
 - Smaller display size
 - Input device not always include pointing device
- MIDP is not a subset of AWT!!!!!!
 - AWT is designed for PCs
 - AWT assumes certain interaction models (e.g., mouse)
 - AWT assumes the use of Windows (drag, move, resize)

MIDP User Interface

- **Basic Class (Display):** output device of the mobile phone
 - 1 display -> multiple Screens
 - 1 Application -> 1 Display
- **Basic interface (Displayable):** each screen of the services
 - 1 Application -> multiple Displayable objects
- **Two kind of Displayable Objects (Cannot be mixed):**
 - **Screen:** High Level API, each MIDP application has a Display in which a single screen is shown (title, multiple commands, ticker)
 - **Canvas:** Low Level API, it is extended for drawing

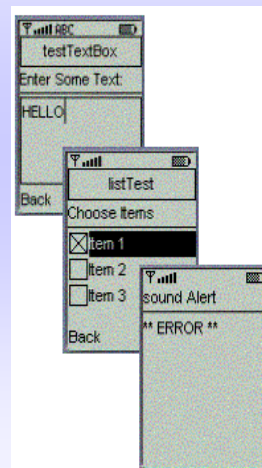
MIDP User Interface

- **High Level API**
 - Intended for applications where portability is important
 - High Level widgets, developer has no control on their look (appearance) and feel (interaction)
- **Low Level API**
 - Intended for applications where portability is not as important as control over the graphics
 - Developer has full control over what is drawn, where, and how



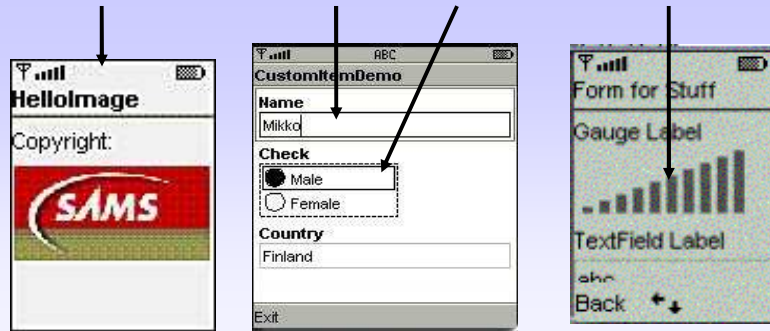
MIDP User Interface: High Level API

- **TextBox:**
 - Screen that allows the user to enter and edit text
- **List**
 - Screen that contains a list of choices
 - Implicit: like a menu
 - Exclusive: select one element (radio buttons)
 - Multiple choice: select many elements (check boxes)
- **Alert**
 - Screen that shows a message and an optional image to the user



MIDP User Interface: High Level API

- Forms
 - Screen that contains a combination of items
- Items:
 - Components of a Form
 - ImageItem, StringItem, TextField, ChoiceGroup, DateField, Gauge



MIDP User Interface: Low Level API

- Developer extends Canvas class and override the paint method to create her own widgets
- Allows developers to:
 - Control what is drawn on the display
 - Handle primitive events (e.g., Key Released)
 - Access concrete keys and other input devices
- Similar to AWT's Graphics:
 - Drawing model: there is not composition of images, the canvas is visible in the display or not visible
 - Double buffer: canvas can be stored as a off-screen image buffer
 - Coordinte system: origin is the upper-left corner of the display
 - Translation: the coordinte system can be translated over X or Y axis
 - Clipping: clipping is possible (so, no modifications are done over constant pixel values)
 - Color model: both gray scale (0 to 255) or color (24 bits)
 - Fonts: requested to the device (never created)

