Mobile and Context-aware Interactive Systems
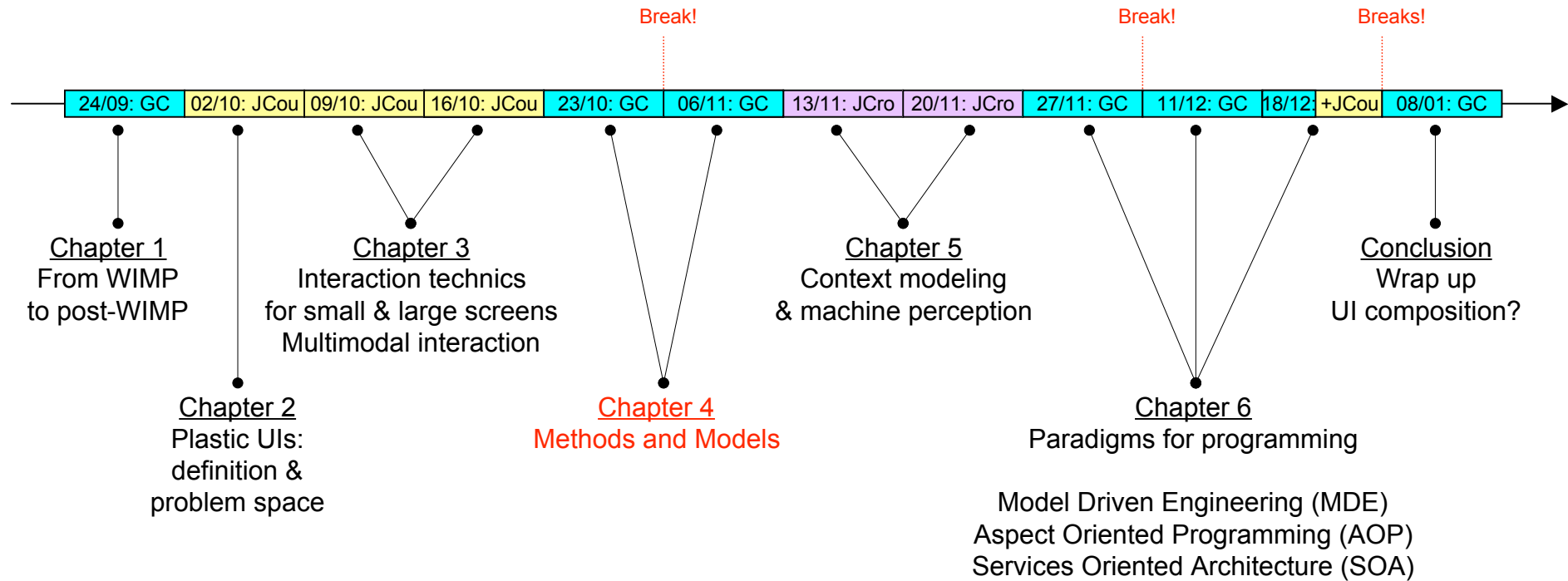
Gaëlle Calvary, Joëlle Coutaz and James Crowley

Université Joseph Fourier (Grenoble I)
ENSIMAG

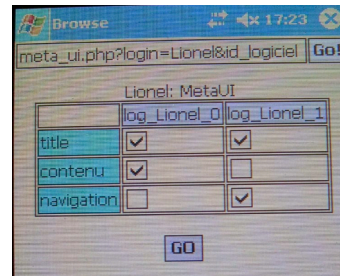Laboratoire d'Informatique de Grenoble (LIG)

Outline and schedule

Break!   Break!   Breaks!

| 24/09: GC | 02/10: JCou | 09/10: JCou | 16/10: JCou | 23/10: GC | 06/11: GC | 13/11: JCro | 20/11: JCro | 27/11: GC | 11/12: GC | 18/12: | +JCou | 08/01: GC |

Chapter 1
From WIMP
to post-WIMP

Chapter 3
Interaction technics
for small & large screens
Multimodal interaction

Chapter 5
Context modeling
& machine perception

Conclusion
Wrap up
UI composition?

Chapter 2
Plastic UIs:
definition &
problem space

Chapter 4
Methods and Models

Chapter 6
Paradigms for programming

Model Driven Engineering (MDE)
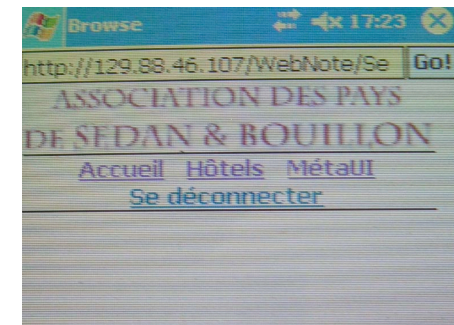Aspect Oriented Programming (AOP)
Services Oriented Architecture (SOA)

GC: Gaëlle Calvary
JCou: Joëlle Coutaz
JCro: James Crowley

How ?

Event: *Connection of the PDA*

Reaction: *Proposition of a UI's redistribution*

- **0- Understand the user's task!**

- **1- Identify the key contexts of use to be covered**
  - It can be done by
    - The designer based on a requirements analysis
    - The system using a *probe* [Ganneau 08]
      - Examples of variables that can be observed *in situ* on a mobile
        - Location (GPS)
        - Time (day and time)
        - Physical state of the mobile (battery)
        - User's activity (applications used and under use)

      - One approach: User modeling by Bayesian networks

      - Examples of exploitation
        - Observability of the context of use
        - User actions anticipation [Cypher 91]

- Bayesian network (Netica on mobile)
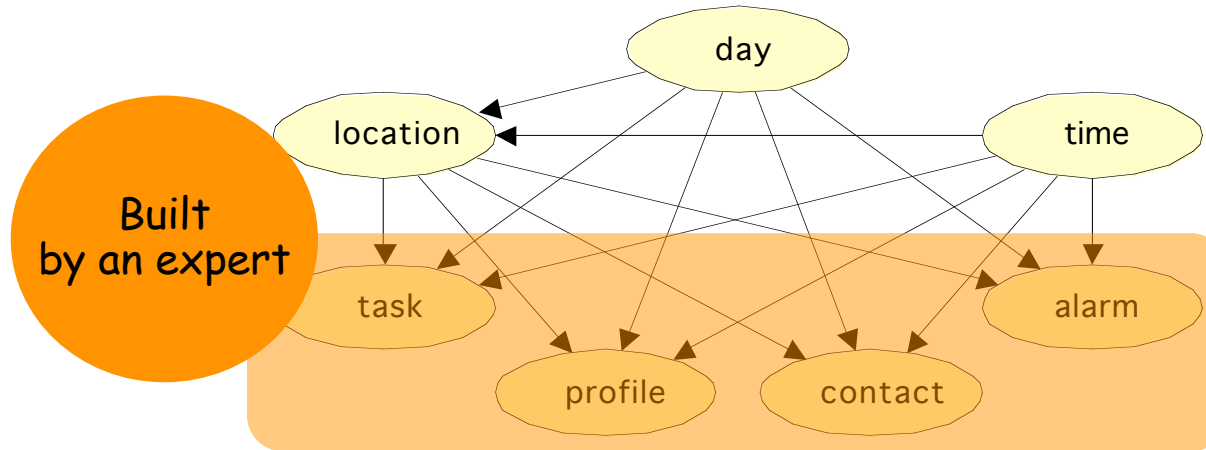
- Raw data

Location	Day	Time	Task
CID 33820 LAC 10241	Monday	Morning	Call
CID 33820 LAC 10241	Monday	Morning	Games
...	...	...	...
CID 55352 LAC 10241	Wednesday	Evening	SMS
CID 55352 LAC 10241	Wednesday	Evening	Contacts

- Data analysis

Methods

- Impact on the UI's design

look & feel   context α   **profile**

context β   **homescreen**

reorganisation   **background**

Trust?

