The universAAL UI Framework

saied.tazari@igd.fraunhofer.de
Note

the abbreviation “UI” throughout this presentation stands for *User Interaction*

and not for *user interface*
Outline

- Rationale behind the universAAL Approach
- The universAAL Approach
- Resources
RATIONALE
From HCI to HEI!

- **HCI**: the traditional Human-Computer Interaction
  - the interaction is usually assumed to be bound to one single computer and its peripherals.

- **HEI**: Human-Environment Interaction
  - Implicit interaction
  - Explicit interaction
Breaking out of the Virtual Realm into the Physical World

T. Berners-Lee, J. Hendler, O. Lassila: The Semantic Web
The Notion of Channel
Explicit UI over I/O channels long enough in the shadow of “implicit interaction” over sensing channels in AmI
Progresses that help explicit UI become more important

- proliferation of (multi-)touch sensing, HD displays, & displays embedded in all possible devices
- new interaction forms supported by special devices with specific sensors
- qualitative progresses in
  - speech recognition
  - natural language processing
  - gesture recognition
- socio-political pressure on “accessibility for all”
I/O Devices in emerging Smart Homes

- living room TV
- sleeping room TV
- a display in the entrance
- a display integrated in the fridge door
- mirrors capable of becoming displays
- microphone arrays installed in all rooms
- loudspeakers installed in all rooms
- phones providing displays, microphones, (loud)speakers
- hi-fi providing loudspeakers
- .
- .

An infrastructure of available I/O channels
Smart Environments as Open Distributed Systems

- Handling distribution & heterogeneity
- "Error 404 Not Found"
- No total consistency
- User experience

Smart Environment is best realized as
The Web
Self-organizing System

Open Distributed System poses concern
facilitates independent development of Software Artifacts

Evolvable
Dynamic
Always working
Competitive
Surprising
The Consequence

Separating applications from the management of the I/O channels

(“UI Handler” is the term used for the managers of the I/O channels)
UIM Derived Goals

- We need to create a UI model for
  - describing user interfaces in a modality-neutral manner
  - performing personalized and context-aware adaptation

- Intelligent (personalized and context-aware) brokerage between applications and UI Handlers (I/O channel managers)

- Introduce a framework for
  - modality fusion when capturing user input from different input channels
  - modality fission when using different output channels for presenting system output to human users
APPROACH
UI Protocol

user interaction communication needs

node$_1$

UI handler

handling request → responding

user interaction handling requests

requesting → processing response

UI caller

caller

processing response → requesting

caller

publishing event → publisher

events

subscriber

handling event

handling request → responding

callee

is a special case of

general inter-component communication needs
UI HANDLERS & THE MANAGEMENT OF I/O CHANNELS
I/O Channels are bound to
- A certain **location**
- A certain **modality**
- Possibly, a “**privacy level**”

➢ Important for adaptation: which channels are more appropriate for a given user in a given situation?!
Accessing Channels

- Channel Binding
  - EmbeddedBinding
    - ContextPublisher
      - may implement: ServiceCallee
  - Driver
    - may be a wrapper of: Legacy Driver
    - wraps: Legacy Applied Framework
  - Software
    - is a: uAAL-aware Driver
Relationship to UI Handlers

UI Handler

may use any of

Legacy Driver

uAAL-aware Driver

Legacy Applied Abstract Driver

may be a wrapper of

wraps
Approach

ADAPTATION FAMEWORK
Context Awareness: using the rights channels
Context Awareness: Follow me without loss of data
Supporting the UI Bus in Adaptation

Parameters provided by the app

- Addressed user
- Content language & privacy level
- Dialog priority

Parameters added by the UI Framework

- the presentation location and modality
- access impairments to be considered
- modality-specific recommendations
Capabilities of the UI Handlers

- Appropriateness for certain access impairments
- Supported languages, modalities & privacy levels
- Locations where output can be presented
- Modality-specific tuning capabilities

(recall that UI handlers are the managers of I/O channels & that channels are bound to specific modalities, privacy levels & locations)
Approach

DESCRIBING A DIALOG
Need for Declarative Languages

- A direct consequence of separating application layer from the presentation layer

analogy to the WWW

| browsers | language = HTML | protocol = HTTP | Web applications |
The problem with HTML

- Not really modality-neutral
- Sometimes posing certain layout
- More abstract and neutral languages investigated since more than 10 years:
  - UIML
  - TERESA XML
  - UsiXML
  - SMIL
  - EMMA
  - XISL
  - XForms
XForms - Separation of Values from Controls

- XForms separates data and the underlying model from presentation:
  - The model specifies the values being collected (the instance), and their related logic
    - Types, restrictions
    - Initial values, Relations between values
  - Logical UI Controls with binding to the model

Source: www.w3.org/2006/Talks/05-26-steven-XForms/
Current solution inspired by XForms

- Apparently the most advanced form-based solution
- Separating the form UI description from the form data

- Define a “dialog package” based on XForms UI controls
- Use own RDF-based data model instead of adding a new complexity
The Dialog Package

- FormControl
  - ctrlRestriction
  - helpString
  - hintString
  - referencedPPath

- Label
  - labelText
  - iconURL

- ChoiceItem
  - value

- ChoiceList
  - subchoices

- Submit
  - submissionID
  - confirmationMessage
  - confirmationType

- Input
  - alertString
  - mandatoryInput

- Group
  - parentGroup

- Select
  - choices

- Select1

- InputField
  - isSecret
  - maxLength

- TextArea

- Output
  - SimpleOutput
    - content

- MediaObject
  - contentType
  - contentURL
  - resolutionMaxX
  - resolutionMaxY
  - resolutionMinX
  - resolutionMinY
  - resolutionPreferredX
  - resolutionPreferredY

- Form
  - creationTime
  - data
  - dialogType
  - title

- SubdialogTrigger
  - selectionXIDPrefix

- ioControls, stdButtons, submits

Eindhoven, 27-Sep-2012
Approach

MISCELLANEOUS
More on the Dialog Manager

- Coherent representation of the whole system
  - Management of Dialogs
    - Per user & priority-based management of dialog queues
    - Suspending dialogs and continuing later
  - Providing the system main menu
  - Handling context-free input
Support for Multimodality

- Delegated to UI handles...

- An example developed within PERSONA
  - On the input side: fusion of speech & gesture
  - On the output side: speech synchronized with visual feedback
Resources

- **www.universaal.org**, esp.
  - all deliverables immediately after release
  - Newsletters, publicity material, comic
- **depot.universaal.org**, the entry point for developers (reachable also through the home page)
  - Getting started developing AAL applications
  - Learning more about the platform & contributing to the development of the platform
- **forge.universaal.org** (reachable also through the Developer Depot) with
  - source codes, Javadocs, & Wiki Pages
  - forum discussions
THANK YOU FOR PAYING ATTENTION!

Questions?

Mohammad-Reza (Saied) Tazari
Fraunhofer-Institut für
Graphische Datenverarbeitung IGD
Fraunhoferstraße 5
64283 Darmstadt

Tel  +49 6151 155 – 228 | Fax – 480
saied.tazari@igd.fraunhofer.de
www.igd.fraunhofer.de