A TAO-based Adaptive Middleware for Pervasive Computing

Yanhui Guo  
School of Information and Technology  
Shandong women’s University, SDWU  
Ji’nan, China  
guoyanhui03@163.com

Fuqiang yang  
School of Information and Technology  
Shandong women’s University, SDWU  
Ji’nan, China

Xuechen Zhao  
School of Information and Technology  
Shandong women’s University, SDWU  
Ji’nan, China

Hong Zhang  
School of Information and Technology  
Shandong women’s University, SDWU  
Ji’nan, China

Abstract—Pervasive computing environments are open, dynamic and heterogeneous. For the above, this paper proposes an adaptive middleware named PA middleware. PA middleware is service-oriented, context adaptive, and supported by QoS. The architecture of PA middleware is based on TAO which is a standards-based, CORBA middleware framework. This paper presents a model for context awareness to allow the adaption, and gives a method of context description for context resource. It also proposes an adaptive strategy which uses a genetic algorithm for optimization.

Keywords—pervasive computing; middleware; components; context awareness; adaptive

I. INTRODUCTION

Pervasive computing needs to process information everywhere, for everyone, all the time. In the way, systems in pervasive computing should perceive and adapt to the environments[1]. They have to understand the changing context in which they are. However, it is difficult to develop application adapting changeable environment[2]. To apply middleware technology is a good solution, which shield heterogeneity by the "black-box" reuse to simplify the development and management of distributed applications. In pervasive computing, a variety of resources are heterogeneous, dynamic and open. For these requirements, the middleware needs to be flexible to dynamically adjust their behavior according to the context change to adapt to the different resources and quality of service.

Traditional middleware does not support the architecture reconfiguration at runtime, named non-adaptive middleware, such as TinyDB[3], Mires[4]. For pervasive computing, the autonomous and adaptive middleware is suitable to address the changing environments. Some studies focus on awareness[5], including resource discovery, context modeling. Some papers mainly study adaptive strategy[6] [7]. Nevertheless, most of pervasive computing middlewares have lack of QoS(Quality of Service). Some middlewares are more complete, such as Gaia[8], PCOM[9]. But, these only given their solution to one problem. By analyzing these projects, component-based and service-oriented middleware technologies in pervasive computing are commonly used and effective. This paper introduces the QoS mechanisms as part of the context awareness, which better meet the adaptability of the middleware. Another contribution of this paper is to propose a architecture model. Finally, this work implemented a prototype system in video streaming system.

II. THE PRINCIPLES OF DESIGN

In pervasive computing, resources are limited and changing. Context changes and user requirements need the application to have the ability of self-adaption and autonomy. So, QoS is widely used in applications, but rarely used in middleware. This increases the complexity of the application, making it difficult to maintain and manage. It is a good solution to take QoS into middlewares. Unlike traditional middleware, we separate functional attributes and non-functional properties. QoS mechanisms are placed in the middleware layer, to ensure the reconfiguration in the runtime. Meanwhile, the component technology and service combination are widely used in this middleware model.

This middleware model is an open, adaptive, component-based, service-oriented architecture. It has QoS management component, and be able to support different levels of QoS requirements. There are four requirements in the middleware architecture in order to adapt to the development and application of pervasive computing:

- Component-based services: to simplify the development and maintenance of middleware.
- Independence: include the various stages of the development process (design, deployment, running).
- Openness: the components and applications can be deployed or replaced according to the requirement.
- Context awareness: the middleware should be aware of the changes in the context and make the appropriate processing.

In addition, we design the context model to adapt to the environment of pervasive computing. The model uses XML to describe. QoS informations are used as context attributes.
III. THE ARCHITECTURE OF PAMIDDLEWARE

A. the Idea of Design

This adaptive middleware we designed is called PAmiddleware, which provide services by the creation and combination of components. The logic type is used, whose configuration parameters and QoS requirements are specified by the users. As an abstract type, it has no specific implementation. PAmiddleware is responsible for positioning and instantiating the logical type, and also responsible for passing parameters between them.

