PHILIPS

Device Independent User Interfaces

W. Dees
Philips Research
walter.dees@philips.com

WWW2003 Developer's Day

Contents

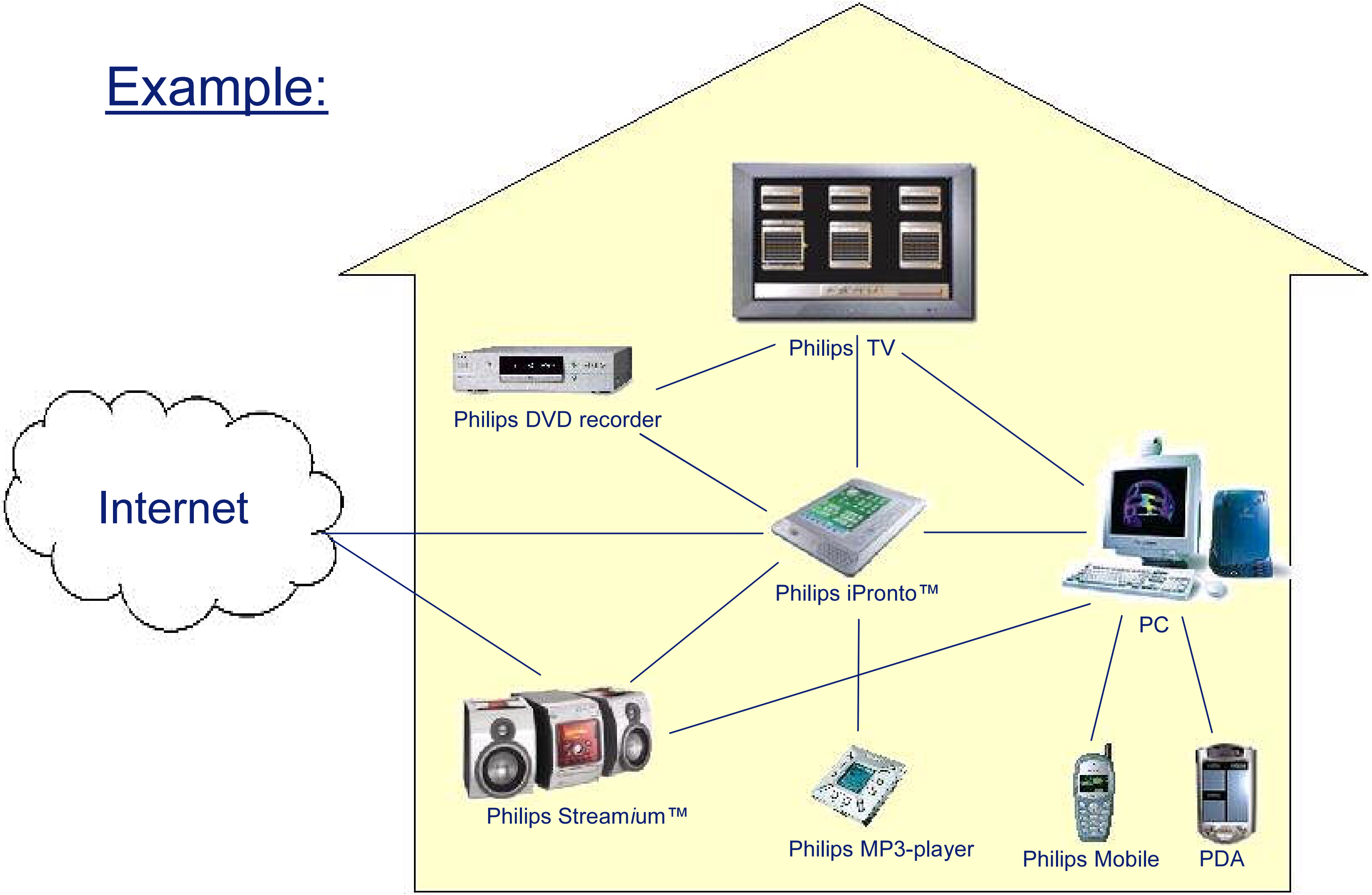
- Scope
- Our approach:
  - Abstraction
  - Multi-level stylesheets
- Challenges
- Conclusions

Contents

- Scope
- Our approach:
  - Abstraction
  - Multi-level stylesheets
- Challenges
- Conclusions

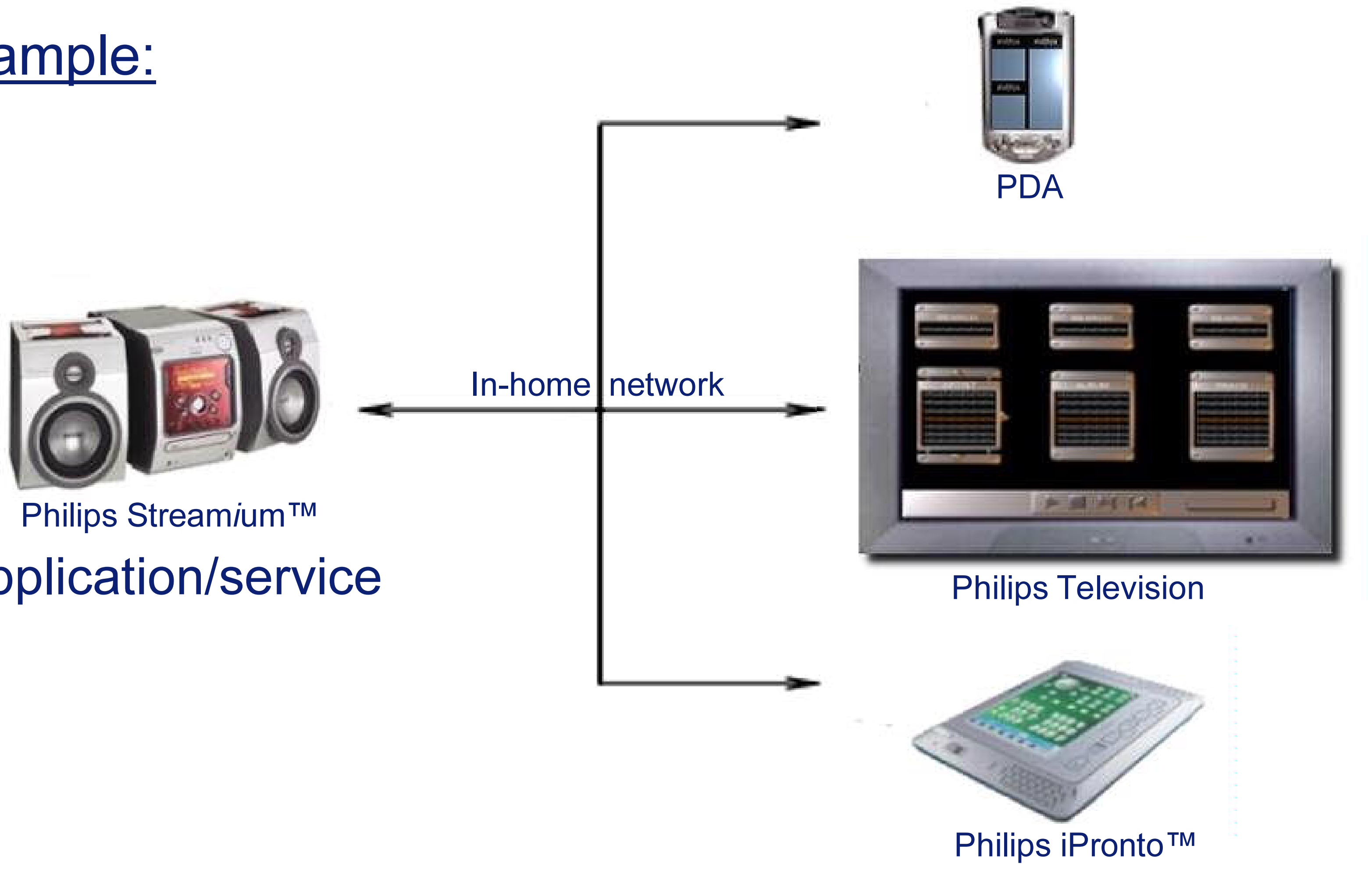
Philips Vision: The Connected Home

Example:



Philips Vision: The Connected Home

Example:



In-home network

PDA

Philips Television

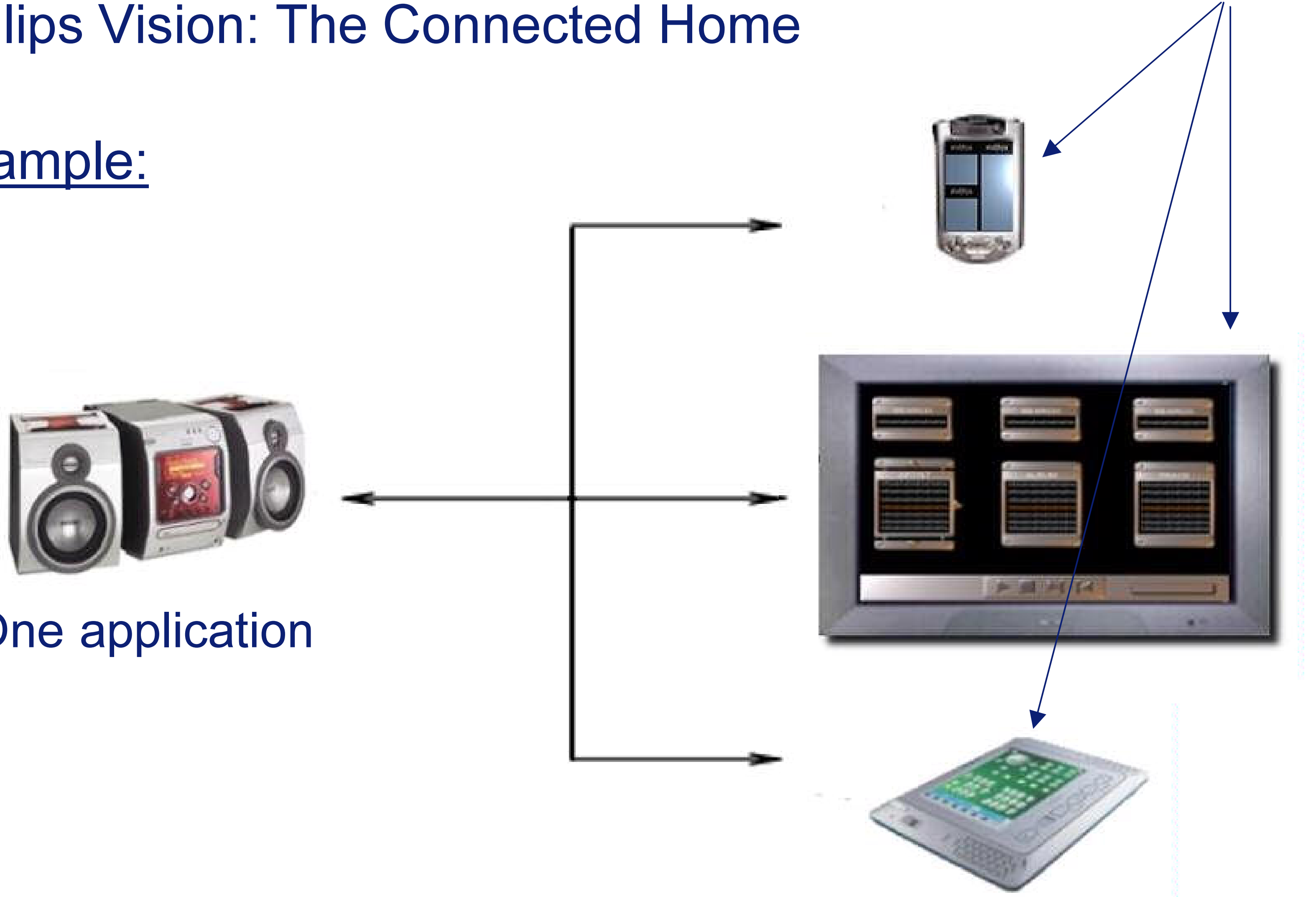
Philips iPronto™

Philips Stream*i*um™

Application/service

Philips Vision: The Connected Home

Example:

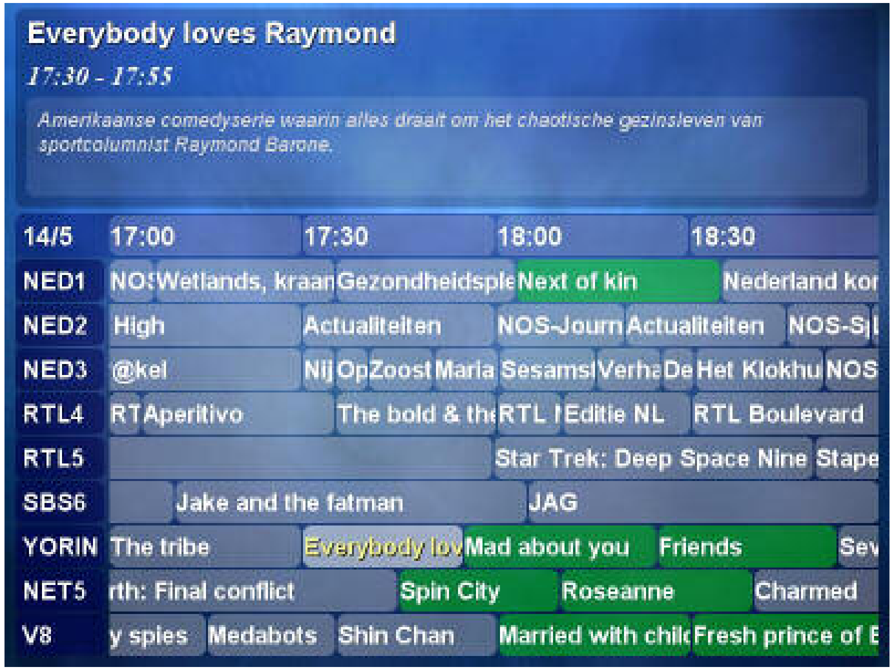


Different capabilities

One application

Many different user interfaces

Example: Electronic Program Guide (EPG)

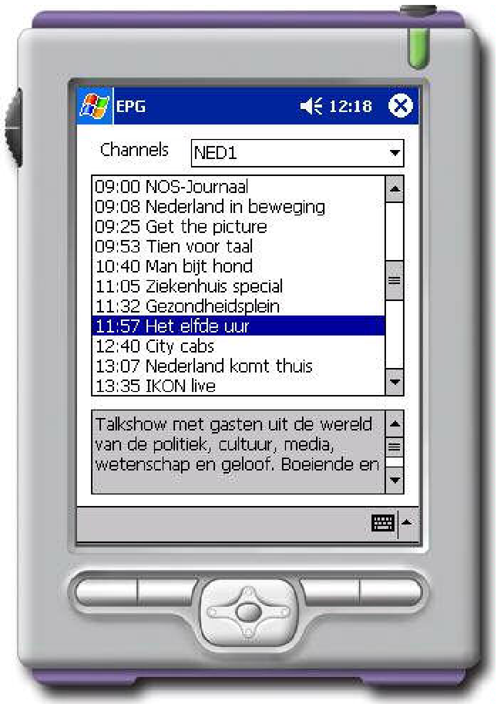


Everybody loves Raymond

17:30 - 17:55

Amerikaanse comedyserie waarin alles draait om het chaotische gezinsleven van sportcolumnist Raymond Barone.

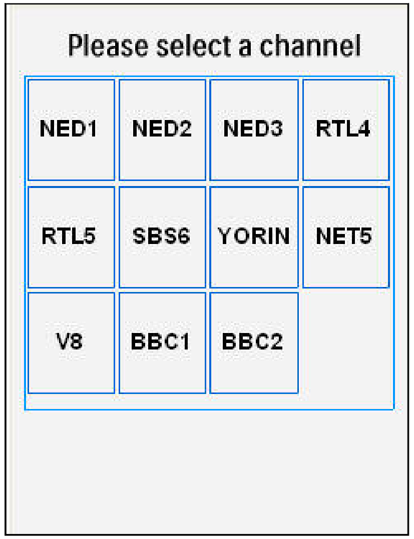
14/5	17:00	17:30	18:00	18:30
NED1	NO! Wetlands, kraan	Gezondheidsple	Next of kin	Nederland kor
NED2	High	Actualiteiten	NOS-Journ Actualiteiten	NOS-S|
NED3	@kel	Nij Op Zoost Maria	Sesamst Verha De	Het Klokhu NOS
RTL4	RT Aperitivo	The bold & the	RTL ! Editie NL	RTL Boulevard
RTL5			Star Trek: Deep Space Nine	Stape
SBS6	Jake and the fatman		JAG	
YORIN	The tribe	Everybody lov Mad about you	Friends	Sev
NET5	rth: Final conflict	Spin City	Roseanne	Charmed
V8	y spies Medabots	Shin Chan	Married with chil	Fresh prince of E



Channels: NED1

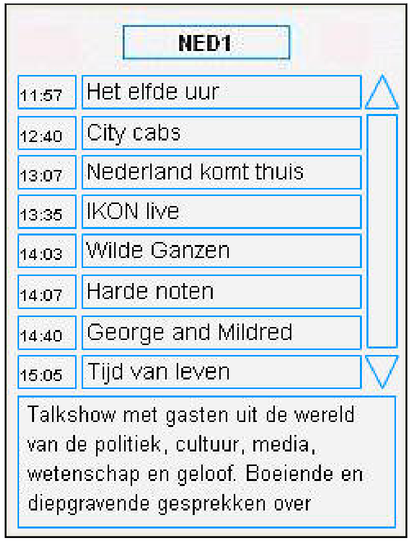
09:00 NOS-Journaal
09:08 Nederland in beweging
09:25 Get the picture
09:53 Tien voor taal
10:40 Man bijt hond
11:05 Ziekenhuis special
11:32 Gezondheidsplein
11:57 Het elfde uur
12:40 City cabs
13:07 Nederland komt thuis
13:35 IKON live

