

## UbiComp Software Architecture and Technology Part 1

Phil Gray

1

## Overview

- Features of UbiComp Software
- UbiComp Architectural Models
- Case Study: The Context Toolkit
- Other UbiComp Tools
- Plasticity

HCI4 2006

UbiComp Software Architecture & Technology 1

2

## Features of UbiComp software

- Software framework and components that provide services related to ubiComp applications
  - » Access to and processing of context information
  - » Location detection & tracking
  - » Access to and control of shared interaction services
  - » Repositories of shared information
  - » Interaction techniques for augmentation

HCI4 2006

UbiComp Software Architecture & Technology 1

3

## Architectural Models

- Application models – the app programmer's point of view
  - » Monolithic – Seeheim, Arch
  - » Component-based – MVC, PAC
- Service models – the context generator's point of view
  - » Components
    - From interactors to "contextors"
  - » Composing components
    - GrumpsNet
    - The Context Toolkit
- Middleware infrastructure – the distributed systems view
  - » Publish subscribe messaging - Elvin
  - » Shared dataspace (tuple space) - Equip

HCI4 2006

UbiComp Software Architecture & Technology 1

4

## Seeheim Model

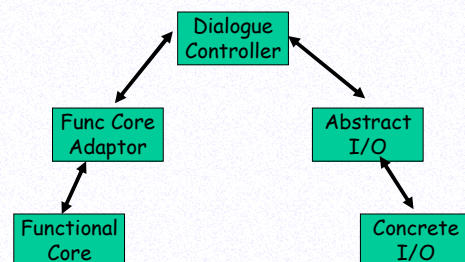


HCI4 2006

UbiComp Software Architecture & Technology 1

5

## Arch Model

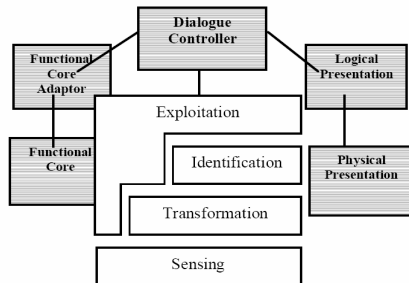


HCI4 2006

UbiComp Software Architecture & Technology 1

6

## The Augmented Arch



HCI4 2006

Ubicomp Software Architecture & Technology 1

7

## Interactors

- Deconstruction of previous monolithic architecture into a set of communicating components
  - » Inheritance
  - » Composition
- MVC
  - » Variant is PAC (Presentation – Abstraction – Control)
- Widgets

HCI4 2006

Ubicomp Software Architecture & Technology 1

8

## Contextors

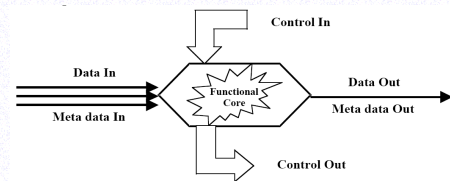
- See Rey, Coutaz & Crowley. The Contextor: A Computational Model for Contextual Information. Proc Workshop on Concepts & Models for Ubiquitous Computing, Ubicomp 2002.
- Contextor is “a software abstraction that models a relation between variables of the observed system context”
- E.g.,
  - » GPS positions
  - » Input from video cameras and microphones
  - » RFID sensor events
  - » Keystroke events

HCI4 2006

Ubicomp Software Architecture & Technology 1

9

## Contextors

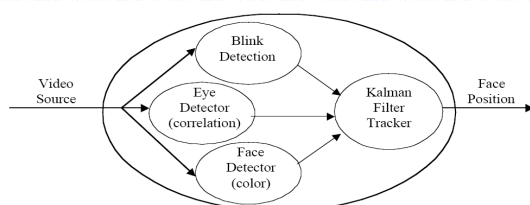


HCI4 2006

Ubicomp Software Architecture & Technology 1

10

## Example of Contextor Network



HCI4 2006

Ubicomp Software Architecture & Technology 1

11

## GrumpsNet

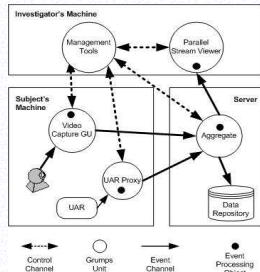
- Distributed System for capturing, distributing, storing and processing usage data (i.e., data generated by user-computer interaction)
- Develop at GUCSD as part of the Grumps Project (Generic Remote Usage Monitoring Production System), 2000-2004

HCI4 2006

Ubicomp Software Architecture & Technology 1

12

## GrumpsNet Architecture



For more details, see:  
Evans et al. The Pervasiveness of Evolution in GRUMPS Software. *SP&E*. Feb 03.