Ethics?

7

- **2- Design the UIs for the key contexts of use**
  - It can be done by
    - The designer: *pre-computed* UIs
    - The system: *generated* UIs
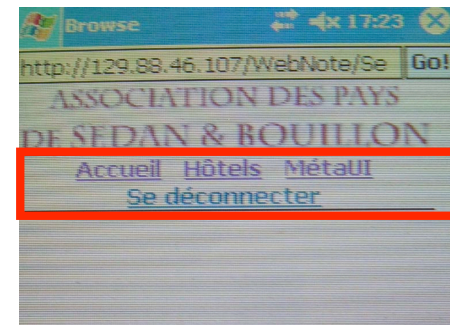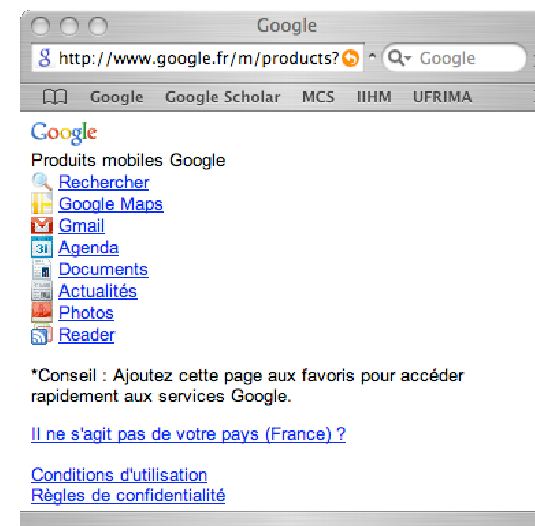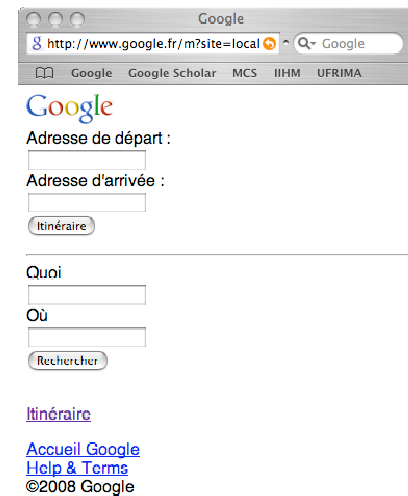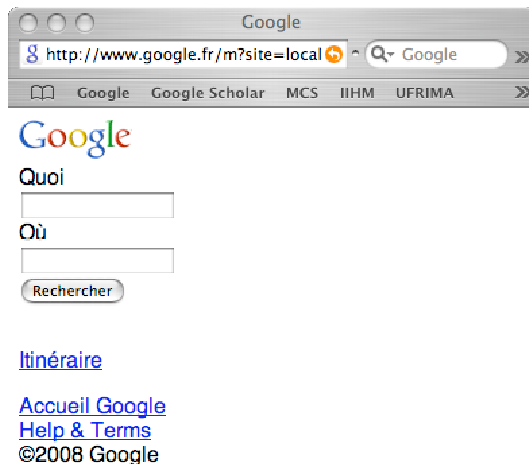   - The end-user: *customized* UIs
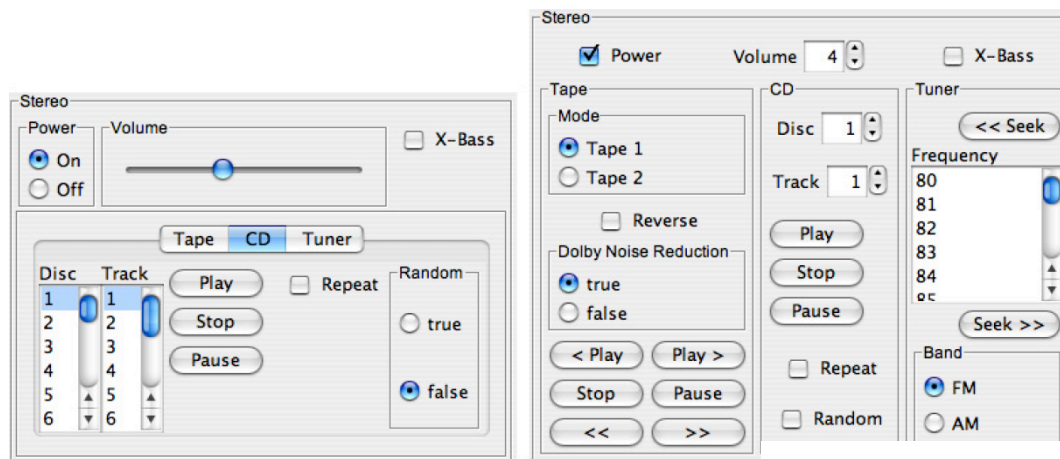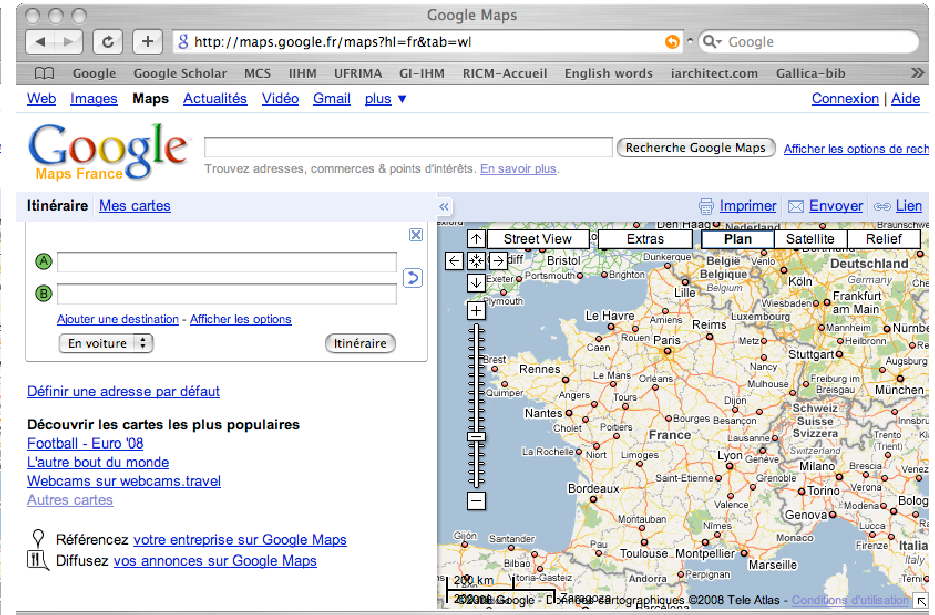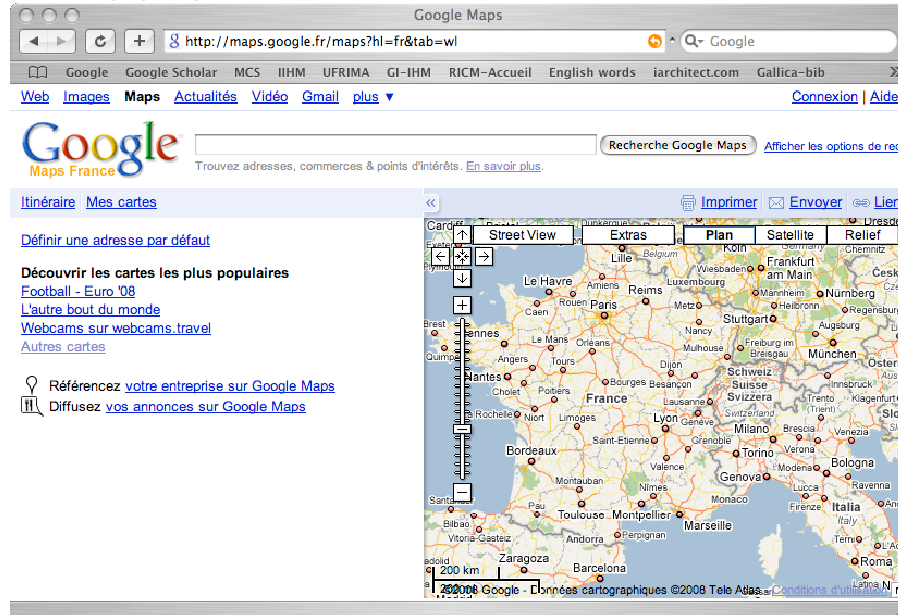

