



Universitatea Tehnică din Cluj-Napoca
Facultatea de Inginerie Electrică

Cap.12

PROIECTARE INTERFETE GRAFICE PENTRU DISPOZITIVE MOBILE



Curs: Proiectarea interfețelor grafice pentru monitorizare și control

CUPRINS

- Introducere
- SO Android
- Proiectare apps pentru Android



Introducere

Necesitati

- Proliferarea dispozitivelor mobile (telefoane,tablete,palmtop notebook, TabletPC)
- Medii computationale ubicue (*ubiquitous computing computing*)

Consecinte:

- Proiectare & implementare interfete interfete-utilizator independente de platforma
- Descriere interfete intr-o maniera universala si extensibila



Introducere

Solutii : specificarea in XML a **profilului** unui dispozitiv

- Caracteristici**: memorie, rezolutia ecranului ecranului, suportul pentru interactiunea cu utilizatorul etc.
- Platforma** (software) utilizata
- Preferintele** implicite de exploatare



Introducere

Profilul unui dispozitiv: trebuie sa asigure

- ❑ proiectare interfata independenta de dispozitiv
- ❑ efectuare teste evaluare componente sau prototip interfata

Caracteristici profil dispozitiv:

- ❑ Resurse hardware reduse (procesor, memorie, ecran la rezolutie redusa (Ex. 176 x 220, 240 x 320)
- ❑ Conectivitate cu alte dispozitive/PC
- ❑ Interacțiune via tastatura (limitată), *stylus (pen)*, *touch screen*,...
 - interacțiune multi-modala: tastatura versus *stylus*
 - afișare *landscape/portrait/square*



Introducere

Exemplu (XML & RDF: Resource Description Framework

```
<rdf:Description <!-- Caracteristici <!-- Platforma -->
  rdf:about="u <c:hardware> <c:platform>
  <!-- Informa <c:processor>.. <c:os version="...">...</c:os>
  <c:device ty <c:memory>...< <c:framework>...</c:framework>
  <c:name> <c:devices>...< </c:platform>
  <c:produc <c:power>...</c <!-- Preferintele utilizatorilor -->
  <c:owner: </c:hardware> <c:preferences>
  ... <rdf:Bag><rdf:li>
  </c:device> <rdf:Description rdf:about="#User">
  </rdf:Description> <c:input>...</c:input>
  </rdf:Bag> <c:output>...</c:output>
  </rdf:Description>
  </rdf:Description> </rdf:li>
  </rdf:Bag>
  </c:preferences>
```



Implementare interfețe pentru dispozitive mobile

Elementele de interfață se pot specifica în XML ⇒ descriere unei „metameta-interfețe”, ulterior transformată în construcții sintactice specifice fiecărei platforme disponibile pe un anumit tip de dispozitiv

Pe baza profilului de dispozitiv, reprezentarea de nivel înalt a interfeței va putea fi automat transformată în limbajul de specificare a interfeței pe care platforma/aplicația îl suportă: XHTML, WML, SVG, XUL, XAML,...



2. SO Android

SO Android: SO mobil bazat pe o versiune modificată de **Linux** (pentru gestiunea componentelor hardware, a proceselor și a memoriei) și **biblioteci Java** (pentru telefonie (audio/video), conectivitate, grafică, programarea interfețelor cu utilizatorul)

Caracteristici:

- este **cel mai popular SO** pentru dispozitive portabile
- produs **open-source** dezvoltat în prezent de Google, conceput pe ideea transformării dispozitivelor mobile în adevărate mașini de calcul.



SO Android

Istoric:

- ❑ construit în jurul unui nucleu Linux, (similar Chrome OS ,MAC OS) de o companie anonimă Android Inc.
- ❑ 2005 achiziționat de Google.
- ❑ 2007 înființare consorțiul comercial Open Handset Alliance pentru a stimula evoluția în domeniul dispozitivelor portabile; Lansare SO Android.
- ❑ 2008, apare pe piață primul terminal cu Android 1.0, HTC Dream, facilități : Wi-Fi, integrarea serviciilor Google, Android Market
- ❑ Diverse versiuni Cupcake, Donut, Eclair, KitKat, Lollipop
- ❑ in prezent, Android versiunea 8.0-Oreo

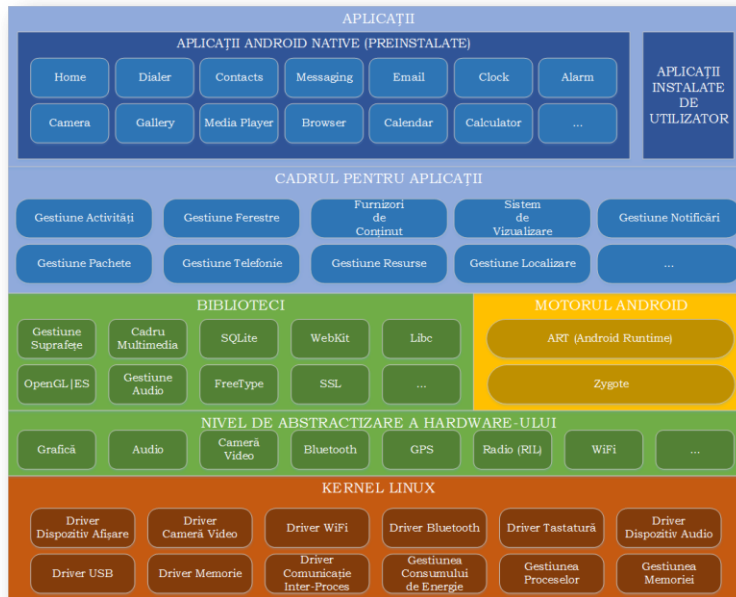


SO Android





Arhitectura SO Android



Arhitectura SO Android

1. Kernelul Linux:

- ❑ conține drivere pentru hardware (ecran, cameră foto, tastatură, antenă WiFi, memorie flash, dispozitive audio),
- ❑ este responsabil cu gestiunea proceselor, memoriei, perifericelor (audio/video, GPS, WiFi), dispozitivelor intrare/ieșire, rețelei și consumului de energie;



