SmartMote – A run-time adaptive universal control device for ambient intelligent production environments

W3C MBUI WG Presentation
9. February 2012 @ DFKI, Kaiserslautern

Dipl.-Inf. Marc Seißler
Dr.-Ing. Gerrit Meixner
Dipl.-Inf. Kai Breiner
Agenda

- Mobile Human Computer Interaction in Industrial Environments
- Benefits of Context-Sensitive Systems
- Development Methodology
  - Related Work
  - Shortcomings & Requirements
  - Concept description & Meta-Model Architecture
  - Abstract Modeling Phase
    - The Useware Markup Language (useML)
    - The Useware Dialog Modeling (useDM) Language
Mobile Human Computing Interaction in Industrial Environments

Interaction Zones: Use of Context-Information can help to resolve these issues
Location-based User Interface Adaptation – Hazardous interactions

1. Track user in environment
2. Match with interaction zones
3. Disable/Enable hazardous functions in interaction device

Positioning System

Safe Interaction Zone

Safety Critical Interaction Zone

Universal Interaction Device

move arm
X- Y- Z-
X+ Y+ Z+
Location-based User Interface Adaptation – Workflow support

1. Track user in environment
2. Match with interaction zones
3. Display device functions

Positioning System

Robot Interaction Zone

Pump Interaction Zone

Universal Interaction Device

move arm

X- Y- Z-

X+ Y+ Z+
Development Methodology
Related Work

Promises of MBUID: Use of semi-formal Methodologies to...
• give different perspectives on UI
• support seamless UI Engineering
• increase reuse and automation in E.-Process
• ...

Shortcomings & Observations:
Review of Related Work shows that...
• different **types of models** are used.
• the **models expressiveness** varies.
• there is only **limited tool-support** in early phases.
• automatic generation often results in **unusable User Interfaces**.
• concepts only offer **limited flexibility** and manual intervention.
• **they often do not provide a clear separation of concerns in early phases**
• ...

Architectures & Languages

CAMELEON Reference Framework
Dygimes MASP TERESA
Useware Architecture
UsiXML MARIATerrestrial Remote Console

Center for Human-Machine-Interaction

© ZMMI 2012-7
Requirements

Methodology

- **Explicit Specification** of Task, Context, Dialog, Presentation and Adaption of User Interface!
- Integration of **Backend Functions** in Task Modeling Phase!
- Give Developer *more control in early development phases*!
- Clear separation of concerns:
  - Task Model → Functional Requirements
  - AUI Model → Interaction specification
  - Distinguish between *development time* and *run-time models*

User Interface Description Language(s)

- Use of **Light-weight models**!
- Avoid redundancies!
- Provide *clear element mappings*!

Tool-Support

- Support **graphical modeling**!
- Support **interactive model transformation** in early development phases!
- Support **run-time Interpretation**!
SmartMote Meta-Model Architecture

- **Domain Model**
  - Describes Backend-Functions

- **Function Model**
  - uses & adapts

- **Dialog Model (useDM)**
  - Structure + Behavior
  - uses

- **Task Model (useML)**
  - Which functions, which task, in which context situation

- **Context Model**

- **Presentation Model (UIML)**
  - Presentation
    - Style + Layout
  - uses

- **Adaptation Model**
  - If Interaction Zone B, then:
    - Show dialog 2
  - uses & adapts

- **Interaction Zones**
  - Safety-Critical IA Zones

---

Analysis Models

Design & Runtime Models

Center for Human-Machine-Interaction

© ZMMI 2012-9
Useware Markup Language 2.0

- task structure
  - platform independent

- abstract tasks

- atomar interactions

Different tasks types (e.g. system task, interactive task)
Elementary use objects → more detailed specification of interactive tasks
Optionality and Cardinality
Logical and temporal conditions (pre-conditions, invariants, post-conditions)
5 Temporal Operators
  + Interface to functional backend
  + Read-/Write Collections
Useware Dialog Modeling (useDM) Core Meta-Model

- Modality-Independent Interactors
- Event-based Behavior Model
- Relative + absolute Navigation descriptions
- Enhanced Reuse concept via templates
- Extensible Presentation Semantics
Use of extendible, semantic Selectors

