Remote User Interfaces for Mobile Scenarios

Daniel Thommes, School of Computing,
Supervisors: Prof. Christos Grecos (UWS), Dr. Qi Wang (UWS), Prof. Ansgar Gerlicher (HdM)

Introduction & Hypotheses

For a better mobile remote user interface

Existing remote user interface protocols like VNC [1], RDP [2] and X-Server [3] are not suitable for remotely controlling mobile devices or remotely controlling arbitrary components with mobile devices. The reasons are

- different concepts of operation between desktop and mobile user interfaces
- missing applicability of performance-enhancing technologies
- a relative high energy consumption
- non-optimised bandwidth requirements.

The performance of a user interface protocol for mobile scenarios can be increased significantly in comparison to the protocols mentioned above. This can be achieved by applying technologies like graphic primitives for UI description (e.g. [4]), hierarchical prioritised transfer of UI widgets, caching and prefetching (e.g. [5]).

Scenarios

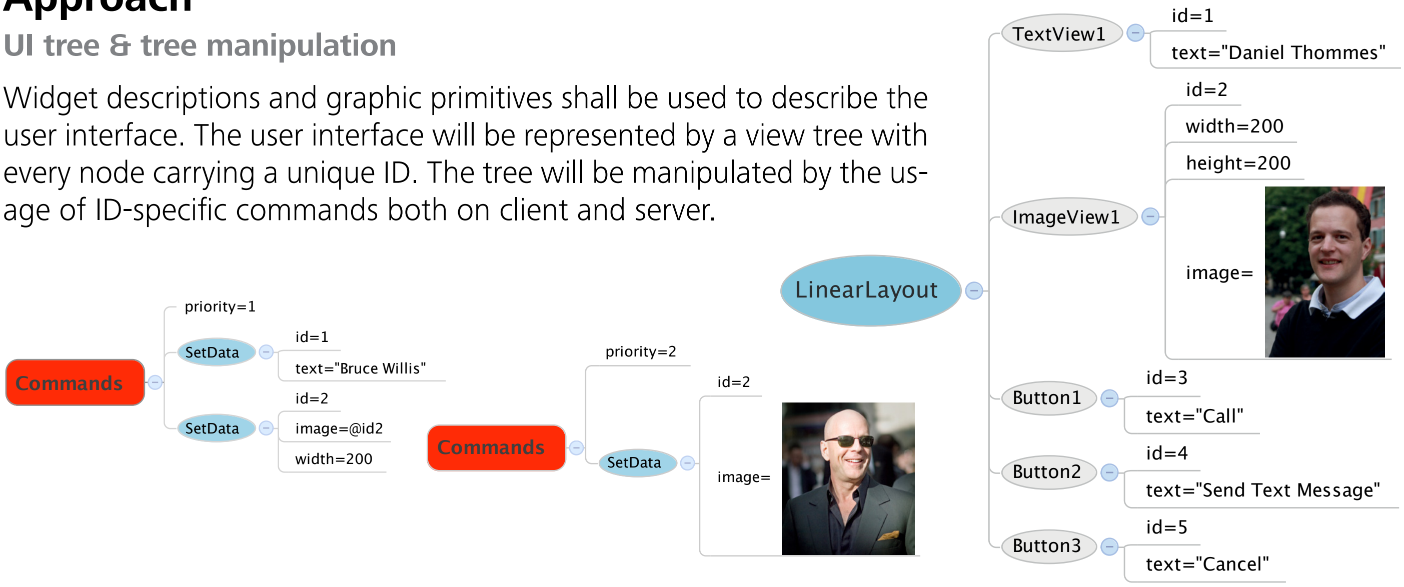
Where and why to use a mobile remote UI

- In-car remote control of personal mobile devices:
  A car driver controls his personal mobile device that is located in his pocket via the board computer's display of his car.
- Remote control for devices in a local area network:
  Fixed, powered but display-less devices within a local area network, e.g. media servers or smart home devices, use the protocol to serve their remote user interfaces.
- Remote-control via internet:
  A remote UI server that is reachable via internet shall be controllable with the new protocol.

