Rapid Prototyping of Multimodal Interactive Applications Based on Off-The-Shelf Heterogeneous Components

Jean-Yves Lionel Lawson[1], *Jean Vanderdonckt*[2], *Benoît Macq*[1]
[1]Communications and Remote Sensing Laboratory (TELE)
[2]Belgian Lab. of Computer-Human Interaction (BCHI)
Université catholique de Louvain (UCL), Belgium
{jean-yves.lawson, jean.vanderdonckt, benoit.macq}@uclouvain.be

ABSTRACT
OpenInterface Kernel is a lightweight open-source platform designed for supporting the effective prototyping of multimodal interactive systems. Iterative design of such applications requires the easy integration, replacement, interconnection or upgrade of components. OpenInterface provides a thin integration platform able to manage these key elements with little programming knowledge, and thus provide the research community a tool to fill the gap in the current support for multimodal applications implementation. The platform offers non-intrusive tools and techniques to assemble various modalities developed with different implementation technologies, while keeping a high level of performance of the integrated system.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. – Prototyping. D2.2 [Software Engineering]: Design Tools and Techniques – Modules and interfaces; user interfaces. D2.m [Software Engineering]: Miscellaneous – Rapid Prototyping; reusable software.

General terms: Design, Experimentation, Verification

Keywords: Prototyping, component-based architecture, reusable software component, multimodal interfaces, multimodal software architecture.

1. INTRODUCTION AND MOTIVATION

Prototyping is an important phase of the multimodal application development process, as it allows designers to iteratively build working applications. A prototyping tool must isolate users from low level programming details and enable them to *plug and play* with modalities, by easily combining them, and reusing the work done in previous stages with little low-level programming knowledge.

Available tools for developing multimodal interfaces usually support one programming language or specific implementation technologies and are still designed for a limited set of techniques, and interaction paradigms. The goal of our research is to provide a generic, multi-language, reusable and lightweight software –library, multimodality support and development tools– which could be used with minimal programming efforts as runtime platform for the realization of new multimodal toolkits, full multimodal applications or extension of existing multimodal toolkits.

UIST'08, October 19–22, 2008, Monterey, California, USA
ACM 978-1-59593-975-3/08/10.

2. OPENINTERFACE PLATFORM OVERVIEW

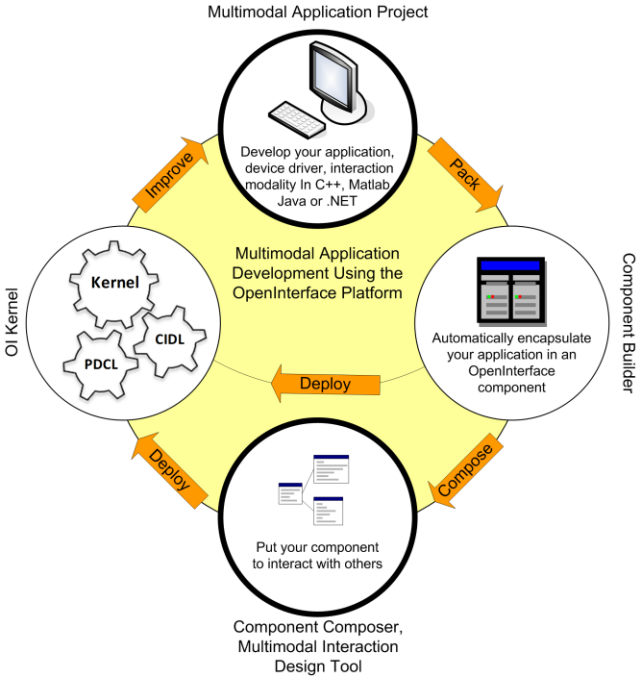

Figure 1: Multimodal Application Development Using the OpenInterface Platform

Overview

In OpenInterface, each component interface is described and registered into the repository using the XML-based Component Interface Description Language (CIDL). The C++ kernel then automatically generates proxies to perform the actual integration. Using a graphical front-end or the kernel API (allows embedding the platform within the final application) users can configure components and compose complex execution pipelines for implementing multimodal applications. Pipelines are also expressed in XML using the Pipeline Description and Configuration Language (PDCL) and are interpreted by the kernel.

Heterogeneous Components

We achieve the reusability requirement by adopting an extensible modular architecture in which *components* are the base objects manipulated by the *OpenInterface Platform* and follow the dependency injection pattern [1]. Within the system, a *component* is only characterized by its interfaces regardless of the implementation language or technology. The CIDL, which is automatically generated from provided source code, is used to declaratively describe inputs and outputs of components and to eventually generate C++ proxies for mediating the communication with other components.