[Gajos 2004]

WinApp

8

- 2- Design the UIs for the key contexts of use
  - Open issues
    - Consistency
      - Consistency / inconsistency? [Workshop @CHI'06]

9

- Exercise 1: Point out the discrepancies between Google and Google mobile

Methods

- Exercise 1: Point out the discrepancies between Google and Google mobile

- **2- Design the UIs for the key contexts of use**
  - Open issues
    - Consistency
      - Consistency / inconsistency? [Workshop @CHI'06]
      - Reference platform?
        - Degradation [Florins 2006]
        - Beautification

    - Inter-usability [Karsenty 05]
      - Additional UIs (observability and/or control)
        - Context of use
        - Evolution
        - Transition

[Ganneau 08]

13
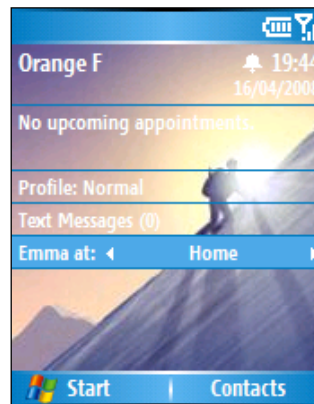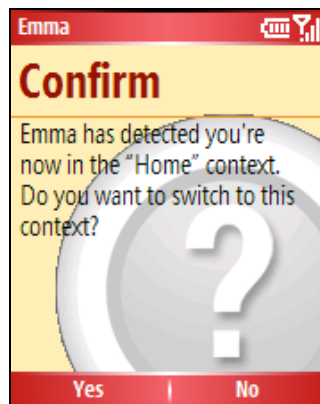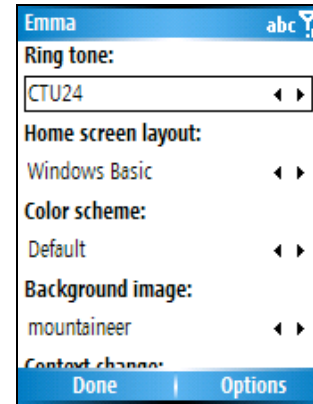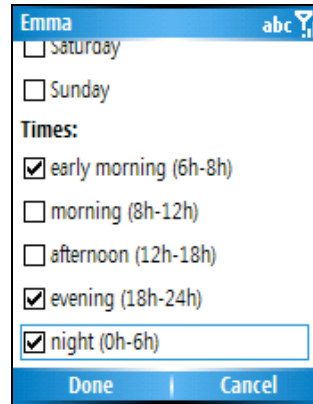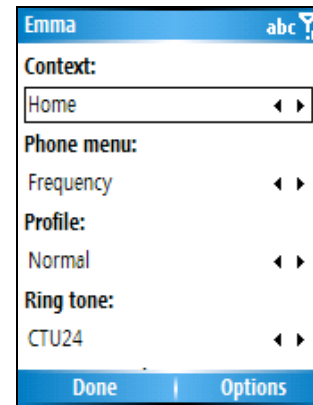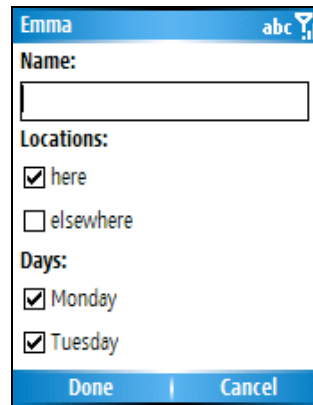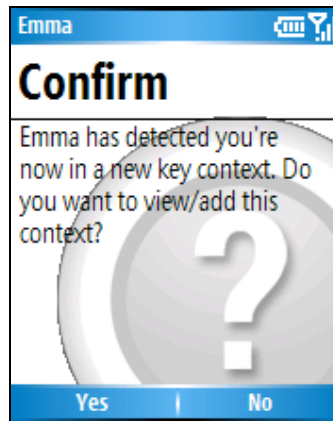
■ 2- Design the UIs for the key contexts of use

