Chapter 10
Lightweight Editing of Distributed Ubiquitous Environments: The CollaborationBus Aqua Editor

Maximilian Schirmer  
Bauhaus-University Weimar, Germany

Tom Gross  
University of Bamberg, Germany

ABSTRACT
Cooperative ubiquitous environments support user interaction and cooperative work by adapting to the prevalent situation of the present users. They are typically complex and have many environment components—interconnected devices and software modules—that realise new interaction techniques and facilitate collaboration. Despite this complexity, users need to be able to easily adapt their environments to the respective needs of the workgroups. In this paper, the authors present the CollaborationBus Aqua editor, a sophisticated, yet lightweight editor for configuring ubiquitous environments in groups. The CollaborationBus Aqua editor simplifies the configuration and offers advanced concepts for sharing and browsing configurations among users.

INTRODUCTION
Cooperative ubiquitous environments reach beyond single-user interaction and facilitate cooperation and collaboration among their users. They leverage interaction between users, artefacts, and devices, with the goal of softening or even eliminating the barrier between local and remote participants. For instance, a conference room can capture the positions of present persons and their actions, and then adapt the computer and projector configuration, the lighting, and the window shutters; and it could store these settings to support easy later resumption of a meeting.

The configuration of a cooperative ubiquitous environment describes the settings of the environ-
Lightweight Editing of Distributed Ubiquitous Environments

Lightweight Editing of Distributed Ubiquitous Environments

tment’s components, as well as the degree and shape of the individual interaction between the components. Typically, the task of configuring an environment is realised by programmers or administrators, because it requires great insight into the underlying infrastructure and system architecture, and adequate programming skills. For instance, the rules for the adaptation behaviour of the above conference room are rather difficult to configure.

The configurations should cover the needs of the end-users and their workgroups. However, despite the progress in base technologies such as data acquisition, processing, and machine learning, creating and adapting configurations is still a complex process. In order to facilitate this process, users need empowerment for end-user configuration.

In this paper we present CollaborationBus Aqua—a sophisticated, yet light-weight editor for cooperative ubiquitous environments that supports elegant capturing and storing of data from the physical as well as electronic world, visual composition of configurations, and sharing and browsing of configurations among groups of configuration authors. In the next sections we discuss related work. We then present the concept and implementation of CollaborationBus Aqua and report on its user interaction. We exemplify the user interaction in a scenario. Finally, we conclude the paper.

END-USER EDITORS FOR UBIQUITOUS ENVIRONMENTS

There are several end-user editors for editing and managing configurations of ubiquitous environments. They provide inspiring concepts with respect to their enabling middleware (e.g., eGadgets), their scheme of the configurations (e.g., iCAP), and easy user interaction (e.g., Jigsaw). As a limiting factor they mostly focus on individual end-users editing configurations of single-user settings.

In eGadgets (Mavrommati et al., 2004) a Gadgetware Architectural Style (GAS) framework for interconnecting reusable components in the form of devices, and a GAS editor for building custom compositions were developed. While an enabling middleware manages and controls all components within the framework, the editor hides complexity from users. The editor retains insight to the dataflow to avoid behaving like a black box for users. By means of connecting the components’ inputs and outputs, users generate a range of scenarios consisting of home appliances that have been adapted to be accessible through the GAS platform. The GAS framework models individual components following a plug-synapse model, where each component offers a set of abilities and requests services from other components. Devices in the physical world are represented as plugs. When different plugs are instantiated and connected, they form synapses. This model abstracts and represents compatible data types and data flows, and thus effectively helps users understand which components can be interconnected. In contrast to the eGadgets editor, CollaborationBus Aqua focuses on a cooperative composing process for ubiquitous computing environments, and offers a sharing and browsing mechanism with synergy notifications.

Another related editor is the iCAP (Lim & Dey, 2009; Sohn & Dey, 2003) editor that allows users to prototype applications and scenarios for context-aware environments. Following a pen-based interaction technique, the system’s components (input and output devices) may be interconnected to form a conditional rule-based construct in a user-friendly way. The iCAP editor allows users to draw their own sketches, which are used to represent the underlying devices within the editor environment. These sketches help to generate a deeper understanding of the constructed prototype and the interrelations between devices.
Related Content

A Grid-Based Hole Detection Scheme in WSNs
[www.igi-global.com/article/grid-based-hole-detection-scheme/67558?camid=4v1a](www.igi-global.com/article/grid-based-hole-detection-scheme/67558?camid=4v1a)

Computing Traffic Information in the Cloud
[www.igi-global.com/article/computing-traffic-information-in-the-cloud/114711?camid=4v1a](www.igi-global.com/article/computing-traffic-information-in-the-cloud/114711?camid=4v1a)

Maintaining Replicated Recovery Log for RESTful Services
[www.igi-global.com/article/maintaining-replicated-recovery-log-for-restful-services/141298?camid=4v1a](www.igi-global.com/article/maintaining-replicated-recovery-log-for-restful-services/141298?camid=4v1a)

The Red Storm Architecture and Early Experiences with Multi-Core Processors
[www.igi-global.com/chapter/red-storm-architecture-early-experiences/64449?camid=4v1a](www.igi-global.com/chapter/red-storm-architecture-early-experiences/64449?camid=4v1a)