MIDP Multimedia

- Game API (MIDP 2.0)
 - GameCanvas: subclass of Canvas with specific game functionalities
 - Layer: visual element of the game (abstract class)
 - Sprite: animated layer that can display several graphical frames
 - TiledLayer: enables the creation of large areas of content, but at a low resource cost
 - LayerManager: to control the layers and the user's view
- 3D Graphics API (optional package)
 - Two APIs for displaying 3D content
 - Immediate mode API: create and manipulate 3D elements directly
 - Retained mode API (scene graph): load and display entire 3D scenes
- Mobile Media API (MMAPI) (optinal package)
 - Extends MIDP functionality by providing audio, video and other time-based multimedia support
 - It is not JMF
 - MIDP 2.0 includes the audio-only subset

Problems

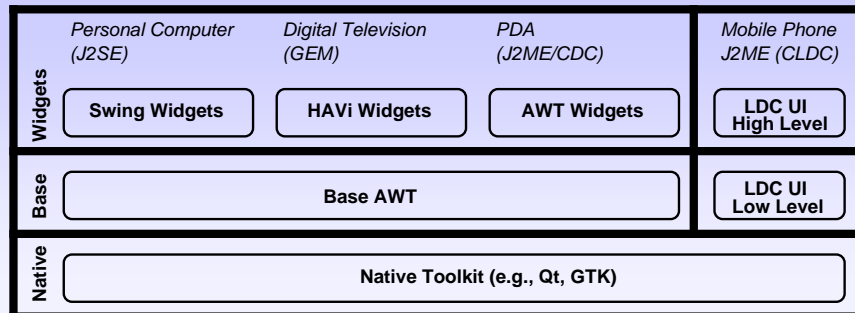
Size of graphics package:

- CLDC: 436 KB
- CDC: 527 KB

New Classes:

- Form (Container?)
- CustomItem (Component?)

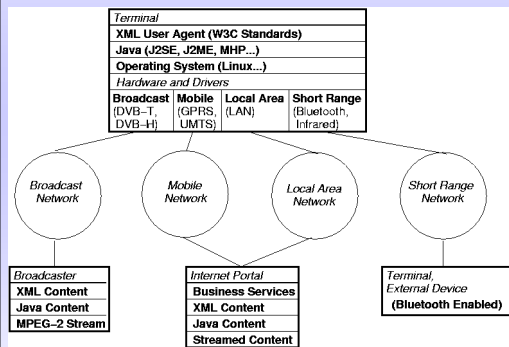
Interoperability:



Outline - Part III Graphics Systems in Embedded Devices

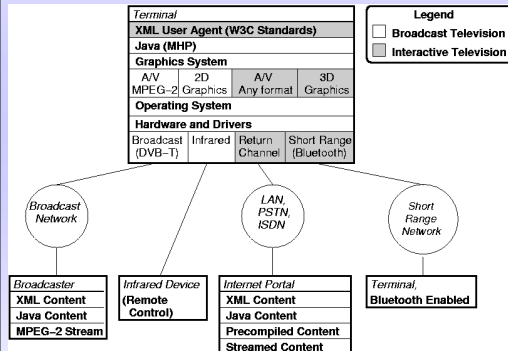
- Introduction
 - Multimedia Terminals Chaos
 - Digital Television
- Graphics Architecture
 - From Windows to Scene Based Graphics Architectures
- Reference Implementation
 - Ubik Architecture
 - Ubik Screenshots
 - High Level Languages Integration
 - High Level Languages Profiles
 - High Level Languages Screenshots

Introduction



- Variety of multimedia devices
 - Desktop (PCs)
 - Non-desktop (STB)
- Number of networks
 - Broadcast
 - Mobile
- Diversity of Content
 - MPEG-2
 - Java
 - XML based

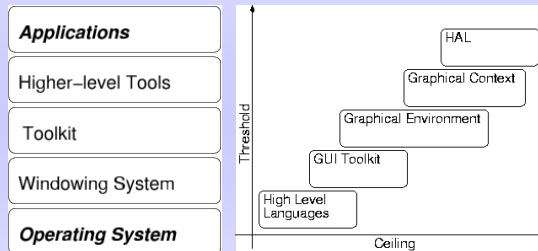
Introduction Digital Television



- Digital TV receivers today
 - Starting to show technological maturity
- Digital TV receivers tomorrow
 - Evolution, so different configurations depending on the targeted group
 - Broadcast
 - Only uses broadcast network
 - Interactive
 - Uses, as well, interaction channel

