Workshop on W3C's Multimodal Architecture and Interfaces
November 16-17, 2007, Fujisawa, Japan

Kinesthetic input modalities for the W3C Multimodal Architecture

Authors

Florian Metze (Deutsche Telekom Laboratories), Thomas Ziem (T-Systems), Ingmar Kliche (T-Systems)

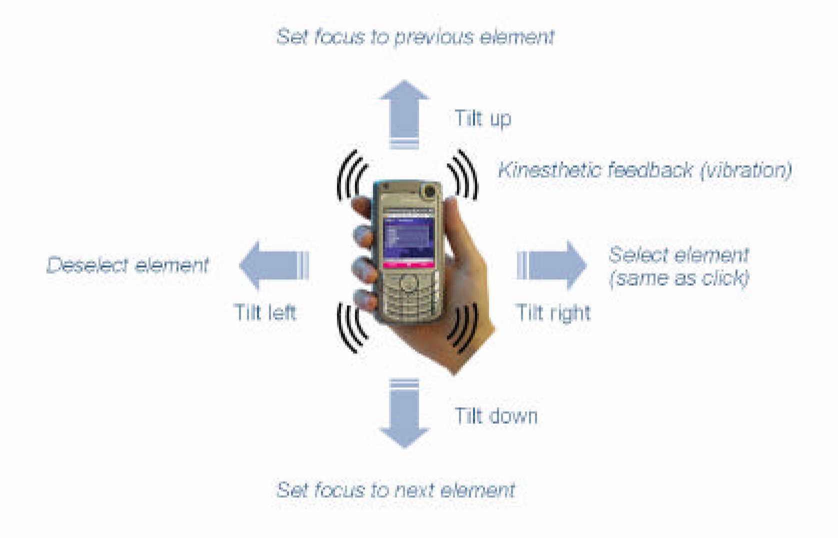
Corresponding author: Florian Metze, florian.metze@telekom.de

Abstract

Deutsche Telekom Laboratories and T-Systems recently developed various multimodal prototype applications and modules allowing kinesthetic input, i.e. devices that can be moved around in order to alter the applications' state. We developed first prototypes were using proprietary technologies and principles, whereas latest demonstrators more and more follow the W3C's Multimodal Architecture. This paper describes how we propose to integrate kinesthetic input into the multimodal framework as a new modality and discusses how this component fits into the W3C Multimodal Architecture.

Introduction

Mobile devices are getting more and more powerful while at the same time they are becoming smaller and smaller. Most of today's mobile handsets are equipped with cameras and there are first mobile phones shipped with acceleration sensors. These developments enable new input paradigms and capabilities, which provide intuitive input modes. Deutsche Telekom Laboratories and T-Systems have integrated various motion sensors based e.g. on acceleration sensors or analysis of a camera picture into prototypical applications, which could potentially profit from input gestures, in which the whole device is moved around. The motion patterns (e.g. tilt up, tilt down, tilt left, tilt right) are derived from raw sensor or camera output using digital signal processing. The output patterns are mapped to corresponding events and may be used within the application control logic as user input events to manipulate the user interface. In our first experiments, tilt up and down were proven to be effective for menu and list navigation. The general interaction logic is shown in the picture below. It follows a "sphere" metaphor, in that the motion patterns can be thought of as laying in the directions indicated on a virtual sphere ("tilt up" means tilting the upper part of the device down along a sphere's surface):


Set focus to previous element
Tilt up
Kinesthetic feedback (vibration)
Deselect element
Tilt left
Select element
(same as click)
Tilt right
Tilt down
Set focus to next element

This paper describes the technical integration of the motion detection sensor into the multimodal application and discusses the possibility to use it as a modality component within the W3C Multimodal Architecture. In analogy to the VoiceXML use case referred to in the W3C Working Draft on Multimodal Architecture and Interface [MMIARCH], we also introduce the cases of local and remote processing and discuss their advantages and disadvantages.

Kinesthetic modality component

The digital sensor (or camera) together with the digital signal processing can be seen as a modality component within the W3C Multimodal Architecture. It generates events corresponding to user motion inputs. To be integrated into the W3C Multimodal Architecture, this new modality component has to support the Modality Component API [MMIARCH].

