A Framework Towards Challenges and Issues of Multi-Surface Environments.

Chi Tai Dang
University of Augsburg
Human Centered Multimedia
Universitaetsstr. 6a
Augsburg 86159, Germany
dang@hcm-lab.de

Elisabeth André
University of Augsburg
Human Centered Multimedia
Universitaetsstr. 6a
Augsburg 86159, Germany
andre@hcm-lab.de

Figure 1: Interactive portals with mobile devices and tabletop.

Abstract

In this paper, we discuss four of the most challenging technical issues for interactions and applications in environments comprised of nowadays widespread ecologies of touch-enabled mobile and immobile devices. Such issues are important particularly for applications in the wild. We address these issues by means of an appropriate software architecture that is implemented as the reference framework *Environs* in order to foster interactions and applications in multi-surface environments and help bring those into the wild. The framework is available as opensource software thereby we contribute to basic enabling technologies for multi-surface environments.

Author Keywords

Multi-Surface Environment; Portal; Lense; Framework; Cross-device interaction; Tabletop; Tablet; Smartphone; Mobile Device.

ACM Classification Keywords

D.2.6 [Programming Environments]: Interactive environments; H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: Miscellaneous

Introduction

During the last decade, people's interaction habit with daily devices has changed remarkably. While ten years ago peo-

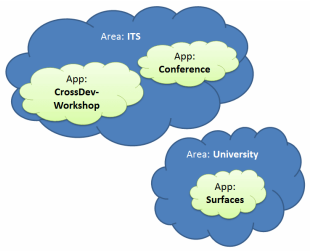

Figure 2: Partinioning concept for MSEs: areas as containers for application environments. Multiple logically separated application areas can exist within the same physical network.

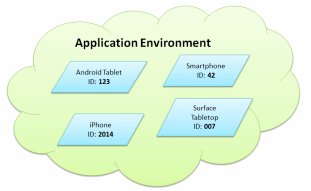

Figure 3: An example application environment comprised of multiple devices which are identified through a numerical ID.

ple were used to interact with mice, touchpads and keyboards only, today, interaction with touch-enabled surfaces has become an inherent part of nowadays interaction repertoire of everyday people. More and more touch enabled mobile (smartwatches, smartphones, phablets, tablets) and immobile (tabletops, wall-mounted display) devices entered the consumer market since around 2007/2008 and the amount of device types as well as proliferation of devices is still increasing rapidly. Hence, multi-surface environments (MSE) comprised of mobile and immobile interactive surfaces are quite likely to become commonplace in the foreseeable future. The increasing number of recent research articles targeting applications and interactions within MSEs and across multiple surfaces endorse this trend. With the increase of MSE occurrences, the desire for cross-device interactions and applications will inevitable rise. However, issues and questions originating from differences between lab environments and outside the lab environments need to be addressed by research to enable successful transition of cross-device interactions and applications from labs into the wild.

Outside the lab, Issues and Challenges

Even though HCI research investigated MSEs for many years with great results, studies were conducted in controlled *sterile* lab environments and situations. When going into the wild, things may be different and what worked in the lab may not necessarily work in the wild. Within this paper, we briefly discuss the most challenging issues from a technical point of view and present our research aiming at those issues.

Heterogeneity of platforms is the most challenging issue for interaction designers as well as for application developers. While the device ecology in a lab is manageable, device ecologies in the wild are literally wild. There are different form factors (smartwatch, smartphone, phablet, tablet, etc.), different set and kind of embedded sensors (accelerometer, gyroscope, GPS, heartbeat, etc.), different operating systems (Android, iOS, Windows Phone, etc.), or different programming APIs and platform languages (Java, Objective-C, C#, etc.). Even within one and the same device platform, the fragmentation of the operating system may result in a multitude of differences.

Network and device management are interrelated and not necessarily optimal in the wild. In terms of network, there may be environments with mobile data only, with wireless network but no internet access, or multiple logically separated/connected subnets with/without internet access. In terms of device management, devices usually take part in an ad-hoc manner and may vanish suddenly which is usually the case for decentralized loosely coupled devices. Both aspects together renders centralized server-based approaches quite difficult for robustness and stability of a system in the wild.

Performance, efficiency, stability, and low latency have direct influence on users' experience. Approaches that work perfectly well for one's lab devices may require further research for other device platforms or to be scalable across device platforms.

Security and safety of data on devices and on transport channels are usually neglected in lab studies. However, those aspects are nowadays mandatory requirements for applications in the wild. Users would behave different or prefer different strategies in studies if they know about the the safety of their data.

Development of MSE Applications

In software engineering and development practice, frameworks or toolkits are regularly used for recurrent and/or ab-

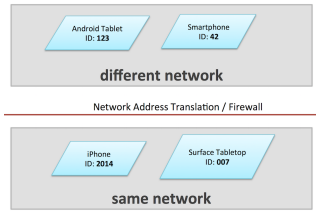

Figure 4: Supported network configurations of Environs: Devices within the same network (broadcast); Devices in different networks (STUN/STUNT mechanism).

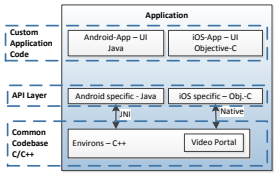

Figure 5: 3-layer architecture of Environs exemplified for the Google Android and Apple iOS platforms.

stract tasks, e.g. node.js, jQuery, or jQT. Hence, it's conceivable that this will also be the case for development of MSE applications. For example, a multi-platform framework that automatically handle network and connectivity, or device management and communication would greatly unburden developers from implementing the required logic for each supported platform and new application which is prone to errors. Considering the high complexity induced by nowadays heterogeneous device ecologies, such an MSE framework would also be beneficial for research studies and reproduction of research results. Previous research efforts [1, 5, 7, 8] that address MSE framework concepts further confirm this assumption. However, they did not target nowadays device ecologies and the challenges in the wild. Furthermore, lack of availability of the presented frameworks and portability of the concepts (e.g. to a smartwatch) often inhibit its reuse.