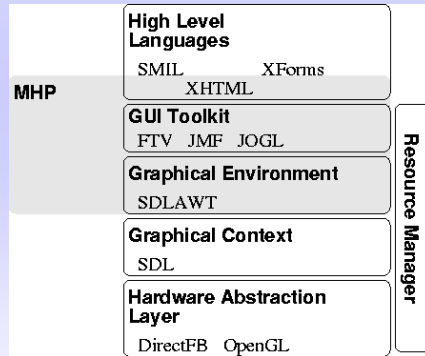
Graphics Architecture

- Lack of pointing device
- Seamless integration of video, 2D and 3D objects
- Screen composed of multimedia objects
- A layered architecture, so developers can implement at any level
- Solutions such as X-Windows are too big



- HAL: an unified way to access hardware (hardware acceleration)
- Graphical Context: cross-platform abstraction of the rendering region
- Graphical Environment: means to control different contexts
- GUI Toolkit: "ready-made" user interface widgets
- HLL: to develop simple services

Reference Implementation Ubik Architecture



- DVB-T reception and visualisation of the A/V stream
- HAL: DirectFB and OpenGL
- Resource Manager: to control the different processes
- Graphical Context: SDL
- Graphical Environment: SDLAWT (java.awt)
- GUI Toolkit
 - FTV: 2D widgets
 - JMF: other video than A/V
 - Java OpenGL: 3D Graphics
- HLL: X-smiles (XHTML, SMIL, XForms)

Reference Implementation Ubik Screenshots (1/2)

- Example composition of scenes:
 - 3D graphics object
 - A/V Stream
- Performance
 - Around 60 FPS



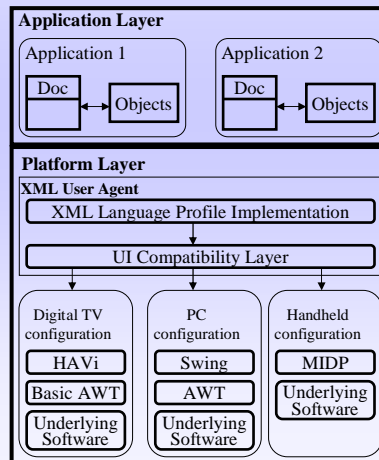
Reference Implementation Ubik Screenshots (2/2)

- Example native 3D Graphics
 - Some games downloaded from a portal



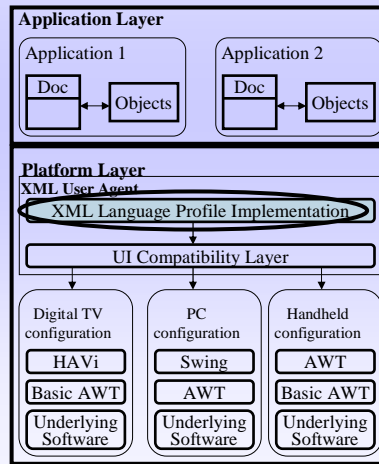
Reference Implementation High Level Languages Integration (1/4)

- Application Layer:
 - XML languages used for implementing interactive multimedia applications
- Platform Layer:
 - User Agent supports the language profile
 - UI Compatibility Layer (Component Factory)
 - Specific Device configuration



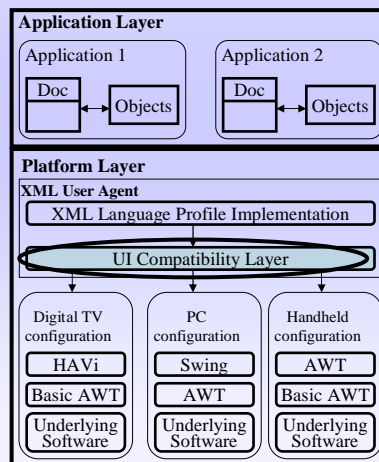
Reference Implementation High Level Languages Integration (2/4)