- Context of use -

| User |

| Platform |

| Environment |

- Adaptation -

| Evolution |

| Transition |

- Quality in use -

| Criteria |

| Metrics |

Also in the UI …

| Task-Concepts | | Task-Concepts |

| Abstract UI | | Abstract UI |

| Concrete UI | | Concrete UI |

| Final UI | | Final UI |
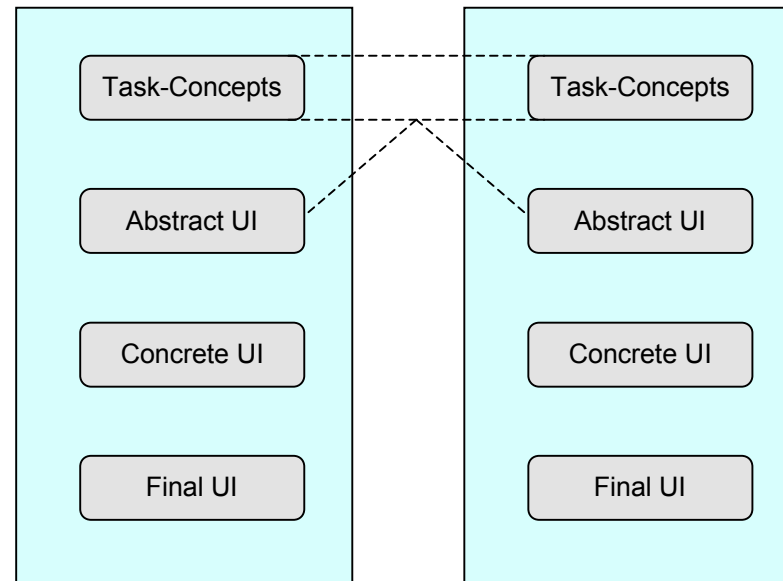
- **2- Design the UIs for the key contexts of use**
  - Exercise 2: Classify the Google's discrepancies
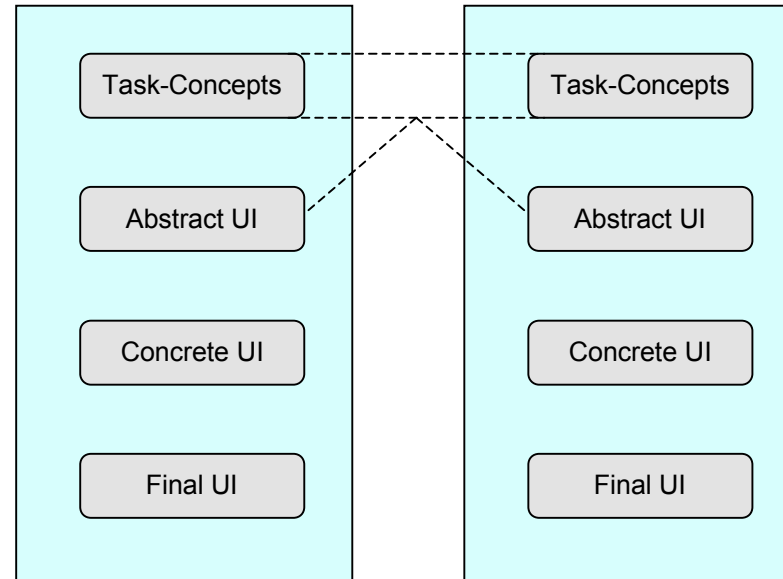
- Context of use -

User

Platform

Environment

- Adaptation -

Evolution

Transition

- Quality in use -

Criteria

Metrics

Also in the UI …

Task-Concepts		Task-Concepts
Abstract UI		Abstract UI
Concrete UI		Concrete UI
Final UI		Final UI

- **2- Design the UIs for the key contexts of use**
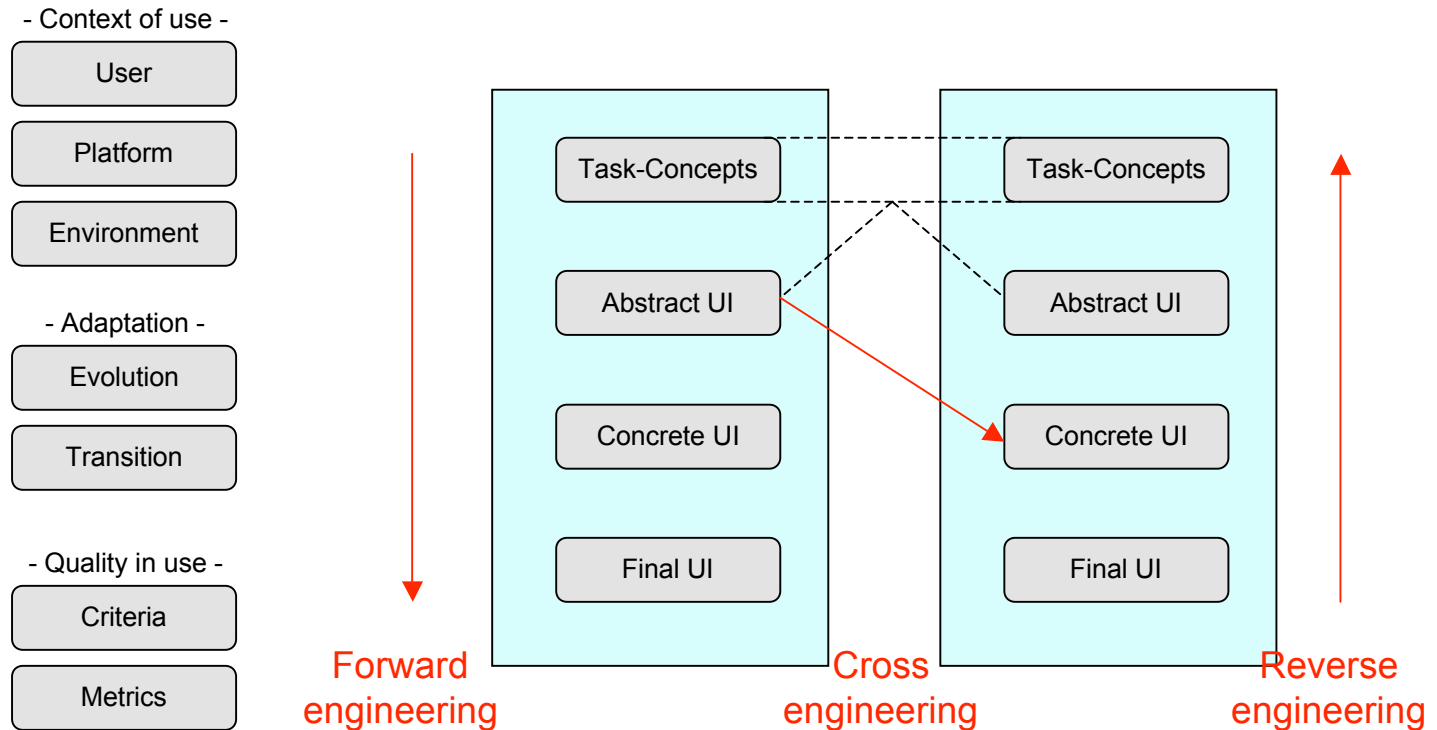 - Many entry points