HC14 2006

UbiComp Software Architecture & Technology 1

13

## Middleware Infrastructure

- Most ubicomp systems rely on a distributed messaging system
  - » Publish-subscribe mechanism
  - » E.g., Elvin
- shared data spaces
  - » in AI, "blackboard" systems
  - » distributed systems approach: tuple spaces
  - » Producers put data in tuple space; consumers retrieve data from tuple space
  - » Tuple space may also perform ops on data

HC14 2006

UbiComp Software Architecture & Technology 1

14

## Equip

- EQUIP is "a purpose-built 'Equator Universal Platform' that merges state sharing and notification-based paradigms in order to allow flexible sharing between components"
- Originated as a VR engine
- Built around a shared data service (tuple space)
- Supports
  - » Publish-subscribe event notification system
  - » 3d spatial models
  - » Integration with wireless enabled handheld devices

HC14 2006

UbiComp Software Architecture & Technology 1

15

## Equip Refs

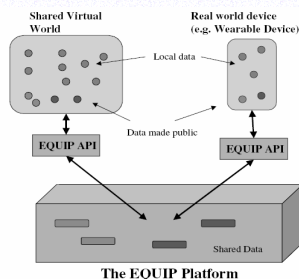
- See <http://www.toolkit.equator.ecs.soton.ac.uk/infr astructure/repository/equip/web/>
- See also Greenhalgh et al. *The EQUIP Platform: Bringing Together Physical and Virtual Worlds*. Equator Technical Report. <http://www.crg.cs.nott.ac.uk/~cmg/Equator/Downloads/docs/equip-platform.pdf>

HC14 2006

UbiComp Software Architecture & Technology 1

16

## Equip Architecture

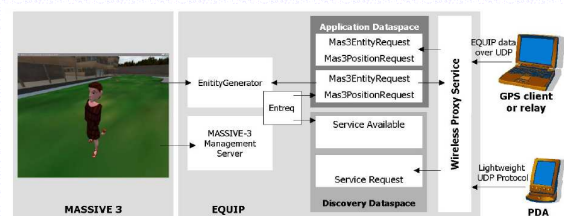


HC14 2006

UbiComp Software Architecture & Technology 1

17

## Equip plus VR



HC14 2006

UbiComp Software Architecture & Technology 1

18

## Case Study: The Context Toolkit

- Developed at Georgia Tech by Anind Dey, Daniel Salber and Gregory Abowd
- Provides services for handling context information
- Available for research purposes
- Still being developed via Open Source project
- Remainder of this talk based on materials from Dr Anind Dey
- See Dey, Abowd and Salber. *A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications*. HCI Journal, 16 (2-4), 2001, pp. 97-166.

HCI4 2006

UbiComp Software Architecture & Technology 1

19

## Dey & Abowd's Definition of Context

- Context is any information that can be used to characterise the situation of entities (whether a person, place or object) considered relevant to the interaction between a user and an application, including the user and application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects.

HCI4 2006

UbiComp Software Architecture & Technology 1

20

## Difficulties with Handling Context

- Unlike user input, there is no common way to handle context
  - » comes from many computers
  - » requires additional abstraction
  - » Context sensing components typically not coupled to application
- Context-aware applications are usually built
  - » in an ad hoc manner
  - » heavily dependent on underlying technology

HCI4 2006

UbiComp Software Architecture & Technology 1

21

## Requirements of Handling Context Input

- Separation of Concerns
- Context Interpretation
- Transparent, Distributed Communication
- Constant Availability of Context
- Context Storage
- Resource Discovery