We implemented and tested two different types of sensors for kinesthetic control of devices. Both types are operated under the assumption that the device is not translated (i.e. moved in 3D space), but rotated (i.e. turned) around two or three axes:

- Sensors based on accelerometers, returning tilt angle in 2D or 3D coordinates by measuring acceleration force
- Sensors evaluating the optical flow using camera pictures as described e.g. in [ROHS]: if the content of the pictures (which are captured by a camera in movie mode) moves to the left, this means the camera was rotated across a vertical axis

Both types of sensors are using digital signal processing with a predefined pattern matching algorithm to detect the respective motion patterns. As soon as a pattern is detected, the modality component generates a corresponding event.

Both types of modules are implemented in C/C++ as ActiveX components for Microsoft Windows Mobile 5 devices. The accelerometer based module on most devices requires an external hardware, whereas the camera based module makes use of the internal (preferably front) camera and therefore does not require any additional hardware.

State of the art prototype implementations

Our prototype applications were implemented using HTML + JavaScript and integration of the previously mentioned ActiveX components. The components are controlled using JavaScript and generate events corresponding to the user input. These events are handled by JavaScript functions which implement the control logic and manipulate the user interface accordingly. Therefore, in this scenario, the interaction control logic resides locally within the HTML/JavaScript pages.

Support for distributed interaction management

To support a distributed interaction management according to the W3C Multimodal Architecture the kinesthetic modality component has to support the MMI life cycle events. Within the life cycle events, we consider to use the *data event* to carry the user input information. We propose to represent the input itself using EMMA [EMMA].

Open Issues

We are currently working on an integration of the kinesthetic input modality into a distributed multimodal framework and as part of this effort we are designing an event structure for the representation of kinesthetic inputs.

A server side interaction management will introduce further latency into the event flow (e.g. a tilt event needs to be sent to the server, the server side interaction management sends a command to change the user interface according to the applications control logic to the client, e.g. to highlight the previous element in a list). This will lead to delayed responses at the clients user interface. Therefore it may be necessary to use a distributed interaction management, where (mainly) user interface related decisions are taken directly at the client device and other application control decisions are taken at the server portion of the interaction management.

Our current implementations of the kinesthetic input modules generate events based on predefined pattern matching rules (e.g. a “tilt up” is represented by a certain pattern of acceleration values at a given sensor corresponding to one axis). In the future, it might be useful to derive a pattern description language to describe motion inputs, comparably to grammars for speech recognition.

Furthermore, our current pattern matching takes a “digital” decision whether a predefined pattern has been recognized, or not. This allows a relatively simple implementation of motion patterns like “tilt up” or “tilt down”, which might be used to navigate through a menu list. More complex use cases might make use of the current tilt angle value, for example to browse pictures [TILT]. These types of user input require continuous events (e.g. every change of the tilt angle value) and require further investigation in terms of event representation and event flow.

Conclusion

In this paper, we presented a kinesthetic input modality component, which we implemented based on existing hard- and soft-ware technologies. We also presented a proposal on how to integrate our solution into the W3C Multimodal Architecture. In the future, we plan to integrate the kinesthetic components into our distributed multimodal framework.

References

[MMIARCH] W3C Multimodal Architecture and Interfaces, Jim Barnett et al., World Wide Web Consortium. http://www.w3.org/TR/mmi-arch/

[ROHS] The Smart Phone: A Ubiquitous Input Device; Rafael Ballagas, Michael Rohs, Jennifer Sheridan, and Jan Borchers; IEEE Pervasive Computing Journal, Vol. 5, No. 1; Jan-Mar 2006

[EMMA] EMMA: Extensible MultiModal Annotation markup language, Michael Johnston et al., World Wide Web Consortium, http://www.w3.org/TR/emma

[TILT] Dynamics of Tilt-based Browsing on Mobile Devices, Sung-Jung Cho, Roderick Murray-Smith, Changkyu Choi, Younghoon Sung, Kwanghyeon Lee, and Yeun-Bae Kim; Proc. CHI 2007; ACM; San Jose, CA; Apr/ May 2007