- **2- Design the UIs for the key contexts of use**
  - Many entry points

Methods

- **2- Design the UIs for the key contexts of use**
  - Many entry points

- **2- Design the UIs for the key contexts of use**
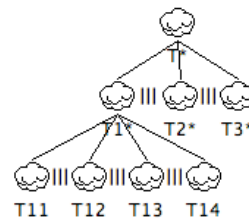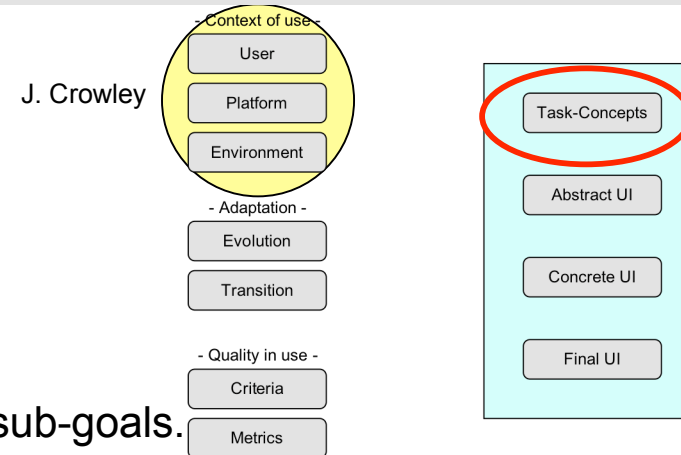 - Many entry points
  - Many design paths

- Context of use -

| User |
| Platform |
| Environment |

- Adaptation -

| Evolution |
| Transition |

- Quality in use -

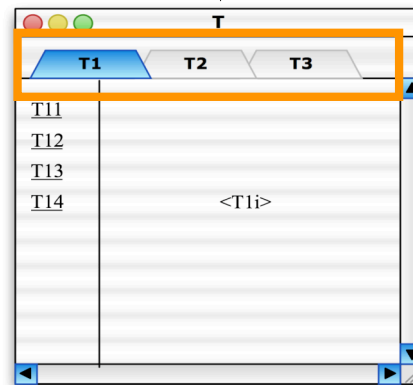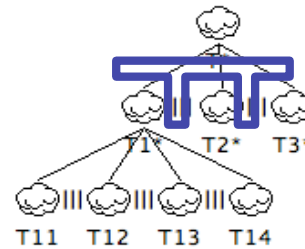| Criteria |
| Metrics |

Task-Concepts		Task-Concepts
Abstract UI		Abstract UI
Concrete UI		Concrete UI
Final UI		Final UI

Forward engineering   Cross engineering   Reverse engineering

19

- **2- Design the UIs for the key contexts of use**
    - Many entry points
    - Many design paths
    - Many tools
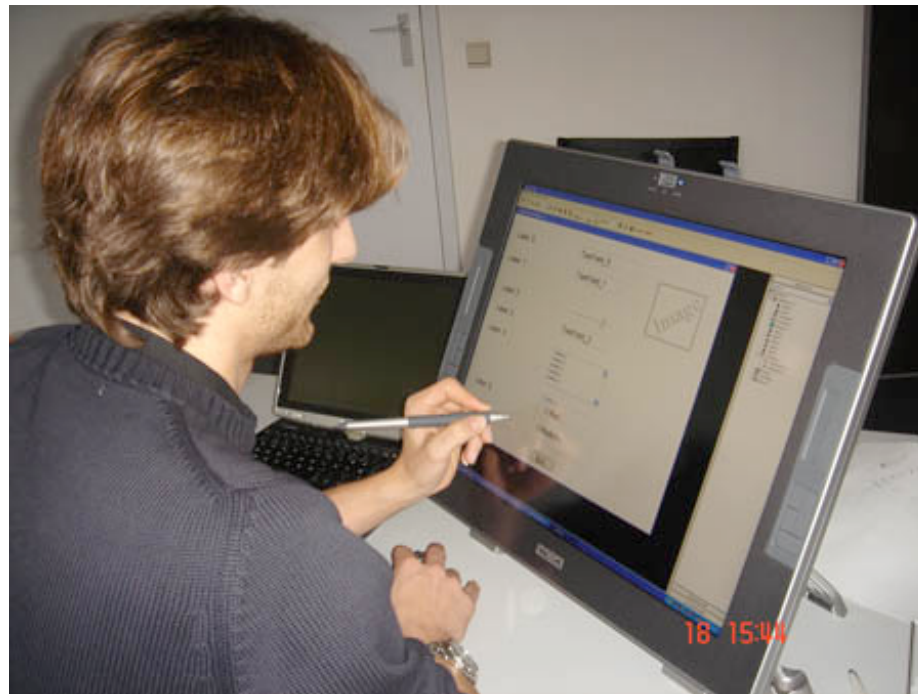
- **3- Evaluate**
  - In labs
  - In situ
    - Challenging while in mobility … probes

Models

- **Context of use**
  - See J. Crowley

- **Design of the UI**
  - Task & concepts model
    - Task: A decomposition of the user's goal into sub-goals. CTT [Paterno, 97] is the reference in task modeling.
    - (Sub-)tasks may be decorated (e.g., domain concepts, frequency, criticity … as well as the « Context of use » in which the task makes sense)
    - Concepts: A description of the data that are manipulated by the user. They can be described as a UML class diagram.

J. Crowley

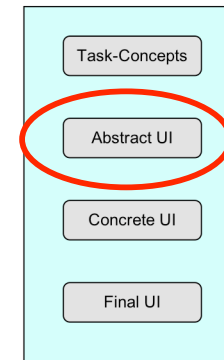- Context of use -
  - User
  - Platform
  - Environment

- Adaptation -
  - Evolution
  - Transition

- Quality in use -
  - Criteria
  - Metrics

Task-Concepts
Abstract UI
Concrete UI
Final UI

A CTT task model [Paterno, 97]

Task Properties

General   Objects   Time Performance

Task Properties
Identifier : Get weather info
Name :
Category : Abstraction
Type :
Frequency :
Platforms : ☐ PDA ☐ Desktop ☐ Cellphone
Description :

☐ Iterative ☐ Optional ☐ Part of Cooperative Task
Precondition :

