Chapter 1
A Middleware Architecture for Ambient Adaptive Systems

C. Goumopoulos

Abstract Ambient adaptive systems have to use mechanisms to regulate themselves and change their structure in order to operate efficiently within dynamic ubiquitous computing environments. First of all we outline a survey on existing middleware solutions for building ambient adaptive systems. After, discussing the limitations of the existing approaches, we present our propositions for a middleware architecture to support dynamic adaptation within ambient environments. Our approach is based on the Service-Oriented Architecture (SOA) paradigm which can be considered as an evolution of the component-based design paradigm. The aim is to use component interfaces for the identification and automated connection of components acting as service providers/consumers. The proposed middleware provides a solution that supports the adaptation of applications at the structural level, where the structure of the application can change through dynamic service composition. We call this adaptation ‘polymorphism’ in analogy with the synonymous term found in the object-oriented programming paradigm. Besides SOA, we use a set of intelligent agents to support adaptive workflow management and task realization based on a dynamically composed ontology of the properties, services and state of the environment resources. An experimental prototype is provided in order to test the middleware developed.

1.1 Introduction

Intelligent environments (IE), like smart homes, offices and public spaces, are featured with a large number of devices and services that help users in performing efficiently various kinds of tasks. Combining existing services in pervasive computing environments to create new distributed applications can be facilitated by middleware architectures, but this should accommodate special design considerations, including...
context awareness, adaptation management, device heterogeneity, and user empowerment [6].

Traditional middleware, such as Remote Procedure Calls [4], OMG CORBA [11], Java Remote Method Invocation (RMI) [44] and Microsoft Distributed Component Object Model (DCOM) [20] facilitate the development of distributed applications and help to resolve problems such as tackling the complexity of programming inter-process communication and the need to support services across heterogeneous platforms. However, traditional middleware is limited in its ability to support adaptation.

Ambient adaptive systems which are a special category of distributed systems operate in a dynamic environment. The dynamicity of the environment may relate with evolving user requirements and varying execution context due to the diversity of available devices, user preferences and services. Consequently there is a need for both applications and infrastructure to be designed for change. The evolution of user requirements calls for system evolution. The dynamic execution environment calls for dynamic adaptation. In order to allow evolution, the internal structure of the system must be made open in order to support proactive and reactive system reconfiguration.

In this work, we present firstly a survey of the state-of-the-art on existing middleware solutions for building adaptive ambient systems. After, discussing the limitations of the existing approaches, we present our propositions for middleware architecture to support dynamic adaptation within ambient environments. Our approach uses the service-oriented architecture paradigm coupled with agents and ontologies. The aim is to use component interfaces for the identification and automated connection of components acting as service providers/consumers. The proposed middleware provides a solution that supports the adaptation of applications at the structural level, where the structure of the application can change through dynamic service binding. Behavioural adaptation, not examined here, is also possible when the application logic is changed as a result of learning. An experimental prototype is provided in order to test the middleware developed.

1.2 Related Work

Three key paradigms that can be used to build adaptive systems are computational reflection, Aspect-Oriented Programming (AOP) and service oriented architectures. Researchers have also explored the possibility to combine different paradigms such as AOP and reflection in middleware systems to increase support for the development of dynamic distributed systems [19]. In the following we examine how each one of these paradigms can support the development of adaptive systems.