Talkshow met gasten uit de wereld van de politiek, cultuur, media, wetenschap en geloof. Boeiende en



Please select a channel

NED1	NED2	NED3	RTL4
RTL5	SBS6	YORIN	NET5
V8	BBC1	BBC2	

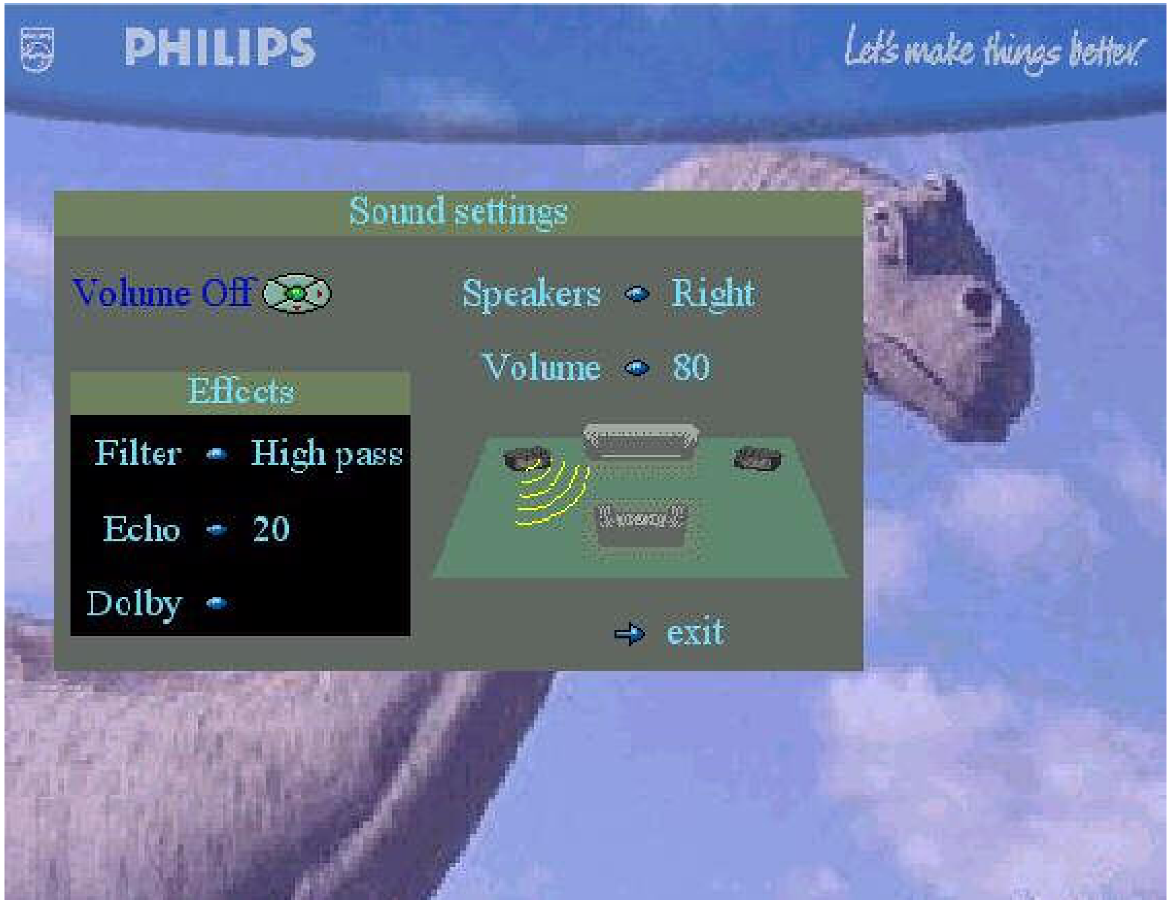


NED1

11:57	Het elfde uur
12:40	City cabs
13:07	Nederland komt thuis
13:35	IKON live
14:03	Wilde Ganzen
14:07	Harde noten
14:40	George and Mildred
15:05	Tijd van leven

Talkshow met gasten uit de wereld van de politiek, cultuur, media, wetenschap en geloof. Boeiende en diepgravende gesprekken over

Example 2: Device control



Sound settings

Volume Off

Speakers ◆ Right

Volume ◆ 80

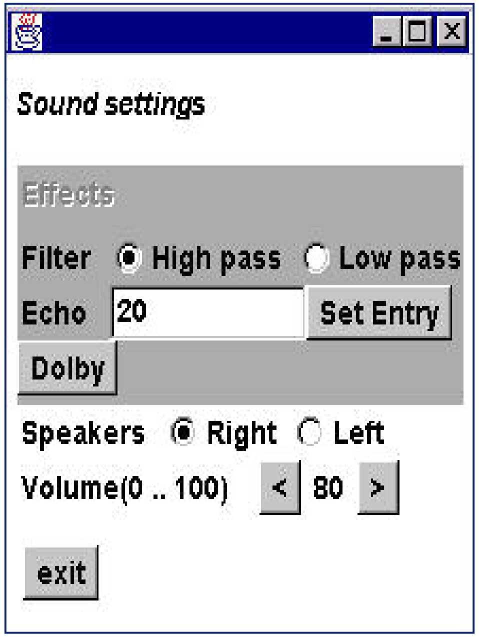
Effects

Filter ◆ High pass

Echo ◆ 20

Dolby ◆

➩ exit



Sound settings

Effects

Filter ● High pass ○ Low pass

Echo 20 Set Entry

Dolby

Speakers ● Right ○ Left

Volume(0 .. 100) < 80 >

exit

(Acknowledgements: Philips Design)

Diversity in UI devices can affect:

- Choice of interaction elements
- Navigation / Pagination
- Layout
- Content
- Delivery mechanism

In worst case: all of these are affected !

Device independent (DI) authoring:

- UI modelling for current and future target devices
- Preserve usability / tailored to target devices
- Off-line generation and run-time adaptation
- Should take less time than authoring UIs separately (i.e. affordable, not too complex)
- Similar to authoring DI web-sites

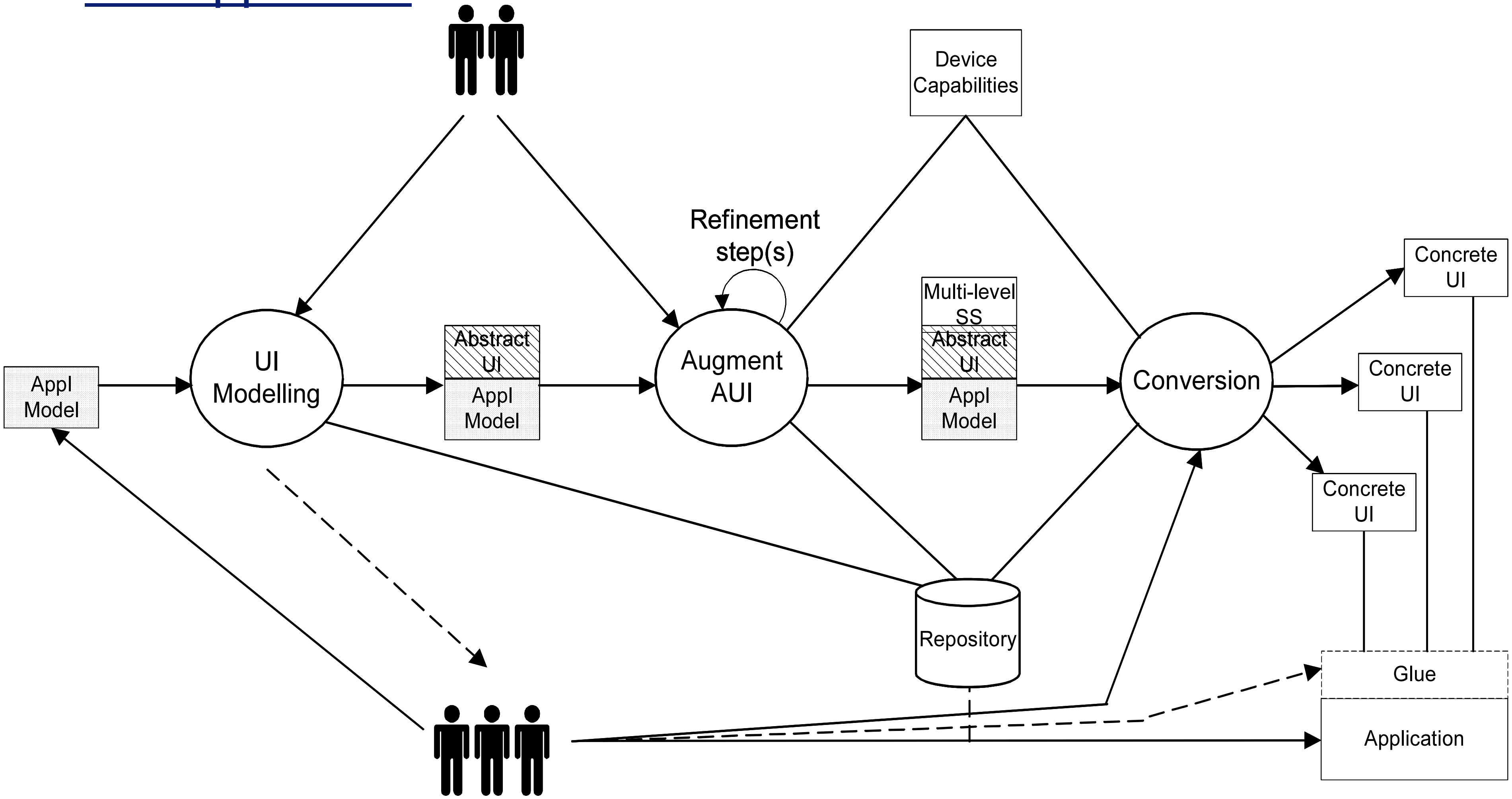
In this presentation:

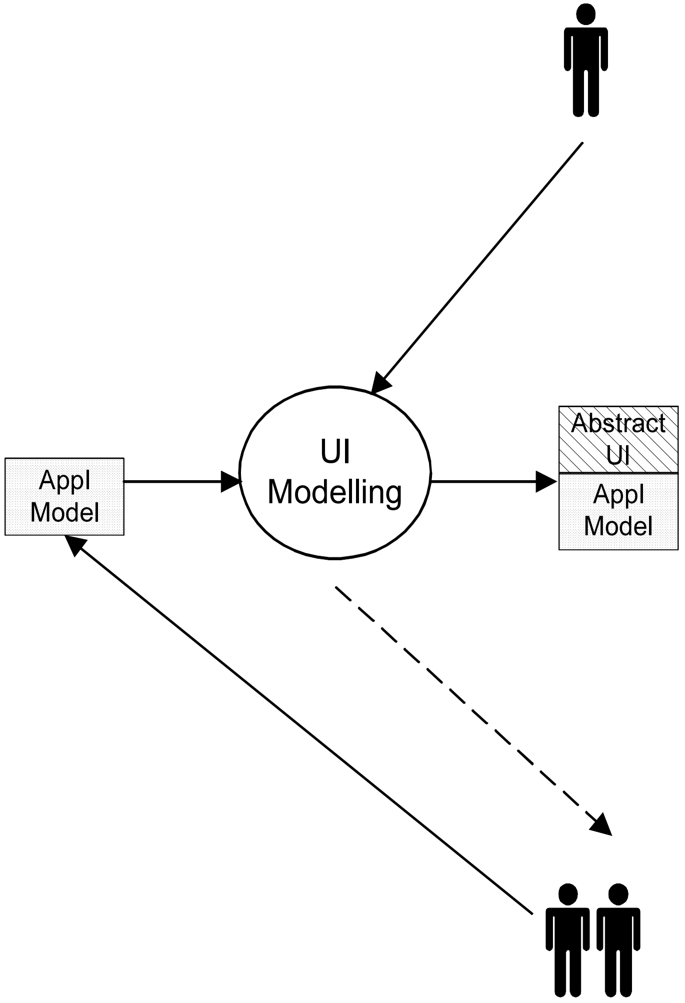
- Focus on interactive applications: content adaptation low priority, structuring high priority
- No context awareness
- Assumption: applications for which it makes sense to re-use on multiple UI devices (e.g. not HTML editor)

Contents

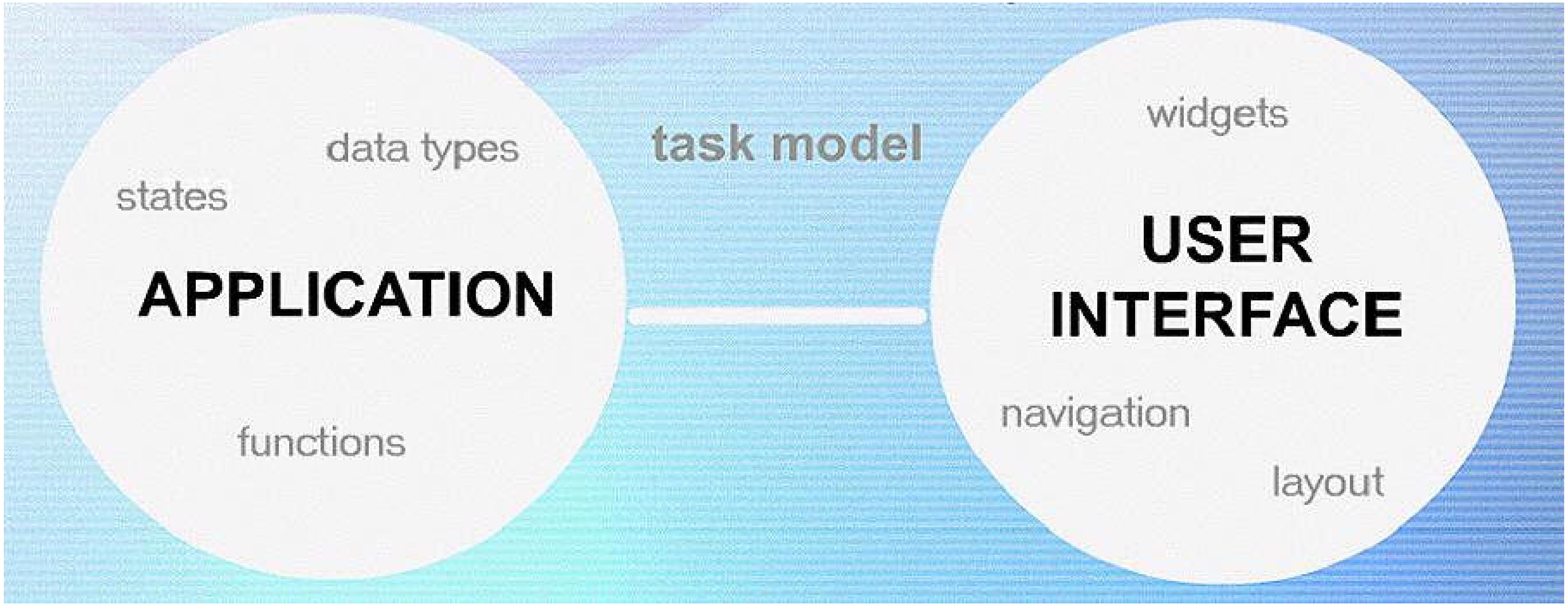
- Scope
- Our approach:
  - Abstraction
  - Multi-level stylesheets
- Challenges
- Conclusions

Our approach:




Appl Model
UI Modelling
Abstract UI
Appl Model

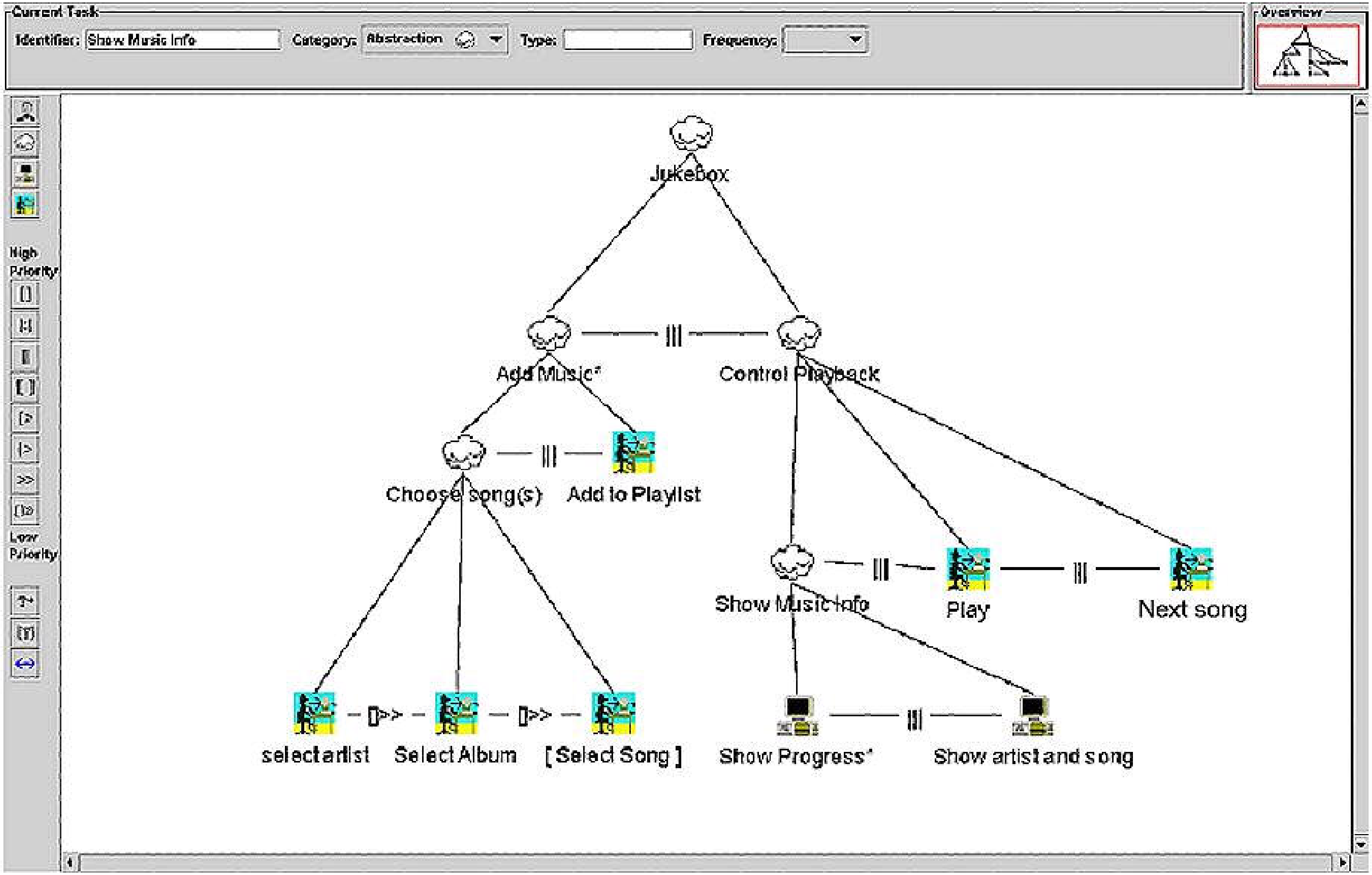
UI modelling:



Task model provides the glue between (UI-agnostic) application and user interface

UI modelling (continued):

- Create an abstract model of tasks (relevant for the intended user-group) and their relation to application
- Group tasks (hierarchy) and prioritize them
- Define vocabulary of terms used in UI.
- For example: ConcurTaskTree Environment (http://giove.cnuce.cnr.it/ctte.html, F. Paternò et al.)



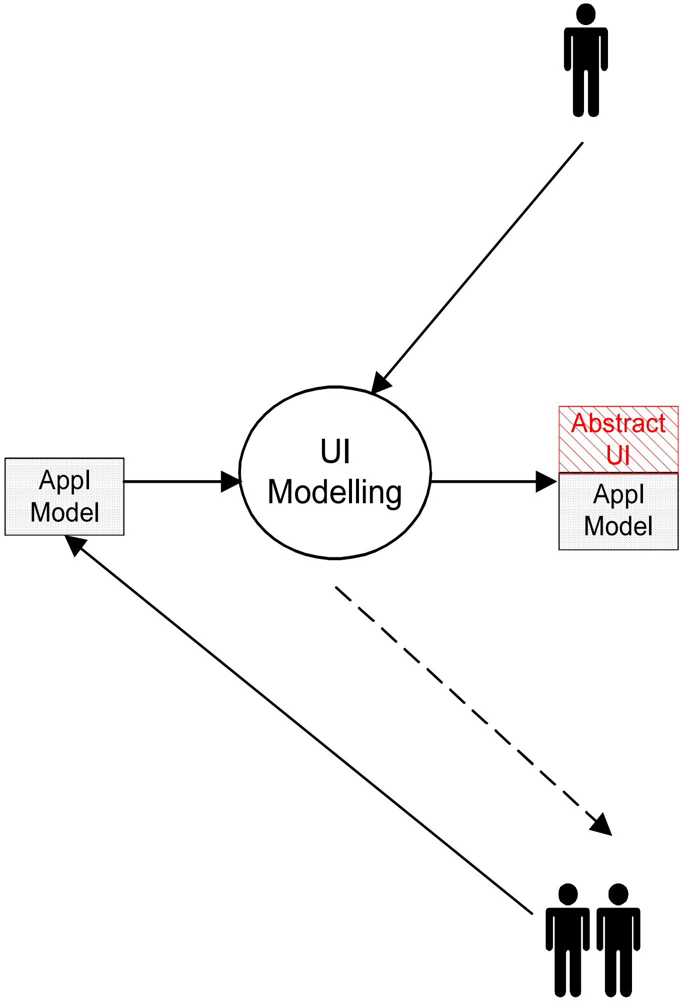
Current Task

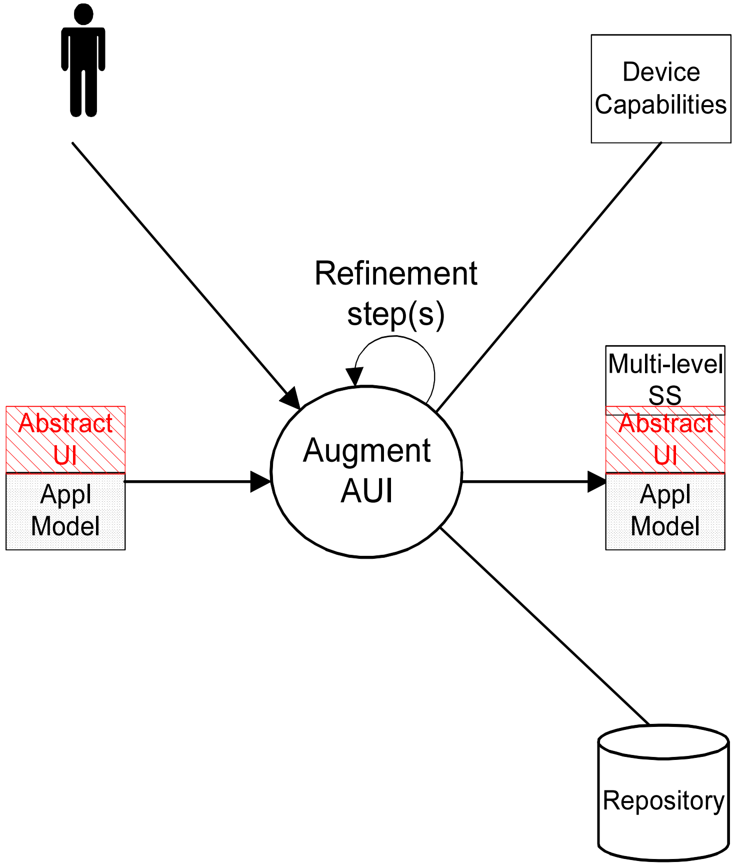
Identifier: Show Music Info

Category: Abstraction

Type:

Frequency:

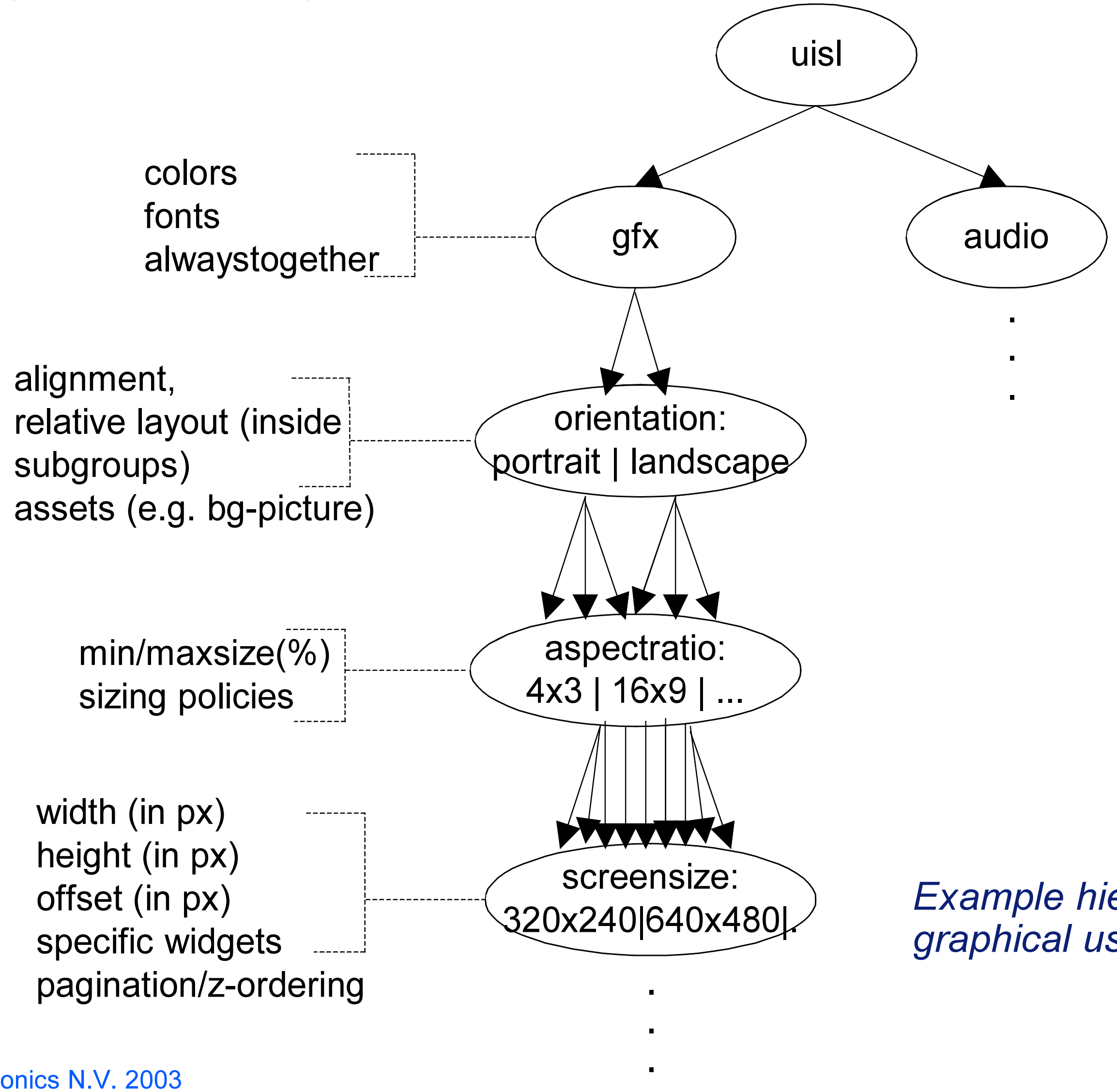

Appl
Model
UI
Modelling
Abstract
UI
Appl
Model


Device
Capabilities
Refinement
step(s)
Multi-level
SS
Abstract
UI
Appl
Model
Augment
AUI
Abstract
UI
Appl
Model
Repository

Augment abstract UI:

- Abstract UI not sufficient: lacks information to create a good user interface (only functional presentation)
- Need for navigation and styling attributes that match characteristics of target device
- To not have to do that (in full detail) for all possible target devices now and in future, we propose to do this on different abstraction levels

- Multi-level stylesheets: style sheets in form of device capability “hierarchy”



uisl

gfx

audio

colors
fonts
alwaystogether

orientation:
portrait | landscape

alignment,
relative layout (inside subgroups)
assets (e.g. bg-picture)

aspectratio:
4x3 | 16x9 | ...

min/maxsize(%)
sizing policies

screensize:
320x240|640x480|.

width (in px)
height (in px)
offset (in px)
specific widgets
pagination/z-ordering

Example hierarchy for graphical user interfaces

Multi-level stylesheets (continued):

- Clustering of device capabilities into abstract classes, by finding common style attributes
- Criteria for clustering:
  - Attribute(s) common for all devices within cluster
  - Common for majority (override in lower levels for exceptions)
  - Lowest-common denominator (refine in lower levels)
  - Threshold boundaries (for flexible style attributes)
  - Adaptation can be automated, e.g.
    - format conversion (e.g. JPEG → GIF),
    - colour coding (e.g. colour → grayscale)
    - resizing bitmaps, fonts, etc.

- Attempt to reduce authoring complexity

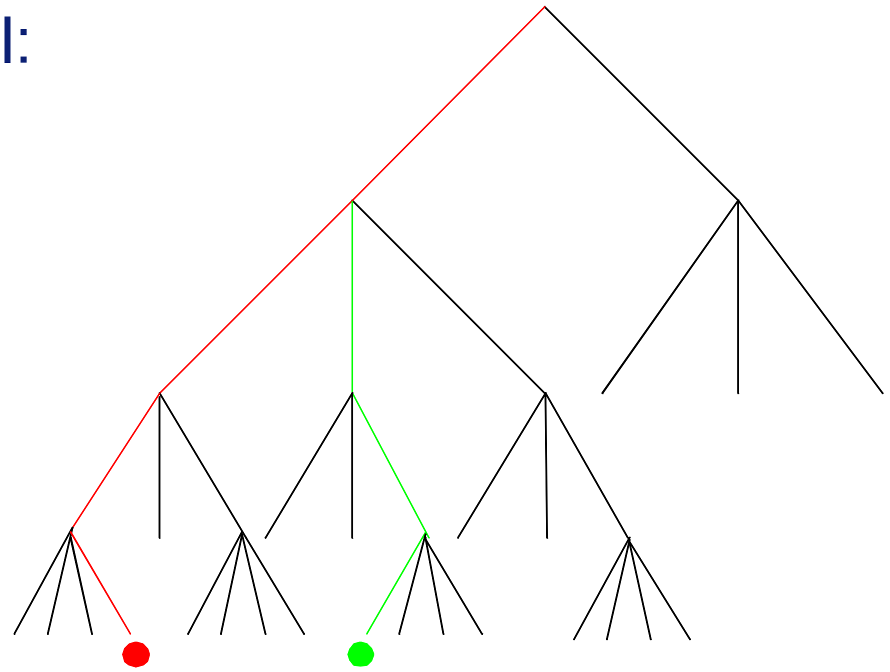
Abstraction level:

gfx

orientation

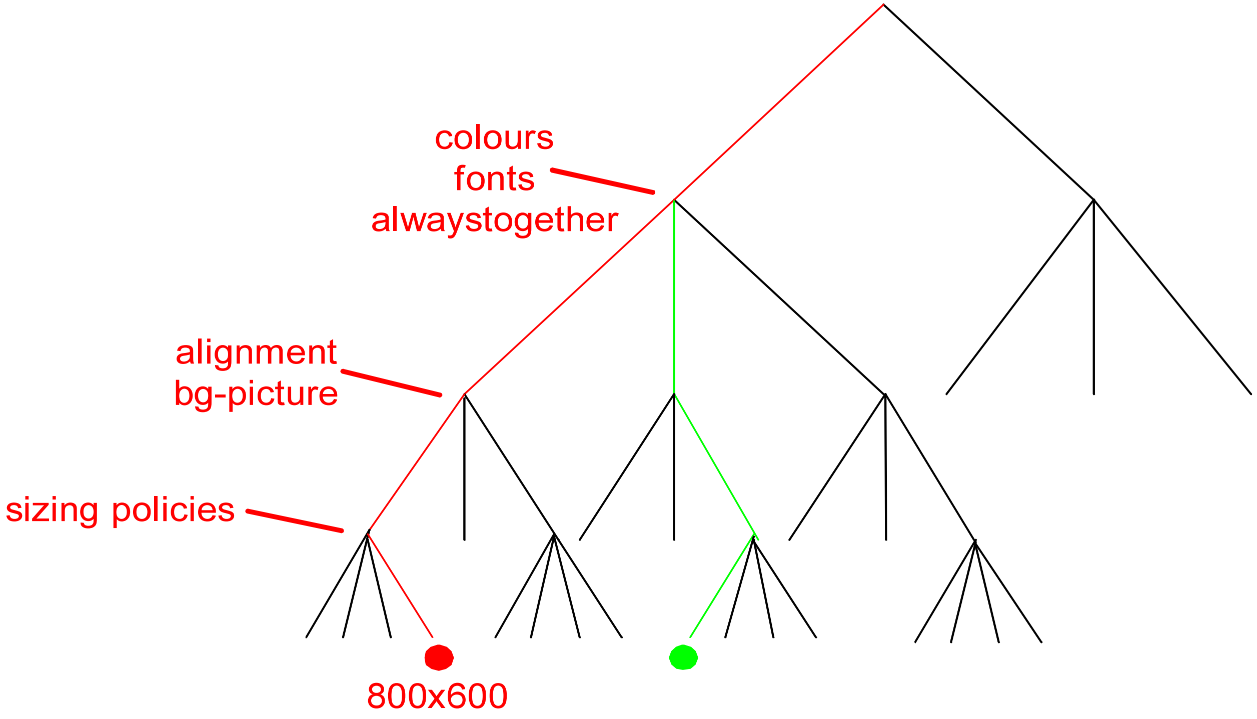
aspectratio

resolution



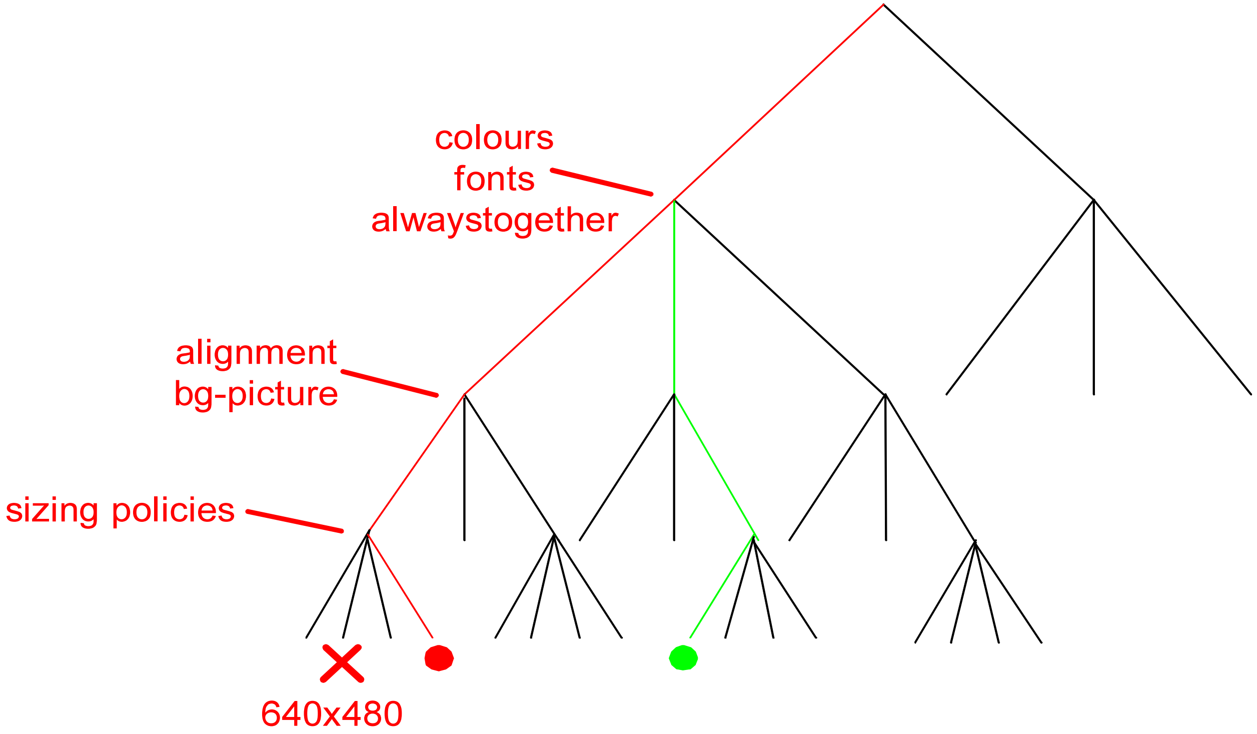
Author can focus on a few specific target devices and still allow re-use of information for other target devices

- Collecting style hints



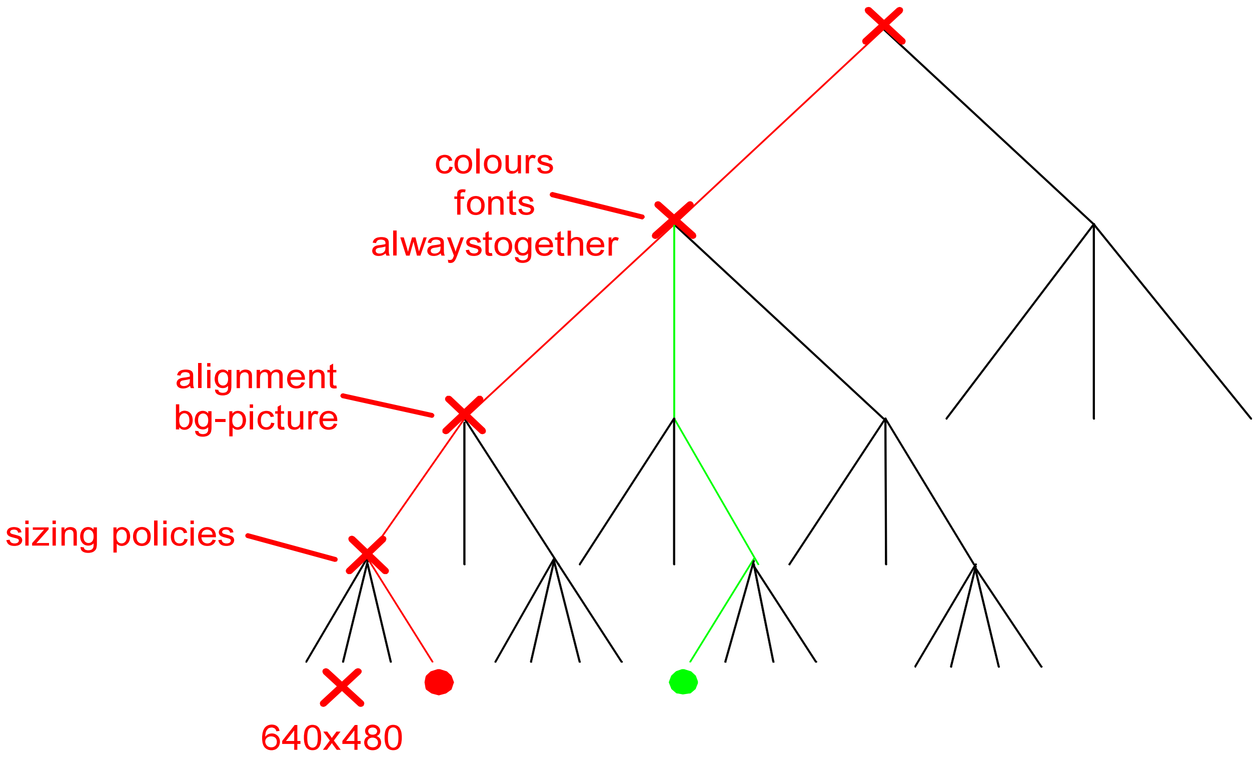
By matching multi-level stylesheets against the device’s capabilities, the relevant style hints can be collected.

- Collecting style hints



By matching multi-level stylesheets against the device’s capabilities, the relevant style hints can be collected.