Update   Cancel   Clear   Close

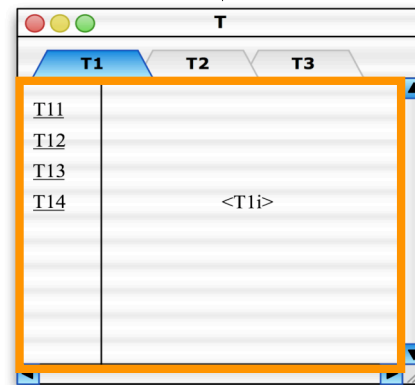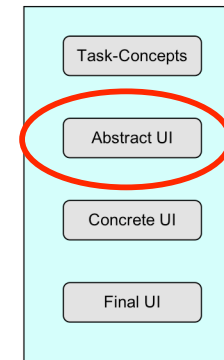22   CTTE [Mori, 02] : http://giove.isti.cnr.it/ctte.html
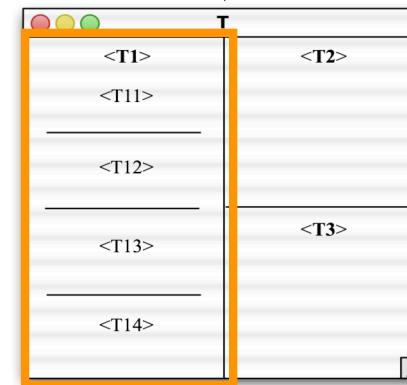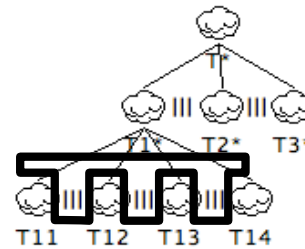
Abstract UI

- An organization of the UI into workspaces and navigation (i.e., *gives access to*)
- It is the result of a grouping process (e.g., group together the frequent tasks)

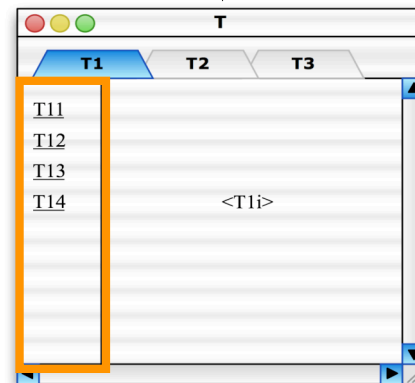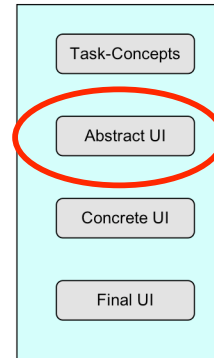- Context of use -

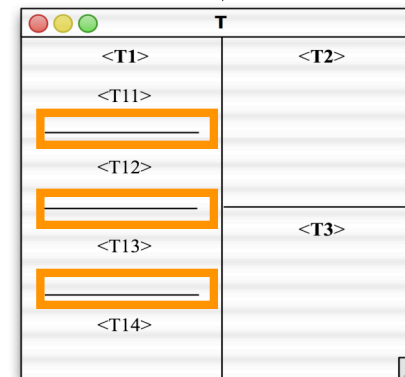| User |
| Platform |
| Environment |

- Adaptation -

| Evolution |
| Transition |

- Quality in use -

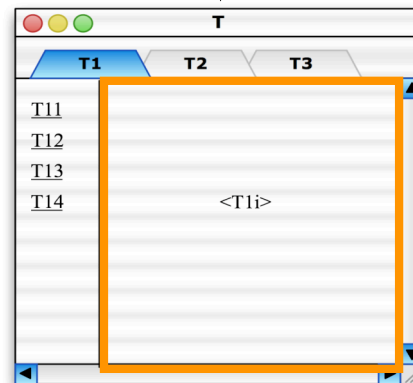| Criteria |
| Metrics |

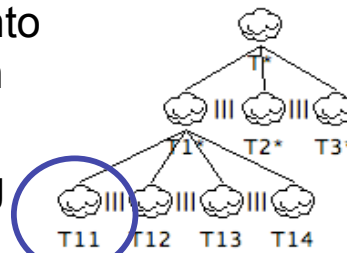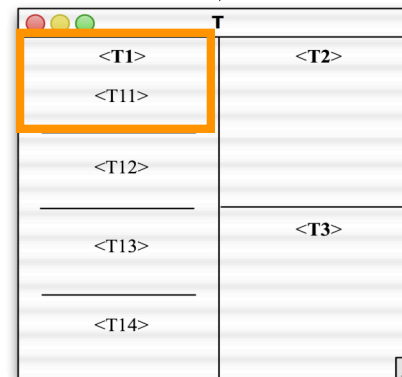| Task-Concepts |
| Abstract UI |
| Concrete UI |
| Final UI |

Browsability
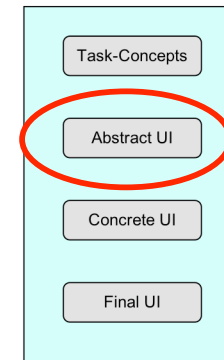
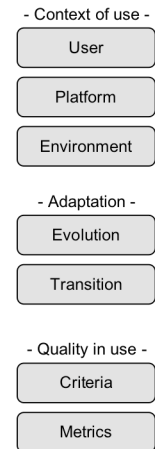Observability

- **Abstract UI**

  - An organization of the UI into workspaces and navigation (i.e., *gives access to*)

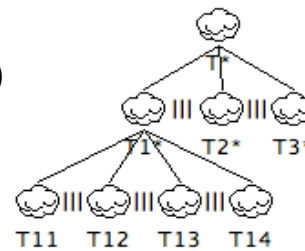 - It is the result of a grouping process (e.g., group together the frequent tasks)

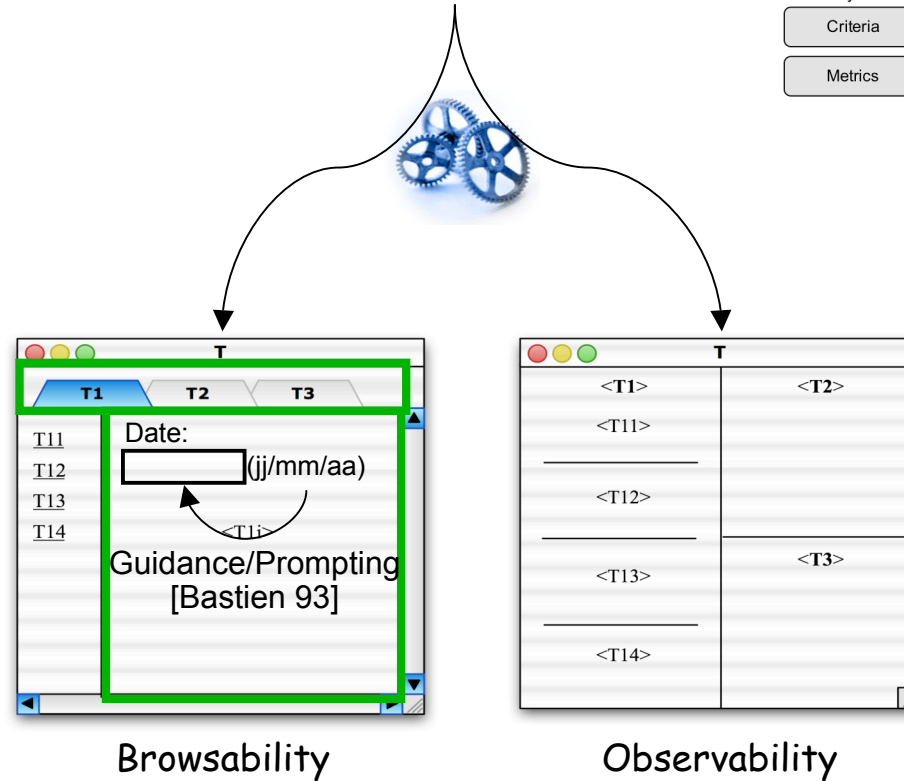- Context of use -

User
Platform
Environment

- Adaptation -

Evolution
Transition

- Quality in use -

Criteria
Metrics

Task-Concepts
Abstract UI
Concrete UI
Final UI

Browsability

Observability

24

Models

- **Abstract UI**

  - An organization of the UI into workspaces and navigation (i.e., *gives access to*)

