Chapter 14

Using Context Awareness for Self-Management in Pervasive Service Middleware

Weishan Zhang
China University of Petroleum, P.R. China

Klaus Marius Hansen
University of Copenhagen, Denmark

ABSTRACT

Context-awareness is an important feature in Ambient Intelligence environments including in pervasive middleware. In addition, there is a growing trend and demand on self-management capabilities for a pervasive middleware in order to provide high-level dependability for services. In this chapter, we propose to make use of context-awareness features to facilitate self-management. To achieve self-management, dynamic contexts for example device and service statuses, are critical to take self-management actions. Therefore, we consider dynamic contexts in context modeling, specifically as a set of OWL/SWRL ontologies, called the Self-Management for Pervasive Services (SeMaPS) ontologies. Self-management rules can be developed based on the SeMaPS ontologies to achieve self-management goals. Our approach is demonstrated within the LinkSmart pervasive middleware. Finally, our experiments with performance, extensibility, and scalability in the context of LinkSmart show that the SeMaPS-based self-management approach is effective.

INTRODUCTION TO CONTEXT AWARENESS MIDDLEWARE

Pervasive computing (Henricksen & Indulska, 2006) is becoming more of a reality in everyday life. As a consequence, we face the question of how to handle the diversity of related computing facilities, ranging from powerful computers to small mobile phones and sensors, and how to tackle the diversity of network technologies. It is becoming
urgent that we can develop and deploy various personalized pervasive service applications efficiently. Pervasive middleware is an underlying technology to address these challenges.

Context awareness (Satyanarayanan, 2001) is one of the key features in pervasive middleware and it is used to achieve openness and dynamism of pervasive systems, to promote knowledge sharing between pervasive systems, and to provide secured and personalized services for end users. These all depend on the knowledge on when and where a service can happen, what triggers a service provision and how to provide a service to whom. That is, to be aware of the current context and to take actions based on this context. This awareness of contexts depends heavily on the modeling approach adopted in the context-awareness implementation as context models decide their potentials to provide reasoning capabilities for achieving the awareness of contexts.

It has been discussed that Web Ontology Language (OWL) ontologies (OWL, 2009) have stronger context modeling capabilities than OWL’s predecessor DAML+OIL and other counter-parts such as XML, key-value pairs, object-oriented models, and RDF (Strang & Linnhoff, 2004; Zhang & Hansen, 2008a). The main reason is that OWL ontologies are using the Open World Assumption (OWA), and openness is the nature of pervasive computing system, therefore OWL ontologies can provide better reasoning potentials and allows applications to intelligently react on a range of different contexts changes (Strang & Linnhoff, 2004; Zhang & Hansen, 2008a). In practice a choice of context modeling approach also depends on whether an approach is easy to use and efficient for resource-constrained devices, and therefore a hybrid of different context modeling approaches can be used, e.g., in the way we are doing in the Hydra EU project (Zhang & Hansen, 2008a), where a combination of key-value pair and OWL ontologies are used for context modeling as will be detailed later. The LinkSmart middleware is a product of the Hydra project, and is a pervasive service middleware for networked and embedded devices based on service oriented architecture using pervasive web services.

The rest of the chapter is organized as follows: In “Middleware for Self Management in AmI,” we will discuss the importance of context-awareness for self-management, and will survey current self-management pervasive middleware to compare their features. “Combining Context Awareness and Self Management in Pervasive Service Middleware” presents the design of the SeMaPS ontologies and our hybrid context-awareness framework. We next discuss self-management rule specification in “Self-Management Specification with the SeMaPS Ontologies using SWRL Rules.” Following this, we show the LinkSmart self-management architecture, design and implementation, based on the SeMaPS ontologies in “Architecture and Implementation of Self-Management in LinkSmart.” We demonstrate our approach with an example scenario in “A Self-Management Scenario with the LinkSmart Middleware,” together with evaluations and discussions. Finally, we conclude the chapter.

**MiddleWare For Self Management In AmI**

Recently, there has been a growing demand for high dependability of pervasive systems. When pervasive middleware is concerned, its manageability depends on whether it has some degree of self-management, as also is the case for Ambient Intelligence (AmI) (Kephart & Chess, 1996).

**Self-Management in Pervasive Middleware**

Self-management is an enabler of dependability, potentially leading to higher quality of pervasive systems. Self-management (Kephart & Chess, 1996) features constitute a broad list, including self-configuration, self-adaptation, self-optimi-