In the PAmiddleware, component service is divided into three steps (design, deploy, run phases). In the design, the type of component service is used. In the deployment, component service is described as a plan, which way is used to achieve the service type. In implementation of services, it implements the service, in accordance with the planning of the deployment phase. Application developers only need to specify the type of component service. the other is completed by the middleware.

Service, as a function unit, is composed by a component or more components. In this paper, we call it component service, and call component as service component. Component service has the following characteristics:

- Component service has two forms: atomic component service, or composite component service.
- Atomic component service is independent, and it is implemented by atomic component.
- Composite component service is composed by atomic components. There are not only functional dependencies among these atomic component service, but also QOS non-functional dependencies.

B. the Architecture of TAO-based PAmiddleware

The following section describes the architecture of our proposed middleware. It also is an improved version of a middleware model [10] that we have done before.

TAO is a freely available, open-source, and standards-compliant real-time middleware. It attempts to provide efficient, predictable, and scalable quality of service (QoS). However, it does not support the context management and adaptive mechanisms of management. This cannot meet the requirements of pervasive computing. This paper extends the TAO from three aspects (fig.1).

Service mapping mechanism: responsible for combination between the application and TAO. It maps from the service request to the service implementation.

Adaptive mechanisms: responsible for the dynamic mediation of the application during operation. Including the configuration of the service request, as well as during the operation of the reconfiguration. Depending on the context information, it can develop various service contracts.

Context Manager: responsible for management on context information, including acquisition, the integration, as well as classification and storage. We can create a different latch of context management so that we can flexibly manage context.

C. the Architecture of Extension Mechanism

In this part, we mainly introduce the extension of TAO. The architecture of the extension is shown as fig.2. The extension is composed of service mapping mechanism, adaptive mechanism and context manager. Context manager provides the same interfaces to the adaptive mechanism. The adaptive mechanism monitor the changing context, make a plan, and then take the plan to the service mapping mechanism. The service mapping mechanism is as a container, managing components for encapsulation, classification and linking.

1) Context Manager

Context awareness is one of the key technologies in pervasive computing. In this paper, resources and context manager are introduced in this paper, to accomplish the integration between physical space and information space, which shield the complexity. According to different types of resource context information, it is divided into different types. Each type has its own context and resources bolt, to
use in your applications. Collection, integration, and classified storage of the context are achieved by gatherers, translator, and memory. You can see our paper [11] what we have done for more details.

a) Resource context manager: mainly for management of the resource context (reference on OMG classifications and definitions of resources), including the storage and extraction of resource information, implementation of resource interface, and resources latch and management on creation and deletion.

b) Environment context management: management of the context information, including collection, extraction, and integration for information, such as user’s location, temperature, humidity, etc. At the same time, context manager should manage on link among different sensors, as well as on context of the implementation of the interface. Modeling and storage of context information also are realized by this part.

In order to manage and understand the context information of pervasive computing, the context can be divided into three levels, the context node layer, the logical abstraction layer, and the user layer.

a) The context node layer: This is the bottom of pervasive resource management, involving various types of pervasive resource node physical, such as a single sensor node equipment, a variety of software services, this layer mainly refers to the management and control. Each resource node has attributes, such as the access capability of the access terminal, the display capability or the service function provided by the software service entity. Its main function is to collect the original data, and to quantify the original data.

b) The logical abstraction layer: virtual resource layer, This is the middle level of pervasive resource management, the layer of the layer is the main context of the semantic representation of context information is analyzed, abstracted can be provided to the user layer using the context information.

c) User level: This is the highest level of pervasive resource management, involving the establishment of user service quality model, user preferences and user context and other functions. In this paper, the context information is classified according to the different application fields.