- Temporal Dimension
- Spatial Layout / Flow Layout
- Multimedia Objects Support
 - Continuous
 - Discrete
- User Interaction
 - Links
 - Validated Entry
 - Submission (server-side)

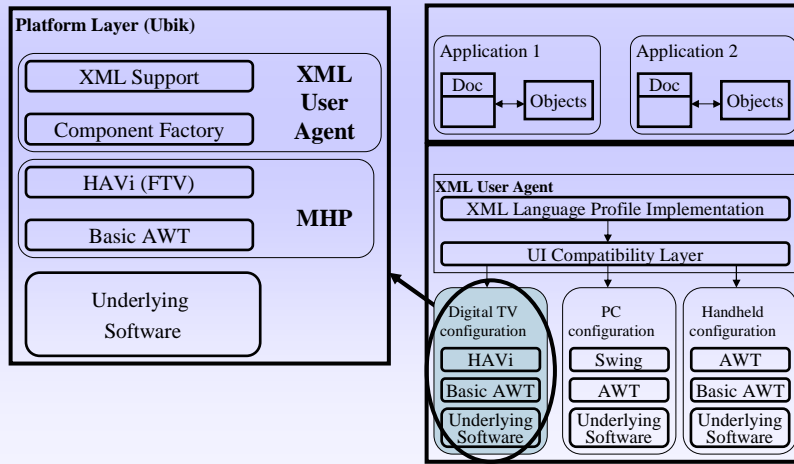


Reference Implementation High Level Languages Integration (3/4)

- Platform uses Java
- Most platform independent option
- The actual APIs for user interface development differ between multimedia terminals:
 - Digital TV: HAVi
 - PC: Swing
 - Handheld: MIDP



Reference Implementation High Level Languages Integration (4/4)



Reference Implementation High Level Languages Profiles

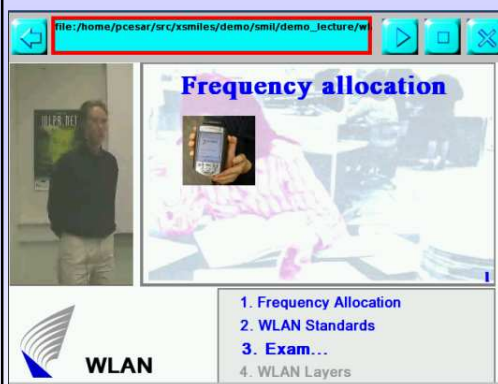
- | Internet Access | High End |
|---|---|
| <ul style="list-style-type: none"> • Requirements: <ul style="list-style-type: none"> - Access to Web pages - Limited navigation - Images, text - Small size • Proposed profile: <ul style="list-style-type: none"> - XHTML 1.0 - CSS | <ul style="list-style-type: none"> • Requirements <ul style="list-style-type: none"> - Access to complex Web applications - Interactive - Temporal Synchronisation - Video and audio - Bigger size • Proposed profile: <ul style="list-style-type: none"> - XHTML 2.0 + Timesheest - SMIL + XForms |

Reference Implementation High Level Languages Screenshots (1/2)



- XML Based Document
- Profile:
 - Interactive Access
- Target:
 - Information Services (e.g., Teletext)
- Languages:
 - XHTML 1.1
- Interaction:
 - Navigation
- Multimedia Objects:
 - Images
 - Text

Reference Implementation High Level Languages Screenshots (2/2)



- XML Based Document
- Profile:
 - High End
- Target:
 - Complex Applications (e.g., E-learning Portal)
- Languages:
 - SMIL (or XHTML+Timesheets)
 - XForms
- Interaction:
 - Buttons/Selections...
 - Navigation
- Multimedia Objects:
 - Images/Text
 - Video/Audio