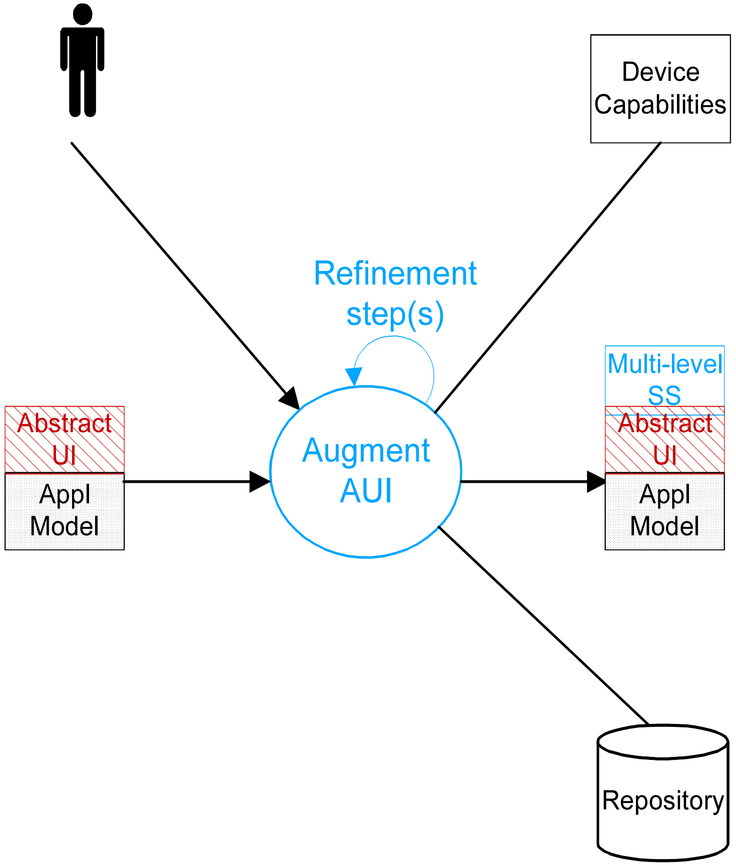
- Collecting style hints

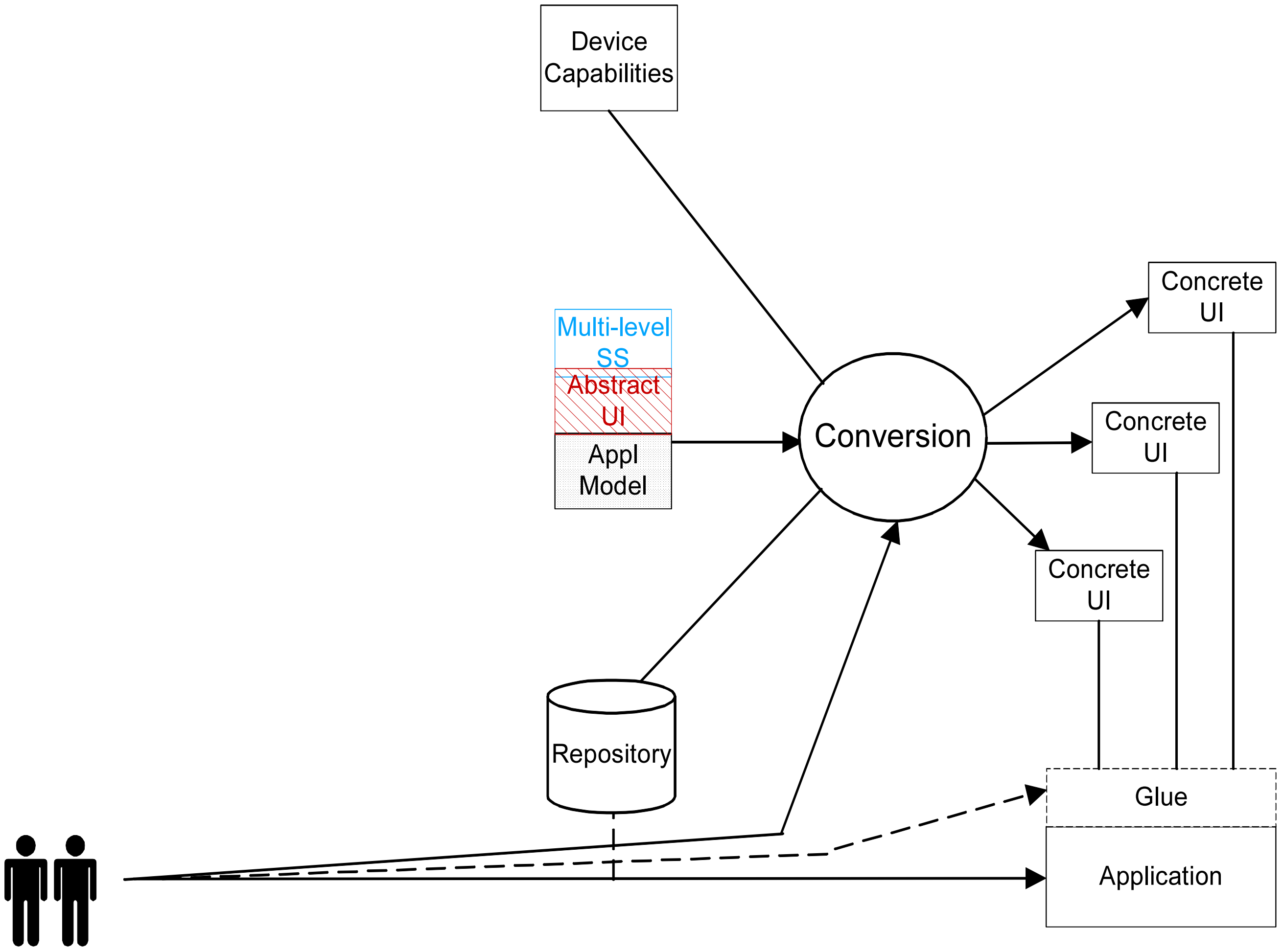


By matching multi-level stylesheets against the device's capabilities, the relevant style hints can be collected.

Multi-level stylesheets (continued):

- Some attributes may be missing for unspecified devices, but higher level attributes can be used for adaptation
- Still provide some level of support for all kinds of unknown/future devices
- Less work if later want to add support for another device


Device
Capabilities
Refinement
step(s)
Abstract
UI
Appl
Model
Augment
AUI
Multi-level
SS
Abstract
UI
Appl
Model
Repository

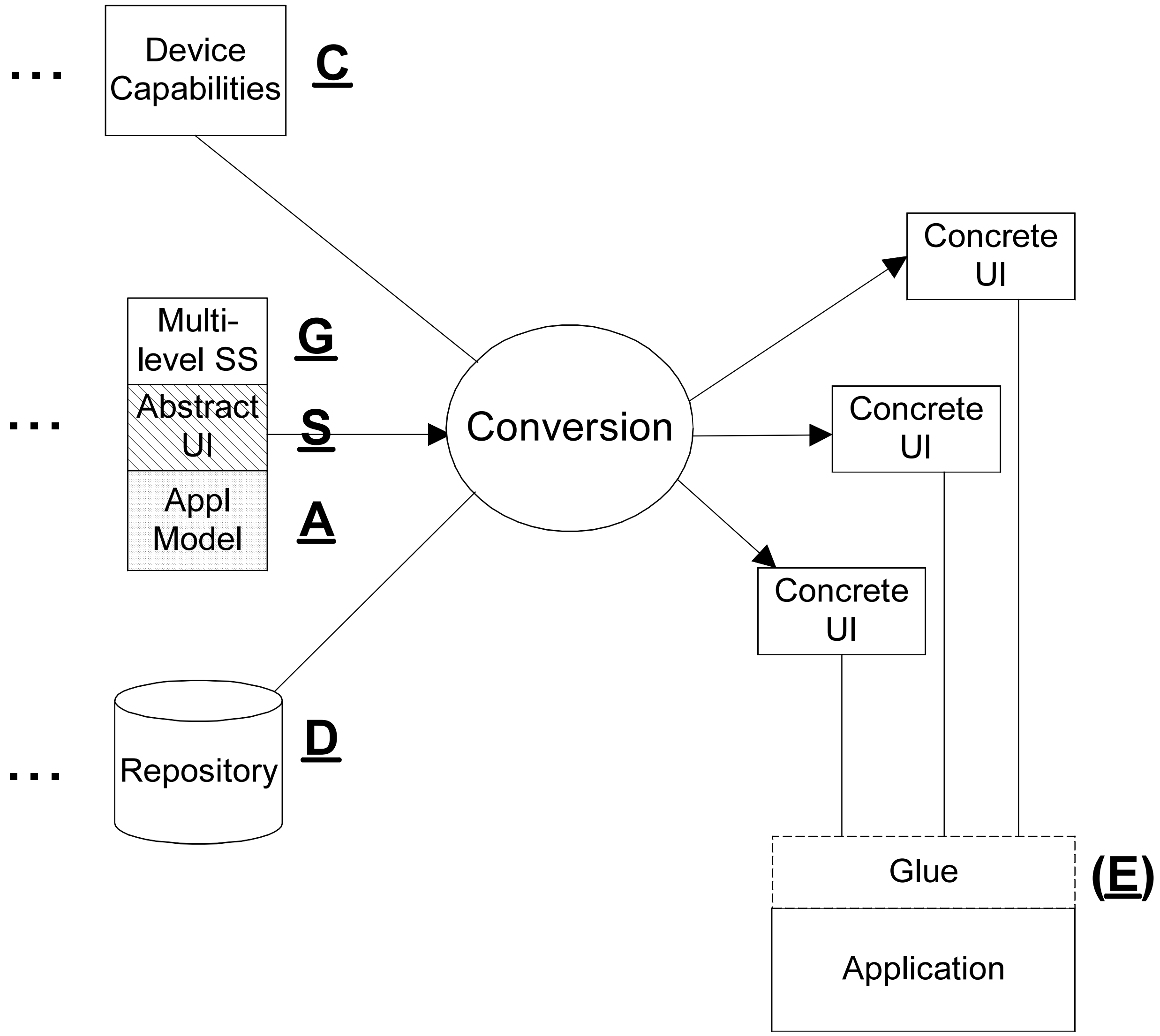

Device Capabilities
Multi-level SS
Abstract UI
Appl Model
Conversion
Concrete UI
Concrete UI
Concrete UI
Repository
Glue
Application

Conversion:

Concrete UI = Adapt(A, S, C, G, D), with:

A = **A**pplication model
S = UI **S**emantics
C = Device **C**apabilities
D = Instance **D**ata
G = UI design **G**uidelines

Also possible at run-time, if you have all this information (+E)


...
Device Capabilities
C
...
Multi-level SS
G
Abstract UI
S
Appl Model
A
...
Repository
D
Conversion
Concrete UI
Concrete UI
Concrete UI
Glue
(E)
Application

Conversion (continued):

- Mapping abstract UI elements (tasks) to concrete interaction elements, depending on device capabilities
- Collecting the style hints by matching against capabilities
- Feed instance-data into UI
- Possible link to application
- Can generate code or e.g. XML markup (or run-time)

Contents

- Scope
- Our approach:
  - Abstraction
  - Multi-level stylesheets
- Challenges
- Conclusions

Challenges:

- Introducing to designers: abstract thinking/complexity, vocabulary/tools, Does it really help them? Affordable?
- Easy to fall into trap of thinking of specific UI-instances: e.g. implicit navigation/layout restrictions during task-modelling
- Still requires lot of effort to make DI authoring mature, e.g. finding differences/commonalities between UIs and UI devices, technology (XForms?, …), etc.

Contents

- Scope
- Our approach:
  - Abstraction
  - Multi-level stylesheets
- Challenges
- Conclusions

Conclusions:

- Device independent UI authoring important in area of in-home networking (one application, many different UIs)
- Abstraction not enough
- Multi-level stylesheets provides a way to augment abstract UI with style and navigation information
- Attempt to reduce complexity

PHILIPS