2) Adaptive mechanism

Shown as fig.2, adaptive mechanism is composed of five parts. Adaptive mechanism as a core on pervasive computing, should regulate by itself according to the context, to ensure the user's QoS request. Firstly, when available resources increase, the service should improve the level. Secondly, when available resources decrease, on reduce the level of service of system. It includes the following sections.

a) Context monitor: get and monitor the changes on context information, completion of the context acquisition in the service configuration, responsible for monitoring.

b) Service planner: plan the service and get a reasonable service contract, according to the user’s requirement.

c) Configuration Manager: accomplish service configuration, according to service contract which is generated by service planner.

d) Reconfiguration manager: adaptive reconfiguration on mediation process.

e) Comparison Analyzer: in reconfiguration stage, get a new contract, analyze similarities and differences between old service contract and the new one, get a improved component service.

3) Service mapping mechanism

Service mapping mechanism is the link between on extension mechanism and TAO. It is responsible for the classification and registration of component service, and TAO component package. It mainly includes the following sections:

a) Component service encapsulation mechanism: encapsulate a variety of component service as a unified format to use.

b) Classification mechanism: reclassify the TAO’s component depending on application behavior, in order to fit on adaptive mechanism.

c) Mapping mechanism: responsible for conversion from component type to component implementation.

Here, we need to understand the concept of component-based service. A service is a functional unit provided by the service provider to its customers, which is composed of one or more components. Component-based services have the following features:

a) There are two types of component services: either an atomic component service or a composite component service.

b) The function of atomic component service is independent, and its function is used by accessing the interface of atomic service component.

c) The composite component service is provided by the combination of atomic service components. There are not only functional dependencies among these atomic component services, but also QOS non-functional dependencies.

IV. CONTEXT DESCRIPTION

A wide variety of contexts increase the difficulty of the use of the context. A unified context description method is necessary, which is also beneficial to the classification and storage of context information.

In this paper, the context information is described as a three tuple CP= (SN, PR, QoS), where SN is the only identification of context; PR={P1, P2, ... Pn} is composed of a series of Pi = (A, T, V), where A is the attribute name, T is the attribute type, V is the attribute value; QoS = {Q1, Q2, ... Qm} is a series of Qi = [T, Max, Min], where T represents the aspect of QoS concern, Max represents the maximum resource provided, Min represents the minimum value provided by the resource.

V. PROCESS OF THE ADAPTATION

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and
import your prepared text file. You are now ready to style your paper.

Adaptive mechanism is the core of adaptive middleware, which indicates the manner in the configuration. Services are classified in on adaptive process, and play a crucial role. The service planner plays an important role in this process. It generates service contract in accordance with user’s requirement and context. The planner also links with service configurator and service reconfigurer. Together, they complete the adaptation. The entire workflow is as follows.

![Workflow Diagram](image)

Figure 3. the Process of the Adaptation

a) User or application sends a request to the service planner, and requests the appropriate service.

b) According to the type of request, service planner gets context through the monitor. It also makes context monitor to oversee the change.

c) Service planner, according to the context information, generates service contract.

d) Service planner notifies service configurator to configure the service in accordance with the service contract.

e) In the operating mechanism, the component is instantiated.

f) If the context monitor oversees the context change, it notify service planner, which should compare the configuration information in the conflict in the current context and service contracts.

g) Service planner notifies service reconfigurer to complete the service reconfiguration.

h) Compare the old service contract and the new one.

In this section, we proposed an adaptive strategy for pervasive computing middleware. We put forward the concept of service contract. We use genetic algorithm to optimize the service composition. You can see our paper [17] for more details.

VI. SUMMARY

In this paper, we take the process of soft development into account. An improved TAO middleware platform was proposed to be fit for pervasive computing. The next important job is to improve the genetic algorithms in the configuration and reconfiguration mechanisms or adopt other intelligent optimization algorithms. Meanwhile, it is necessary to adopt ontology for modeling context information.

ACKNOWLEDGMENT

This research is supported by the Shandong Provincial Department of Education Fund (Grant No. J14LN54) and the science and technology department of Shandong Province (Grant No. 2013YD01041).

REFERENCES