  - It is the result of a grouping process (e.g., group together the frequent tasks)

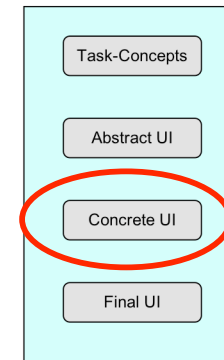- Context of use -

| User |
| Platform |
| Environment |

- Adaptation -

| Evolution |
| Transition |

- Quality in use -

| Criteria |
| Metrics |

| Task-Concepts |
| Abstract UI |
| Concrete UI |
| Final UI |

Browsability

Observability

- **Abstract UI**
  - An organization of the UI into workspaces and navigation (i.e., *gives access to*)
  - It is the result of a grouping process (e.g., group together the frequent tasks)

- Context of use -

| User |
| Platform |
| Environment |

- Adaptation -

| Evolution |
| Transition |

- Quality in use -

| Criteria |
| Metrics |

| Task-Concepts |
| **Abstract UI** |
| Concrete UI |
| Final UI |

Browsability

Observability

26

Models

- **Abstract UI**
  - An organization of the UI into workspaces and navigation (i.e., *gives access to*)
  - It is the result of a grouping process (e.g., group together the frequent tasks)

- Context of use -
| User |
| Platform |
| Environment |

- Adaptation -
| Evolution |
| Transition |

- Quality in use -
| Criteria |
| Metrics |

| Task-Concepts |
| Abstract UI |
| Concrete UI |
| Final UI |

Browsability

Observability

- **Concrete UI**
  - A composition of interactors (e.g., input field, radio button)
  - Is obviously dependent on the platform

- Context of use -

| User |
| Platform |
| Environment |

- Adaptation -

| Evolution |
| Transition |

- Quality in use -

| Criteria |
| Metrics |

| Task-Concepts |
| Abstract UI |
| Concrete UI |
| Final UI |

Browsability

Date:

(jj/mm/aa)

<T1i>

Guidance/Prompting
[Bastien 93]

Observability

<T1> <T2>
<T11>
<T12>
<T13> <T3>
<T14>

Final UI

- A choice of a programming language and an execution environment
- Interactors/components can be close/open adaptive
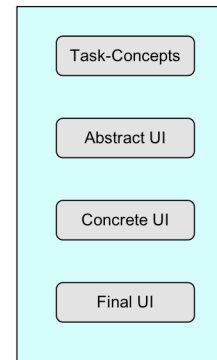- UI's may be pre-computed or generated on the fly

- Context of use -
- User
- Platform
- Environment

- Adaptation -
- Evolution
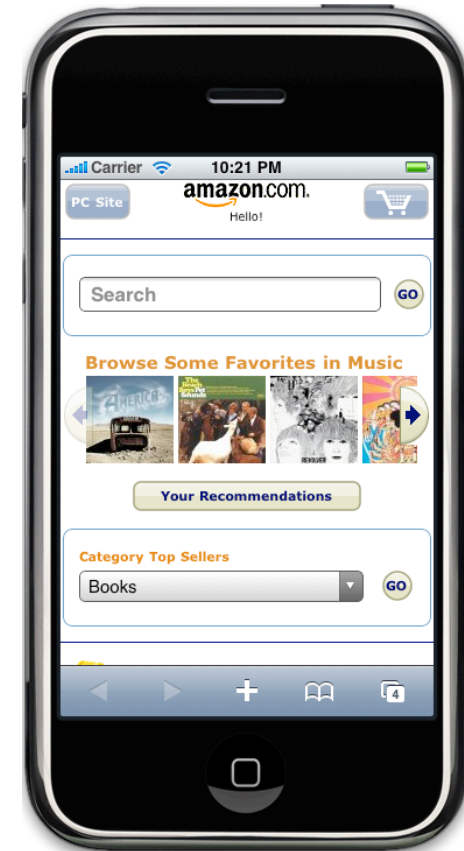- Transition

- Quality in use -
- Criteria
- Metrics

- Task-Concepts
- Abstract UI
- Concrete UI
- Final UI

29
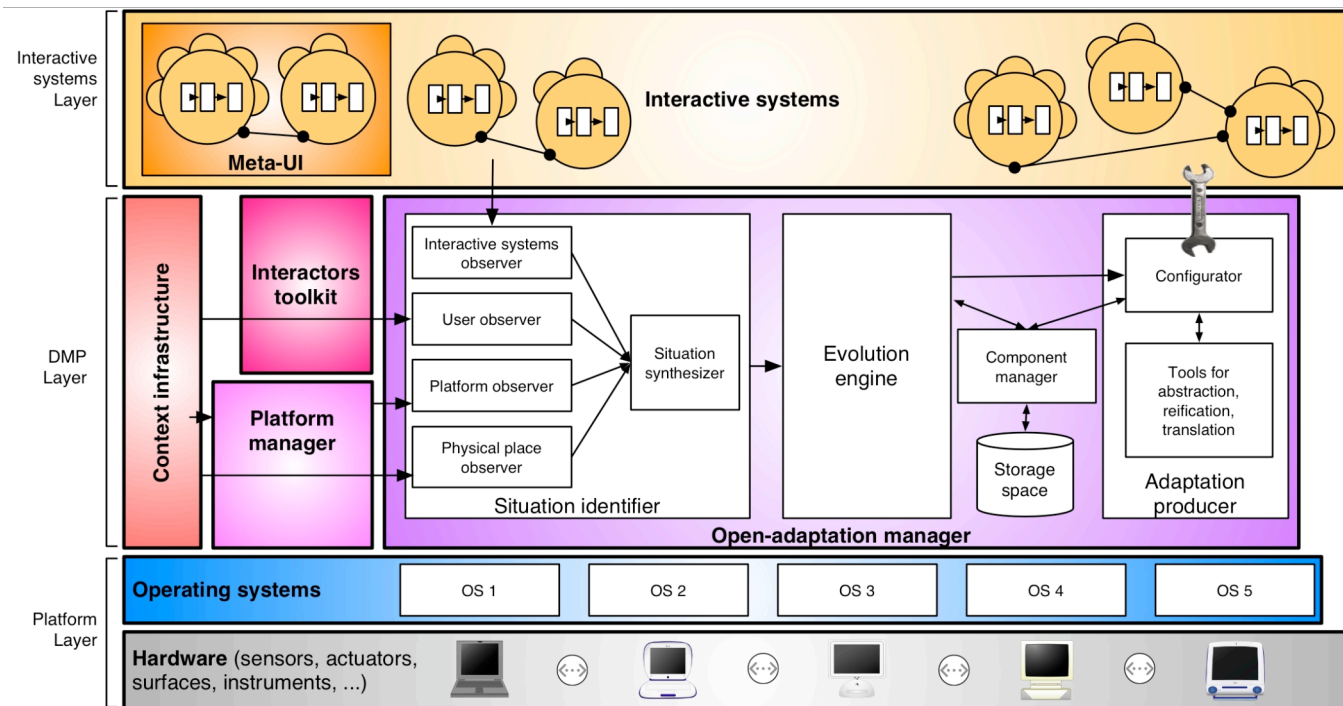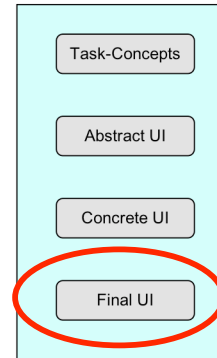
- **Adaptation models**
  - **Evolution**
    - ECA+V Rules:
      On *event* if *condition* then *action* ensures *value*

