An Adaptive Middleware Infrastructure for Mobile Computing

Ronnie Cheung
Hong Kong Polytechnic University
Hong Kong, China
csronnie@comp.polyu.edu.hk

ABSTRACT

In a mobile environment where mobile applications suffer from the limitation and variation of system resources availability, it is desirable for the applications to adapt their behaviors to resource limitations and variations. It is also necessary to exploit optimal application performance. However, adaptation mechanisms by mobile applications usually suffers from the problem of unfairness to other applications, in contrast, adaptation by the operation system focuses more on the overall system performance, while neglecting the needs of individual applications. Hence, the adaptation task is best coordinated by a middleware that is able to cater for individual application's need on a fair ground, while maintaining optimal system performance. This is achieved by a context-aware mobile middleware that sits in between the mobile application and the operating environment.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures

General Terms

Design, Standardization

Keywords

Mobile Environments, Adaptation, Middleware Infrastructure

1. INTRODUCTION

Mobile computing extends the horizons of conventional computing model to a ubiquitous computing environment that serves users at anytime, anywhere. In order for mobile applications to operate efficiently in a mobile environment, mobile systems have to perform two tasks: (1) it needs to sense and reason about the current context of the operating environment; (2) it has to react effectively to the current context by adapting its service to optimize system performance. In order to achieve context awareness for mobile computing, it is redundant and inefficient for mobile applications to maintain the required contexts independently. It is also not appropriate for application developers to check the statuses of all related contexts exactly. Instead, a more generalized and abstracted description of the current context would be adequate for a mobile application to adapt effectively. This extra layer of abstracted contextual description can be provided by a middleware approach. Another problem is how mobile applications utilize the underlying system service to adapt to the current context. This is a challenging task for application developers since mobile applications have to implement their own adaptation action down to the system level. For example, if a mobile application requires an adjustable, partial reliable, rate control transport protocol to adapt to the dynamics of the wireless network, it has to implement its own protocol on top of UDP. Again, a mobile middleware environment provides the generic system services which are programmable for adapting to the current context.

2. SYSTEMS REQUIREMENTS

When we shift the underlying adaptation operations from mobile applications to a mobile middleware, several issues arise. The first issue is that if a mobile application needs information on the current context (either primitive detector based or high level abstracted) for internal logic adaptation, how does the middleware convey such information? It is desirable for the middleware to expose a set of generic interface for context subscription and query. The second issue is that the mobile application needs to specify the requirements for middleware services, such as the type of service (ToS), and the quality of service (QoS). Again, a comprehensive set of abstracted interface will need to be identified for negotiating the required middleware services. The third issue is how the mobile middleware knows under what situation should it carry out adaptation? What are the combinations and parameters for middleware services that should be chosen to best adapt to the current context [1,2], while fulfilling the requirements of the mobile application? To address these issues, we have identified several requirements that a context-aware mobile middleware infrastructure should support:

- Collection of contextual information from detectors.
- Composition of high level context abstraction from low level detector context.
- Context information delivery in terms of subscription-notification approach and query approach.
- Middleware services that are programmable for adapting to the changing context.
- ToS and QoS specification for mobile applications to customize the services provided by the middleware.
- Automated reasoning and adaptation of changing context, which controls the middleware services to optimize and fully utilize the available information and resources.

3. ADAPTIVE MIDDLEWARE INFRASTRUCTURE

We have developed a complete infrastructure that allows generic mobile applications to exercise context awareness. The significance of the project is the integration of all the features, and the synergy provided by a well-established, unified baseline architecture that promotes the development of context-aware computing. The mobile middleware infrastructure that integrates context collection, context composition, context delivery, adaptive middleware services, ToS and QoS. The advantage of introducing adaptation mechanisms in the middleware layer is that the middleware has both the knowledge of mobile application

