Chapter 17
A Generic Context Interpreter for Pervasive Context–Aware Systems

Been-Chian Chien
National University of Tainan, Taiwan

Shiang-Yi He
National University of Tainan, Taiwan

ABSTRACT
Developing pervasive context-aware systems to construct smart space applications has attracted much attention from researchers in recent decades. Although many different kinds of context-aware computing paradigms were built of late years, it is still a challenge for researchers to extend an existing system to different application domains and interoperate with other service systems due to heterogeneity among systems. This paper proposes a generic context interpreter to overcome the dependency between context and hardware devices. The proposed generic context interpreter contains two modules: the context interpreter generator and the generic interpreter. The context interpreter generator imports sensor data from sensor devices as an XML schema and produces interpretation scripts instead of interpretation widgets. The generic interpreter generates the semantic context for context-aware applications. A context editor is also designed by employing schema matching algorithms for supporting context mapping between devices and context model.

INTRODUCTION
With the rapid growth of wireless sensors and mobile devices, the research of pervasive computing is becoming important and popular in recent years. The vision of ubiquitous computing (Weiser, 1993) or pervasive computing (Weiser, 1991) is to integrate hardware, network systems, and information technologies to provide appropriate service for our lives in a vanishing way. A context-aware system is a pervasive computing environment in which users’ preference services can be detected by making use of context including location, time, date, nearby devices and other

DOI: 10.4018/978-1-4666-2785-7.ch017
environmental activities to adapt users’ operations and behavior (Chen & Kotz, 2000). All kinds of context-aware architectures and frameworks have been designed and employed for a wide spectrum of applications (Baldauf, Dustdar, & Rosenberg, 2007). Since most of the systems focus on their specific application domains; the current context-aware systems are heterogeneous in all aspects, such as hardware, mobile resources, operating systems, application software, and platforms. The serious heterogeneous characteristics of context-aware computing are especially important and become significant drawbacks while developing or integrating context-aware applications in pervasive computing environments.

The concept of context independence was revealed in CADBA architecture (Chien, Tsai, & Hsueh, 2009; Chien et al., 2010). Two types of context independence, the physical context independence and the logical context independence, are classified in the article. The physical context independence is to prevent misinterpreting raw data from sensors with various specification standards; whereas the logical context independence is to allow context to be understood and applied by applications. As a result of context independence, cross-domain service applications will be able to be integrated into a unified context-aware system regardless of the heterogeneity in pervasive environment.

An OSGi-based service platform (Gu, Pung, & Zhang, 2004) based on Java VM is one of the practical solutions for accomplishing logical context independence. In this paper, a generic context interpreter is proposed to overcome the physical context dependence problem between context and sensor devices in the framework of context-aware computing. The context generic interpreter is composed of two modules: the context interpreter generator and the generic interpreter. First, the context interpreter generator imports sensor data from sensor devices as an XML schema and produces interpretation scripts instead of interpretation widgets. Then, the generic interpreter generates the semantic context for context-aware applications. An interface tool, the context editor, is also designed by employing automatic XML schema matching schemes for supporting smart context mapping between devices and context model.

The remainder of this paper is organized as follows. First we introduce the foundation and summary of context-aware architectures. The architecture of proposed generic context interpreter is presented next. The detailed design of system components is described. Then, the performance evaluation of context mapping for proposed generic context interpreter is demonstrated and discussed. Finally we conclude the work and express future work.

PRELIMINARIES

The term context-aware first appeared in 1994 mentioned by Schilit and Theimer (1994). Since then, various context-aware computing architectures were proposed: The Context Toolkit (Dey & Abowd, 1999; Dey, Abowd, & Salber, 2001) provided context interpretation using widgets and a set of object-oriented APIs to offer the creation of service components. The Hydrogen (Hofer et al., 2002) is a framework based on layered architecture in which contains the adaptor layer, the management layer and the application layer. The Gaia project (Roman et al., 2002) is a middle-ware based architecture; the system consists of Gaia kernel and application framework to support the development and execution of mobile applications. Another middle-ware system, SOCAM (Gu, Pung, & Zhang, 2004) uses a central server called context interpreter to obtain context data for building and prototyping of context-aware services. The CORTEX system (Biegel & Cahill, 2004) is also a middle-ware structure based on sentient object model which supports context-aware services in an ad-hoc mobile environment.

The above context-aware system architectures generally follow the framework presented in