Approach 1

UI tree & tree manipulation

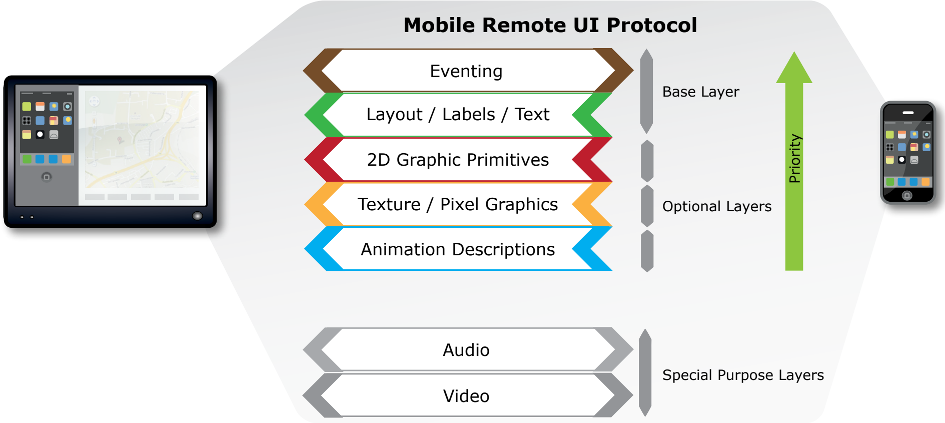
Widget descriptions and graphic primitives shall be used to describe the user interface. The user interface will be represented by a view tree with every node carrying a unique ID. The tree will be manipulated by the usage of ID-specific commands both on client and server.



Approach 2

Hierarchical protocol with prioritisation

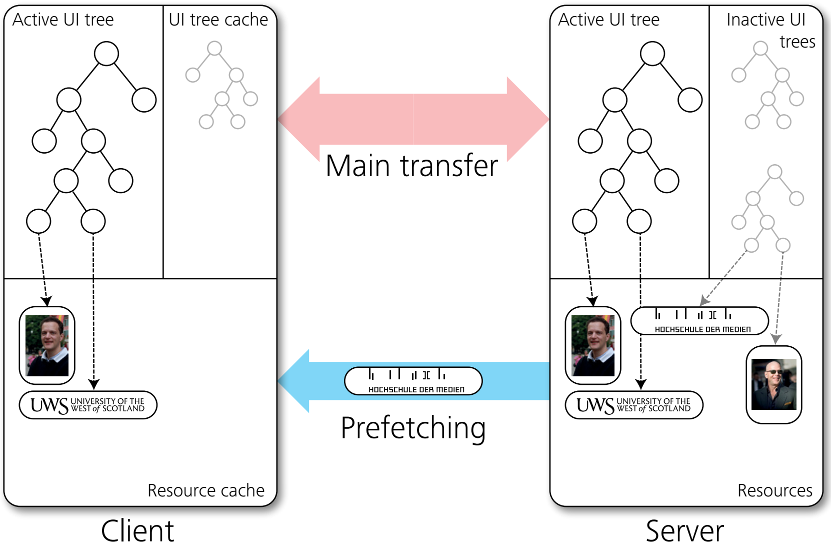
Based on the hypothesis, that the most important UI elements can be expressed by layout and widget descriptions, this type of information shall be prioritised. In low bandwidth environments this shall result in a faster rendering user interface. One challenge of this research project will be to define the right priorities for the different UI elements with the help of usability tests.



Approach 3

Caching & Prefetching

The resources of a remotely controlled application are known to the UI server at runtime. During network idle times they can be transferred to the client in advance. This process must be a low priority background process. The prefetching process must be interruptible and resumable to allow higher priority data (main transfer) to pass by.



References

[1] Richardson, T., 2010. The RFB Protocol - Version 3.8 (2009-11-24). Available at: http://www.realvnc.com/docs/rfbproto.pdf

[2] Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification. Available at: http://msdn.microsoft.com/en-us/library/cc240445(PROT.10).aspx [Accessed January 28, 2011].

[3] Documentation for the X Window System Version 11 Release 7.6 (X11R7.6). Available at: http://www.x.org/releases/X11R7.6/doc/index.html [Accessed January 24, 2011].

[4] Trewin, S., Zimmermann, G. & Vanderheiden, G., 2004. Abstract representations as a basis for usable user interfaces. Interacting with Computers, 16(3), p.477–506.

[5] Vankeirsbilck, B. et al., 2009. Bandwidth optimization for mobile thin client computing through graphical update caching. In Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian. p. 385–390.

[6] Tolia, N., Andersen, D.G. & Satyanarayanan, M., 2006. Quantifying Interactive User Experience on Thin Clients. Computer, 39(3), pp.46-52.

Contact

Dipl.-Ing.
Daniel Thommes

Nobelstr. 10
70569 Stuttgart
Germany

Tel.: +49 711 8923-2757
thommes@hdm-stuttgart.de
http://www.remoteui.org