Arhitectura SO Android

2. **Bibliotecile** (user-space): conțin codul pentru principalele funcționalități a SO Android, fac legătura între kernel și aplicații. Includ:

- motorul open-source pentru navigare WebKit,
- biblioteca FreeType pentru suportul seturilor de caractere,
- baza de date SQLite utilizată atât ca spațiu de stocare cât și pentru partajarea datelor specifice aplicațiilor,
- biblioteca libc (Bionic),
- biblioteca de sistem C bazată pe BSD și optimizată pentru dispozitive mobile bazate pe Linux,
- biblioteci pentru redarea și înregistrarea de conținut audio/video (bazate pe OpenCORE de la PacketVideo),
- biblioteci SSL pentru asigurarea securității pe Internet și
- Surface Manager, bibliotecă pentru controlul accesului la sistemul de afișare care suportă 2D și 3D.



Arhitectura SO Android

3. **Motorul Android** rulează serviciile de platformă precum și aplicațiile care le utilizează, fiind reprezentat de:

a) **ART (Android Runtime):** mașina virtuală Java

- bytecode-ul este transpus în cod mașină la momentul instalării, și este executat direct de mediul dispozitivului mobil;
- fiecare aplicație Android rulează în procesul propriu, într-o instanță a mașinii virtuale ART, izolând astfel codul și datele sale



Arhitectura SO Android

3. Motorul Android : continuare

b) **Zygote**: procesul care gestionează toate aplicațiile, fiind lansat în execuție odată cu SO:

- inițial, creează o instanță a mașinii virtuale Java pentru sistemul de operare Android, în contextul căreia plasează serviciile de bază: gestiunea energiei, telefonie, furnizori de conținut, gestiunea pachetelor, serviciul de localizare, serviciul de notificări;
- atunci când este necesar să lanseze în execuție o anumită aplicație, se clonează, partajând astfel componentele sistemului de operare Android, astfel încât să se asigure performanța (timp de execuție) și eficiența (memorie folosită).



Arhitectura SO Android

4. **Cadrul pentru Aplicații**: expune diferitele funcționalități ale sistemului de operare Android către programatori, astfel încât aceștia să le poată utiliza în aplicațiile lor.

La nivelul de **aplicații** se regăsesc:

- produsele împreună cu care este livrat dispozitivul mobil (Browser, Calculator, Camera, Contacts, Clock, FM Radio, Launcher, Music Player, Phone, S Note, S Planner, Video Player, Voice Recorder),
- produsele instalate de pe Play Store sau cele dezvoltate de programatori.



Comunitate Android

Exista numeroși dezvoltatori :

[Google Android Training](#): conține tutoriale și exemple de clase grupate pe tematici, utile în deprinderea cunoștințelor de bază pentru dezvoltarea aplicațiilor Android.

[Stack Overflow](#): forum pentru programatori editat în mod colaborativ, conținând întrebări și răspunsuri la acestea (cele mai bune putând fi identificate cu ușurință prin voturile primite de la participanți).

[Android Discuss](#): listă de discuții monitorizată îndeaproape de echipa Android de la Google astfel încât reprezintă un loc unde pot fi clarificate numeroase nelămuriri



3. Implementare aplicații Android

Implementare aplicație Android :

- [kit-ul de dezvoltare pentru limbajul de programare Java](#)
- [SDK-ul de Android](#), pentru care se descarcă definițiile corespunzătoare unuia sau mai multor niveluri de API
- un mediu integrat de dezvoltare (IDE)
 - [Eclipse](#), cu plugin-ul ADT (Android Developer Tools)
 - [Android Studio](#)
- un dispozitiv pe care să se ruleze aplicațiile
 - un emulator
 - [Genymotion](#)
 - [Android Virtual Device](#) (livrat împreună cu SDK-ul de Android)
 - un telefon mobil cu sistemul de operare Android pentru care s-a dezvoltat aplicația



SO Android

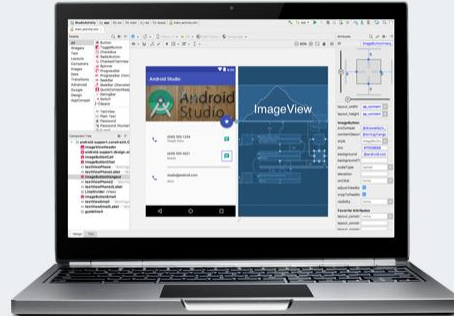
Android Studio

The Official IDE for Android

Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

DOWNLOAD ANDROID STUDIO
3.0.1 FOR WINDOWS (683 MB)



Exemple interfete app SO Android

Multi Converter: Google Apps