HCI4 2006

UbiComp Software Architecture & Technology 1

22

## Context Toolkit Abstractions

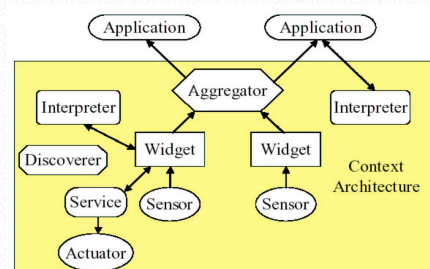
- Widgets
  - » Provide a sensor proxy
- Interpreters
  - » Abstraction services
  - » e.g., badge id to name translation
- Aggregators
  - » Collect logically related data
  - » E.g., a person tracker based on building based sensors
- Services
  - » Actuators for controlling physical devices
- Discoverers
  - » Registry of available capabilities
  - » Resource discovery services

HCI4 2006

UbiComp Software Architecture & Technology 1

23

## Conceptual Framework



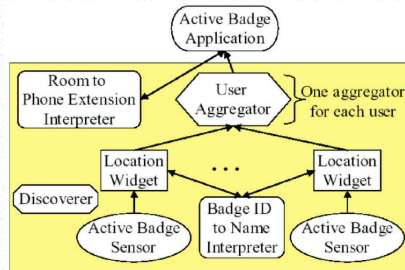
HCI4 2006

UbiComp Software Architecture & Technology 1

24



## Active Badge Forwarding

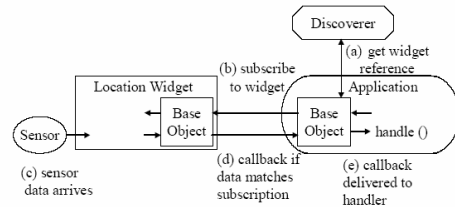


HCI4 2006

Ubicomp Software Architecture & Technology 1

25

## Using a context widget

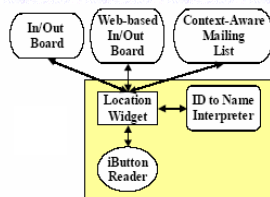


HCI4 2006

Ubicomp Software Architecture & Technology 1

26

## Multiple Use of Context



HCI4 2006

Ubicomp Software Architecture & Technology 1

27

## Other Important Software Tools for Ubicomp

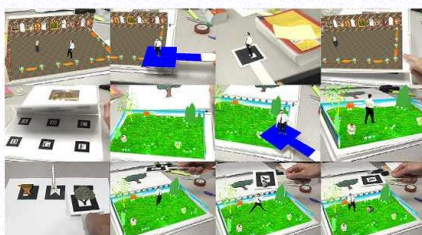
- AR Toolkit
  - » software library for building Augmented Reality (AR) applications that involve the overlay of virtual imagery on the real world
  - » Includes functions for feature recognition from image and video stream data

HCI4 2006

Ubicomp Software Architecture & Technology 1

28

## 3d live Malaysian Tourist Book



HCI4 2006

Ubicomp Software Architecture & Technology 1

29

## Other Important Software Tools for Ubicomp

- Phidgets
  - » toolkit for constructing tangible user interfaces or other systems that use sensors and actuators
  - » Includes
    - Hardware
    - A/D interface
    - Drivers and reusable application components
    - Set of application building tools

HCI4 2006

Ubicomp Software Architecture & Technology 1

30

## Plasticity

- Volatility of the interactive devices in a mobile and ubiquitous setting makes development difficult
- Want to be able to
  - » deploy the “same” application on different devices
  - » Maintain consistency across multiple platforms
  - » Change the application (and/or UI) to reflect changing context

HCI4 2006

Ubicomp Software Architecture & Technology 1

31

## Plasticity

- Differences can be non-trivial
  - » Content segmentation/filtering for WML vs HTML-based output
  - » Controls/layout may have to change
  - » Some tasks may become impossible or have to be modified

HCI4 2006

Ubicomp Software Architecture & Technology 1

32

## Plasticity

- Plasticity is the capacity of an interactive system to withstand variations of context of use while preserving usability
- Plasticity sub-properties can include
  - » Efficiency
  - » learnability
  - » enjoyment
  - » Responsiveness
- See Calvary et al. *A Unifying Reference Framework for Multi-Target User Interfaces*. IwC 15,3 (2003), pp. 289-308.