    - Evolution rules may be placed under the control
      of the end-user via a dedicated UI.
      This UI can be weaved into the *business* UI

  - **Transition**
    - A user's guidance
    - It may be useful while remolding as well as redistributing the UI

- Context of use -
- User
- Platform
- Environment

- Adaptation -
- Evolution
- Transition

- Quality in use -
- Criteria
- Metrics

- Task-Concepts
- Abstract UI
- Concrete UI
- Final UI
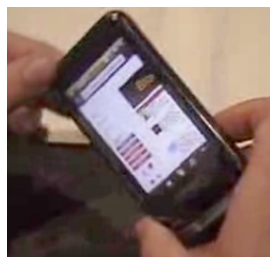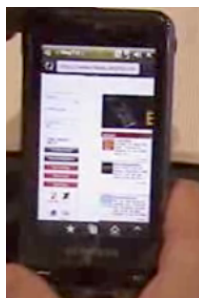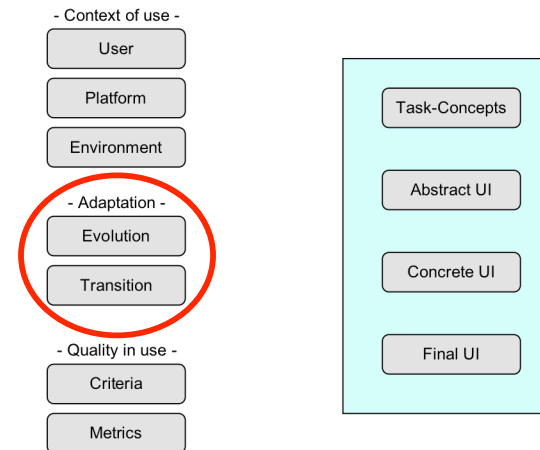
Samsung

See Rekimoto

30
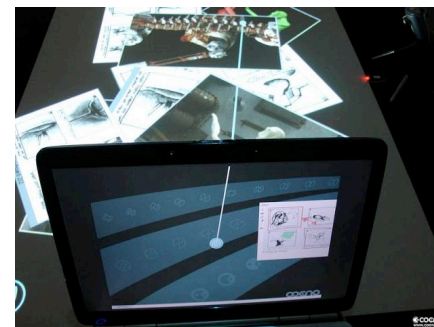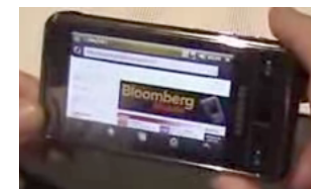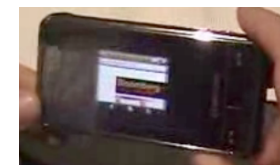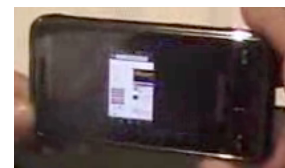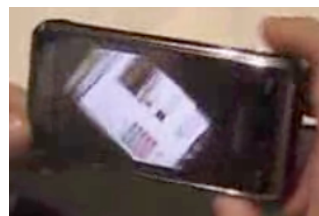
- Exercice 3: FlexClock [Grolaux 01]
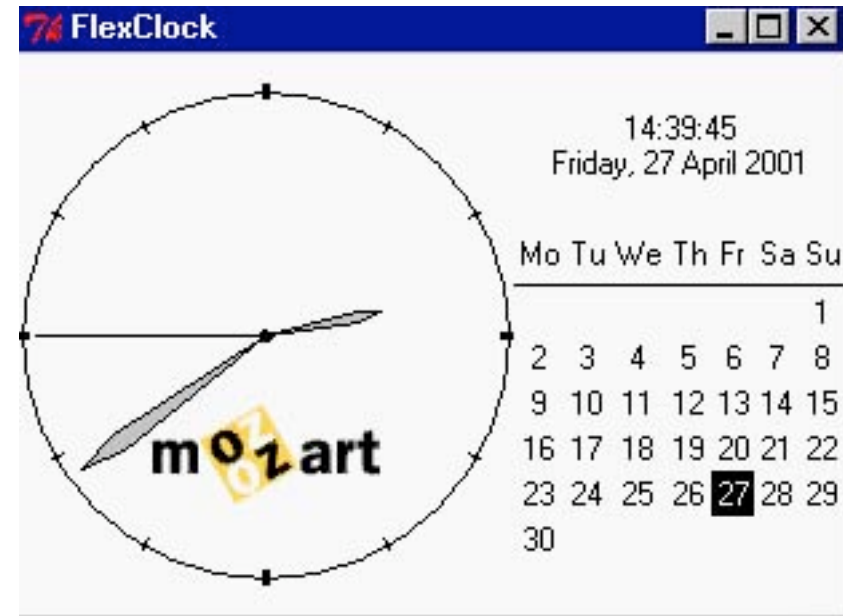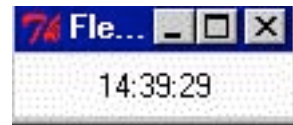
- Models
  - Context of use
  - Adaptation (driving value?)
  - UI
  - Beautification & degradation rules?

31

- **Towards a mapping model …**

■ **Exercice 4: Discrepencies between desktop and mobile versions**

■ Exercice 5: Discrepencies between desktop and mobile versions

34

- Exercice 6: Small is beautiful?
  - Elicit examples of degradation rules at each level of abstraction