We briefly describe some highlights of the multi-platform MSE framework called *Environs* [2] which explicitly addresses the aforementioned issues and challenges. The reference implementation together with several introduction tutorials are publicly available[1] as opensource software so as to foster MSE research as well as development. Moreover, the framework easily enables real-time video-based interactive portals that open up a rich direction for demanding cross-device interactions and applications, e.g. aboard ships [4] or for collaborative tasks [3, 6].

Heterogeneity of platforms is handled by Environs through a 3-layer architecture for applications, see Figure 5. The *application layer* represents the actual application logic and UI that may be designed and auto-generated for multiple platforms by appropriate development tools. The framework itself is implemented in the remaining two layers, whereof the *API layer* provides a thin object oriented API to access the native layer. Under the hood, API objects merely keep object states and function as a proxy to native calls. The *native layer* is realized as a common code base for all platforms and contains the majority (~90%) of the framework logic which is implemented in portable C/C++. Hence, the whole native layer can be compiled for all platforms thereby greatly reduces development time, increases manageability and maintainability, and benefits from less programming errors. Currently, Environs supports the platforms Google Android/Android Wear, Apple iOS/WatchOS/OSX, Microsoft Windows (.NET/Surface 1/PixelSense 2/MultiTaction Cells), and Linux.

Network and device management is completely handled by the native layer which supports devices within the same network as well as devices across different networks. Environs manages so called application environments which can further exist in logically separated areas (e.g. meeting room, office, airport), see Figure 2, thereby enable multiple separated application environments within the same physical network. Each device assigns itself into an application area with a numerical ID, see Figure 3. Devices across different networks (e.g. both devices behind firewalls) require an additional mediator service which helps connect each other by means of STUN/STUNT mechanisms known from peer-to-peer networks. Overall, devices operate in a loosely coupled decentralized network (peer-to-peer), but server-like services are still possible through specialized device nodes.

Performance, efficiency, stability, and low latency is addressed through the native C/C++ implementation and native optimizations. Low latency network communication is based on priority handling of data types that employs low latency transport channels as well as a channel for large

[1] http://hcm-lab.de/environs

chunks of bulk data. For example, touch events are passed to other devices using shortest code paths and UDP channels. Furthermore, interactive portals make use of hardware encode/decode for low latency.

Security and safety is handled transparently in the native layer by state of the art AES encryption of transport channels. Each device automatically generates its own private/public key and certificate so as to encrypt AES session keys. This is particularly important when connecting multiple locally different MSEs to one application environment over unsecured networks (internet).

Author's Interests and Further Research

Our interest lies in research of natural and intuitive interaction techniques and enabling technologies for novel interactive portal applications in MSEs, which nowadays powerful touch-enabled device ecologies easily enable. Currently, Environs detects the location of devices within an MSE only by means of markers under mobile devices and only if they are placed on supported tabletops. Therefore, we intend to add additional position and spatial tracking of devices and users by means of location nodes within the MSE as proposed in [1] in order to investigate spatial interaction techniques for interactive portals.

Acknowledgments. The work described in this paper is partially funded by OC-Trust (FOR 1085) of the DFG.

REFERENCES

1. Apoorve Chokshi, Teddy Seyed, Francisco Marinho Rodrigues, and Frank Maurer. 2014. ePlan Multi-Surface: A Multi-Surface Environment for Emergency Response Planning Exercises *(ITS '14)*. ACM, New York, NY, USA, 219–228.
2. Chi Tai Dang and Elisabeth André. 2014. A Framework for the Development of Multi-display Environment Applications Supporting Interactive Real-time Portals. In *Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '14)*. ACM, New York, NY, USA, 45–54.
3. Chi Tai Dang, Masood Masoodian, and Elisabeth André. 2015. Private Focus Portals to Shared Energy Visualizations, Fostering Smart Energy Applications. In *Adjunct Conference Proceedings of INTERACT 2015*. Human-Computer Interaction - Lecture Notes in Computer Science, Vol. 9299. Springer International Publishing, 657–658.
4. Veronika Domova, Elina Vartiainen, Saad Azhar, and Maria Ralph. 2013. An Interactive Surface Solution to Support Collaborative Work Onboard Ships *(ITS '13)*. ACM, New York, NY, USA, 265–272.
5. Tony Gjerlufsen, Clemens Nylandsted Klokmose, James Eagan, Clément Pillias, and Michel Beaudouin-Lafon. 2011. Shared Substance: Developing Flexible Multi-surface Applications *(CHI '11)*. ACM, New York, NY, USA, 3383–3392.
6. Yucheng Jin, Chi Tai Dang, Christian Prehofer, and Elisabeth André. 2014. A Multi-Display System for Deploying and Controlling Home Automation. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA, 399–402.
7. Bonifaz Kaufmann, Martin Gratzer, and Martin Hitz. 2012. 3MF - A Service-Oriented Mobile Multimodal Interaction Framework. *PPD'12* (2012), 18–21.
8. Florian van de Camp and Rainer Stiefelhagen. 2013. GlueTK: A Framework for Multi-modal, Multi-display Human-machine-interaction *(IUI '13)*. ACM, New York, NY, USA, 329–338.