HCI4 2006

Ubicomp Software Architecture & Technology 1

33

## Achieving Plasticity

- Current practice
  - » Transformers
    - Linearisation of HTML, HTML -> WML with segmentation
  - » Alternative implementations of widgets
    - Customisable installation
    - AppForge CrossFire for C Sharp
      - implementation of widgets for multi-devices using .Net and VB

HCI4 2006

Ubicomp Software Architecture & Technology 1

34

## User Interface Plasticity

- Languages for Multi-Platform Deployment
  - » UIML (User Interface Markup Language)
  - » AUIML (Abstract User Interface Markup Language)
  - » XUL (eXtensible User-interface Language)
  - » ... amongst many others

HCI4 2006

Ubicomp Software Architecture & Technology 1

35

## UIML

- Developed at Virginia Polytech
- Device-independent declarative representation
- Renderers for HTML, Java, WML, PalmOS and VoiceXML
- Still concentrates on surface features of the interaction

HCI4 2006

Ubicomp Software Architecture & Technology 1

36

# AUIML

- Abstract User Interface Modelling Language
- XML dialect
- Developed by IBM
- platform and a technology-neutral representation of panels, wizards, property sheets, etc.
- display handled by platform-specific renderer
  - » Current versions for Swing and HTML
  - » decides the best way to present the user interface to the user and receive user input.
- AUIML XML output not intended to be read
  - » created using the Eclipse-based AUIML

HC14 2006 Ubiocom Software Architecture & Technology 1 37

# AUIML Example: Specification

```

<GROUP NAME="PersonName">
  <CAPTION>
    -META-TEXT-Person's complete name-/META-TEXT-/
  </CAPTION>
  <CHOICE NAME="personTitles" SELECTOR-POLICY="SINGLE">
    <CAPTION>-META-TEXT-Title-/META-TEXT-/</CAPTION>
    <ITEM>
      -META-TEXT-This is a set of valid titles you may choose
      from-/META-TEXT-/
    </ITEM>
    <STRING NAME="He">
      <CAPTION>-META-TEXT-He-/META-TEXT-/</CAPTION>
    </STRING>
    <STRING NAME="HES">
      <CAPTION>-META-TEXT-Hes-/META-TEXT-/</CAPTION>
    </STRING>
    <STRING NAME="HIS">
      <CAPTION>-META-TEXT-His-/META-TEXT-/</CAPTION>
    </STRING>
    <CHOICE NAME="He">
      <CAPTION>-META-TEXT-He-/META-TEXT-/</CAPTION>
    </CHOICE>
    <STRING NAME="First Name">
      <CAPTION>-META-TEXT-First Name-/META-TEXT-/</CAPTION>
    </STRING>
    <STRING NAME="Initial">
      <CAPTION>-META-TEXT-Initial-/META-TEXT-/</CAPTION>
    </STRING>
    <STRING NAME="Last Name">
      <CAPTION>-META-TEXT-Last Name-/META-TEXT-/</CAPTION>
    </STRING>
  </GROUP>

```

HC4 2006

Ubicom Software Architecture & Technology 1

38

# AUIML Example: Rendered Output

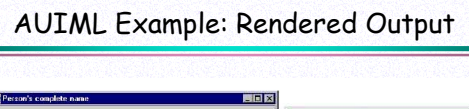
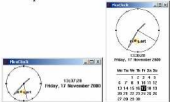


Figure V.9 – Two examples of concrete user-interfaces automatically generated from an AUIML document through: (i) a JavaSwing renderer and (ii) a DHTML renderer


