

Abstract

The communication between human and machine is rapidly changing with the introduction of new commodity hardware, such as Apple's iPad, HP's Sprout, Microsoft's PixelSense and Kinect. This hardware embeds novel input sensors to facilitate a more natural user interaction (NUI) paradigm. The development of NUI applications, where the machine tries to understand and anticipate the user's interaction, typically relies on a continuous monitoring of multiple input channels. The collection of events, the detection of relevant patterns and the embedding of these concerns into the application engenders significant challenges because relevant information is hidden in a continuous stream of events. Moreover, the implementation of the detection process in imperative programming languages is excessively difficult.

In this thesis we present novel software engineering abstractions to describe multimodal interaction patterns. Our approach consists of two major efforts: a programming language and a compatible runtime platform with an extensible architecture. The first effort consists of a domain-specific language, called Midas, which allows developers to express their multimodal tasks in a declarative manner. A declarative programming style allows the programmer to think about *what* the fundamental conditions are, instead of analysing *how* to process input events one by one, as would be necessary with an imperative language. Midas uses declarative rules to express multimodal interaction patterns. These conditions rely on the existence and the spatio-temporal relation of input events that were obtained from various input modalities. Midas provides adequate software engineering abstractions to help developers express these conditions in a modular and composable manner.

Midas programs are interpreted by Mudra, an efficient multimodal interaction architecture and processing engine. Mudra is centred on a global information storage, called the fact base, which is populated by multimodal input events from various devices. As these events arrive in a continuous manner, rules and other processes actively react to changes in the fact base. In order to do this efficiently, Mudra progressively filters and combines facts in order to derive a conclusion. Our high-level Midas programming language and its efficient Mudra runtime platform allows developers to fuse information across the data-level, feature-level and decision-level.

We have successfully deployed our solution in the real world, including live programming sessions and live music performances. Using the software engineering abstractions presented in this dissertation, we foresee the rapid prototyping of a whole range of novel natural user interfaces in a modular and composable manner.