Useware Dialog Modeling (useDM) Meta-Models

Useware Dialog Modeling (useDM) Language

useDM Core Meta-Model

- Dialog
- Interaction Objects
- Input
- Output
- Trigger

useDM Presentation Types Meta-Model(s)

- Output Presentation Types
  - LargeText
  - NumbersOnly
- Trigger Presentation Types
  - Next
  - Back
  - Home

Invariant Core Model

Project-specific semantic presentations

<input id="iEnterSpeed" presentation-type="numberOnly" />

<input id="iEnterSpeed" presentation-type="textOnly" />

<input id="iEnterSpeed" presentation-type="shortText" />

<input id="iEnterSpeed" presentation-type="longText" />
Concept Description Development Path – Task Modeling

**Interactive Tool-Support (Udit 2.0)**

- **Task Model (useML)**
- **Dialog Model (useDM)**

**Core Analysis Model**
- Development Model
- **Core Design Model**
- Development + Run-time Model

**Drag’n’Drop**

- **Task Model (useML)**
- **Dialog Model (useDM)**

**Specify**
- (Task-based) Functional Requirements
- Context Information

**Group Tasks → Dialogs**
- Add Navigations
- Use Abstract Interaction Objects
- Specify Behavior
- Specify Adaptations

© ZMMI 2012-13

Center for Human-Machine-Interaction
Example Use Case Task Modeling

Function Model

Use Model

Context Model

**Use Model**

- UO "Module A Settings"
- UO "Pump Settings"
- UO "Edit Pump Parameters"

**Context Model**

- Module A
- Module B
- Module C
- Safe
- Unsafe

**Interaction Zones**

- Read-/Write Collection: Read-/Write eUO Information at once

**Operation**

- getPower
- setPower
- getMode
- setMode
- ... (Human-Machine-Interaction)

**Function Model**

- Module B
- Module A

**Interaction Zones**

- Safe
- Unsafe

© ZMMI 2012-14
Example Use Case Dialog Modeling

Context Model

Function Model

Task Model (useML)

Adaptation Model

Dialog "Module A Settings"

Container "Pump Settings"
- AIO "input": Enter Speed
- AIO "select": Pump Mode
- AIO "trigger": Write Parameters
- AIO "trigger": Read Parameters

Container "Navigation"
- AIO "trigger": Power on/off
- AIO "trigger": Previous Dialog
- AIO "trigger": Next Dialog

Navigation
Mapping Example useDM

useDM XML Snippet

```xml
<dialog id="DialogA" title="Dialog A Settings">
  <structure>
    <container id="cPumpSettings" title="Pump Settings">
      <input title="Enter Speed:" id="iEnterSpeed"
        presentation-type="numberOnly" variable-ref="varEnterSpeed"/>
      ...
      <trigger title="Write Parameters" id="tWriteParameters" onTriggered="writeValues"/>
      <trigger title="Read Parameters" id="tReadParameters" onTriggered="readValues"/>
    </container>
    <trigger title="Letztes Modul" id="back"
      presentation-type="previous-dialog" onTriggered="previousModule"/>
    <container id="cNavigation" title="Navigation" presentation-type="navigation">
      ...
    </container>
  </structure>
  <behavior>
    <variable id="varEnterSpeed" datatype="string"/>
    <transition id="readValues">
      <call function-name="pump_getSpeed">
        <return-value variable-ref="varEnterSpeed"/>
      </call>
    </transition>
    <transition id="previousModule">
      <relative-target type="previous-dialog"/>
    </transition>
    ...
  </behavior>
</dialog>
```
Summary and Conclusion

• **Addressed Problems**
  - **Explicit Specification** of Task, Context, Dialog, Presentation and Adaption of User Interface!
  - Integration of **Backend Functions** in Task Modeling Phase!
  - Give Developer **more control in early development phases**!
  - Provide a **clear separation of concerns**!

• **Issues Out of Scope**
  - Automatic Layouting
  - Multi-Modal Fusion

• **Future Work**
  - **Finish & Publish tool-support:**
    - Udit 2.0 (estimated: July 2012)
    - Renderer (estimated: May 2012)
  - **Evaluation of Modeling Concept**