# Types of Adaptation to Platform

static



The static UI consists of two windows. The left window, titled 'Clock', shows a standard analog clock face with a digital display at the bottom showing '12:00 PM' and 'Monday, 10 September 2006'. The right window, titled 'Tasks', shows a list of tasks: 'Task 1', 'Task 2', 'Task 3', 'Task 4', 'Task 5', 'Task 6', 'Task 7', 'Task 8', 'Task 9', 'Task 10', 'Task 11', 'Task 12', 'Task 13', 'Task 14', 'Task 15', 'Task 16', 'Task 17', 'Task 18', 'Task 19', 'Task 20', 'Task 21', 'Task 22', 'Task 23', 'Task 24', 'Task 25', 'Task 26', 'Task 27', 'Task 28', 'Task 29', 'Task 30', 'Task 31', 'Task 32', 'Task 33', 'Task 34', 'Task 35', 'Task 36', 'Task 37', 'Task 38', 'Task 39', 'Task 40', 'Task 41', 'Task 42', 'Task 43', 'Task 44', 'Task 45', 'Task 46', 'Task 47', 'Task 48', 'Task 49', 'Task 50', 'Task 51', 'Task 52', 'Task 53', 'Task 54', 'Task 55', 'Task 56', 'Task 57', 'Task 58', 'Task 59', 'Task 60', 'Task 61', 'Task 62', 'Task 63', 'Task 64', 'Task 65', 'Task 66', 'Task 67', 'Task 68', 'Task 69', 'Task 70', 'Task 71', 'Task 72', 'Task 73', 'Task 74', 'Task 75', 'Task 76', 'Task 77', 'Task 78', 'Task 79', 'Task 80', 'Task 81', 'Task 82', 'Task 83', 'Task 84', 'Task 85', 'Task 86', 'Task 87', 'Task 88', 'Task 89', 'Task 90', 'Task 91', 'Task 92', 'Task 93', 'Task 94', 'Task 95', 'Task 96', 'Task 97', 'Task 98', 'Task 99', 'Task 100'. The tasks are listed in a single column, each with a small icon and a text label.

dynamic



The dynamic UI is represented by two overlapping rectangular frames. The left frame contains a red gear icon with the text 'Glo' inside it. The right frame contains a red gear icon with the text 'ss' inside it. This illustrates how the UI content can change dynamically based on the platform or context.

HC14 2006

Ubicomp Software Architecture & Technology 1

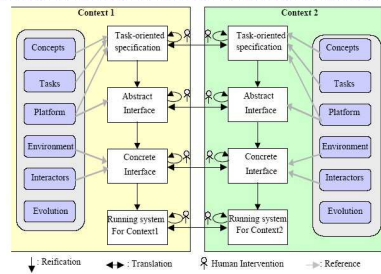
40

Figure 1 consists of two screenshots of a temperature controller interface. Screenshot (a) is a large screen showing a list of rooms: 'kitchen', 'bed room', 'bathroom', and 'living room'. Each room has a horizontal bar chart representing temperature over time, with markers at 0, 6, 12, and 18. The bars are color-coded: red for high temperature, yellow for medium, and purple for low. Screenshot (b) is a small screen showing the same interface but with only one room selected, 'bed room', and its temperature bar chart displayed. Both screens have 'Okay' and 'Cancel' buttons at the bottom.

Figure 2 consists of three sequential screenshots of a WAP-enabled mobile phone screen, labeled (a), (b), and (c). Each screenshot shows a different stage of modifying a temperature setting.

- (a)** The screen displays the text "Chaudhary" at the top, followed by "Set temperature zone" and "Quoted:". Below this, it shows "10.0000" and "TEMP: 10.0000". At the bottom, there are two buttons: "Options" and "Back".
- (b)** The screen displays the text "Chaudhary" at the top, followed by "Temperature (C): 10" and "Convert temperature". Below this, it shows "10.0000" and "TEMP: 10.0000". At the bottom, there are two buttons: "Options" and "Back".
- (c)** The screen displays the text "Chaudhary" at the top, followed by "CONTEMP: 10.0000" and "CONTEMP". Below this, it shows "10.0000" and "TEMP: 10.0000". At the bottom, there are two buttons: "Options" and "Back".

## Reference Model for Plasticity

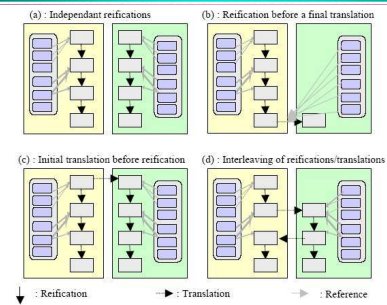


HCI4 2006

UbiComp Software Architecture & Technology 1

43

## Variants of Plasticity



HCI4 2006

Fig. 7. Instantiations of the reference framework.

44