This technique implies minimal programming efforts when integrating component and do not constraint the user to the use of a particular component model. It thus provides additional flexibility by allowing communication with existing component model implementations. Four programming languages are currently supported including C/C++, Java, Matlab and .NET.

Pipelines

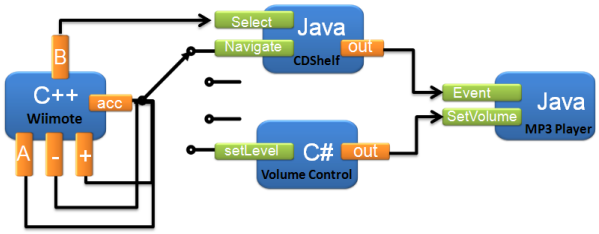

Figure 3: OpenInterface Pipeline of a simple multimodal music player.

In order to build running applications, we define the concept of *Pipeline* as an *interconnection and configuration of components* as illustrated in Figure 3. It allows control over the components' life-cycle and execution site (remote, local), and provides high level (threshold, filter, etc.) and low level (multicast, synchronization, etc.) data flow control for building up a complex systems. Extension points allow enhancing the pipeline's expressiveness.

	Integrated Component
Gestural	•Sketch Recognizer •Gesture Recognizer •Head Tracker (OpenCV) •Wiimote (wiiuse) •Finger Tracking •Data Gloves (5DT, etc.) •Touch Pad (Synaptics) •Face Tracker •Infra-Red dot Tracking (OptiTrack) •Stylus •Hand Posture (HandVU) •Mouse, Keyboard •ARToolkit
Visual	•Generic Pie Menu •Camera(Firewire, USB Webcam) •VirtualMouse •Virtual Pointer
Audio	•Speech Recognizer (Sphinx)

Figure 4: Selection of components currently integrated and available in the OpenInterface component database.

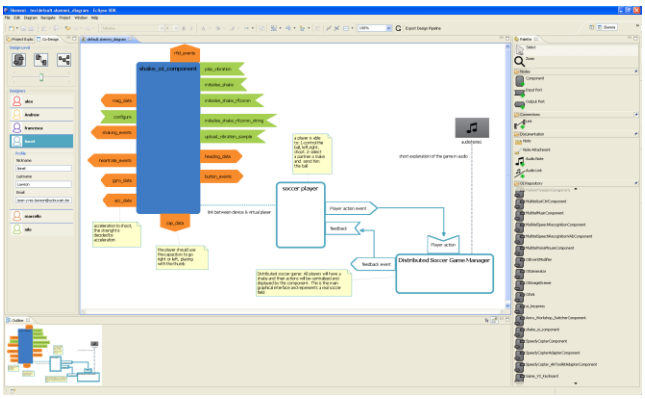

Figure 2: Skemmi Editor

Multimodal Fusion

To facilitate the implementation of functional multimodal systems, the platform embeds a set of reusable and generic fusion mechanisms [2] as well as low level signal fusion mechanisms but also offers a framework for adding new generic or tailored modality combinations at a semantic or feature level.

3. DEMONSTRATIONS

The demonstrations will showcase some components from the repository illustrated by Figure 4 within two applications: a multimodal music player and a multimedia database navigation system. Users will be able to run and quickly modify the multimodal interaction by using the OpenInterface Interaction Development Environment (OIDE) or *Skemmi* –Sketch Multimodal Interactions– a design-time graphical front-end illustrated by Figure 2.

4. CONCLUSION

The concrete implementation of a multimodal application design suffers from the lack of tools supporting this process. The *OpenInterface Kernel*, developed in C++, enables quick integration of multi-language components and reuse of existing components to iteratively design and build multimodal systems with minimal programming efforts.The platform, graphical front-ends and components are available for download from www.openinterface.org.

5. ACKNOWLEDGMENTS

The work on the OpenInterface platform described herein is mainly funded by the European FP6 SIMILAR Network of Excellence (www.similar.cc) and by the European FP6-35182 OpenInterface project (www.oi-project.org). Skemmi editor is part of a work done at Fraunhofer-Institut (FIT) and developed by Ahmad-Amr Al-Akkad.

6. REFERENCES

1. Inversion of Control Containers and the Dependency Injection pattern. http://martinfowler.com/articles/injection.html

2. Coutaz, J. Nigay, L. Salber, D. Blandford, A. May, J. and Young, R. (1995), Four Easy Pieces for Assessing the Usability of Multimodal in Interaction the CARE Properties, *Human Computer Interaction, Interact' 95*, pp. 115-120.