W3C MBUI Face to Face Meeting

Red Hat

Munich is the European head quarter of Red Hat. It's an Open Source Service Provider for Hyper Cloud Solutions, Operating Systems and Application Servers. They provide training lessons and consultees. They want to use MBUI approaches for their Java middle ware components.

Why do they use MBUI:

- Need to degrease deployment time
- User Interfaces need experts
- All stakeholders participate and need a common language (designers, users, developers)

What they use:

- They have UI Kernel which implements an AUI which can be transformed into CUI
- They don't use Task Models because it's description is too low level for their needs
- They use useML
- They generate HTML5 UI (prototypes)

What they have learned:

- W3C AUI is a bit to detailed for them
- Structural parts are easy to implement but when it comes to behavioural parts it's getting hard
- Developers have no problems with web technology more with describing the UI in a proper way

What may be needed:

- Strong authorization access for UI Elements which embody security options
- Future form factors for different devices
- Proper tool chain is needed for AUI in particular for the behavioural parts (right now only XML)

Benefit of MBUI:

- Increase productivity => reduce time to market
- Provides a shared understanding of what's needed in the interface
- Consistency among dialogues
- Provides the possibility of progressive enhancements of their user interfaces

How MBUI is done:

Generate AUI without behaviour => Add behavioural parts => refine AUI => generate CUI => refine CUI to designers wishes ...

There is no common process but progressive enhancements.

Demo (Last Day):

- Low fidelity wire-frames tool <= many iterations => implementation of UI

- Interface Model (AUI) between Resource Model (Domain) and FUI //no CUI
- Interface Coordinator =perspectives based on Interface Model=> Dialogue =reification=> Interface Coordinator
- Dialogue uses resource input/output from Interface Model state which has to be mapped to something in the CUI (Events, Event Handler)
- A Scope Interactor is responsible for the mapping
- useML is used with project specific annotations (without them it won't work)
- The annotation is used as a shortcut container and is able to change the whole container and its content

Introduction & Working Draft Document

Discussion:

- Needs consistent naming
- Needs two appropriate CUI's
- Should promote the use of MBUI in a better way
  - platform independence
  - different (at least 2 well defined) context of use
- More specified functional requirements and links to them

AUI is platform independent but behaviour could differ.

Actions & Dates:

- Draw new CTT „Car Rental Example“ - done
- Create Use Case Document with „Car Rental Example“ - Friday
- Finish Introduction Document
- AUI Schema Update - done

Glossary

UAD Community Group

Community Group vs. Charters

- Anybody can join - you have to be a member of W3C
- no chart
- recommendations only - standardization

Why is MBUID interesting for UAD Community Group:

- Stronger dialogue between research and development
- Standardization
- Data and use cases generates value (others could benefit from that)
- Create open source tools

Developer View:

- Web designed for desktop
  - Use HTML5 for Win8, Android, IOS, Ubuntu Mobile
- Responsive design
  - Image versions to device characteristics (resolution)
  - Server side adaptation of content & styling through http request
  - Client side
    - CSS media query
    - Image source set attribute
    - Web page script
    - Jquery Mobile (Sencha Touch)

What knowledge can be provided by the MBUI Group:

- Desktop vs. Mobile
  - Context of use and displays
- Design Knowledge
  - Domain Model + Annotations
  - Task Model as a basis for dialogue design („Car Rental Example“)
- Layout expertise
  - Design patterns for UI on CUI and new kinds of controls
- Context awareness

Next Steps:

- Find a chair engage people to join
- Intel has shown interests
- Find developers for tools
- Create tutorials

Goal:

- Bring designers, developers and researchers together

Future of MBUI Group

Next Steps:

- Recommendation track
- Update task models
- Public working draft for AUI

    - Template issue

Interoperability Testing:

- 2 implementations needed (XML AUI format)
- Tools for task models which can be exported and then imported in AUI

  // I asked and Udit seems to be an solution

=>Should be finished till November<=

Option:

Stop MBUI Group=> Finish work (implementation, tests, discuss working draft) in community group => Candidate recommendation

AUI Template Issue

- Maybe in common package
- Template could be used in events and composition