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

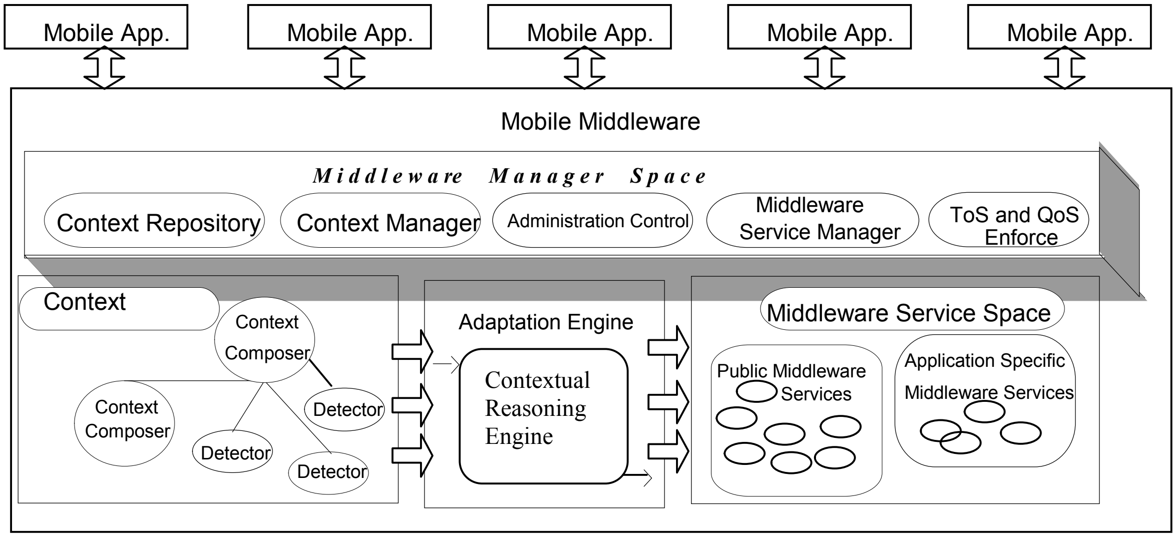

Figure 1. An Adaptive Middleware Infrastructure

, operating system, and the operating environment. Resources are allocated fairly among mobile applications, at the same time fulfilling individual application's requirement. As shown in Figure 1, there are four major modules in the system:

The first module is the *Context Space*, which contains *Context Detectors* objects and *Context Composers* that reason and compose low level contexts into higher level representations. Examples of context detectors include wrappers for OS events, which could also be directly communicating with the device drivers. At a higher level, context detectors could also be detecting an application's communication and computation activity. The context composer gather low level information, marshalling the dynamic and uncertainty of the mobile environment, and describe the current context in a more generic and coarse form. For example, a context composer could be monitoring all the network related detectors, and determine the quality of network connectivity – good, average, poor and no connectivity. High level contextual information is also useful for adaptation with mobile applications.

The second module is the *Middleware Service Space*. This is the execution environment for both *Public Middleware Services* (one set of services that are shared among all mobile application) and Application Specific Middleware Services (each mobile application has its own set of services). These adaptive middleware services are categorized by their service nature into either public or application specific, e.g. a Web caching and prefetching middleware service is categorized as public service; a media transcoding service is categorized as application specific.

The third module is the *Adaptation Engine*, which adjusts the middleware services according to the current context and the requirement of mobile application specific ToS and QoS. In order to control the middleware services, the adaptation engine is aware of the programmable properties of the middleware service. To adjust quantitative controllable parameters for the middleware services, the adaptation engine calls the corresponding adjustment functions. The middleware services are responsible for exporting the required interface for adaptation control, and to adjust the internal logics to follow the adjustments specified by the adaptation engine.

The fourth module is the *Middleware Manager Space*. The middleware manager space contains five system managers, which coordinates all operations within the mobile middleware :

The *Administration Control* component manages the admission of mobile applications that subscribe the services of the middleware. The duties include : authentication, ToS and QoS negotiation, service subscription and un-subscription.

The *Context Manager* controls the runtime environment for the context objects, including the low level context detectors and high level context composers.

The *Context Repository* stores the records on all contextual information, such that queries on context history are possible. Moreover, the past records can be used for predicting the future trend of the contexts.

The *Middleware Service Manager* coordinates with admission control. It also allocates and controls the resources for newly subscribed services.

The *ToS and QoS Enforcer* is used for monitoring the ToS and QoS levels for each connected mobile application.

4. CONCLUSION

We have implemented a middleware infrastructure that facilitates the development of environmental context reasoning and presentation entities, adaptive system services and context-aware mobile applications.

5. ACKNOWLEDGMENTS

This project is supported by UGC CERG grant no. B-Q829.

6. REFERENCES

[1] S.N. Chuang, A.T.S. Chan, J. Cao, Ronnie Cheung, Dynamic Service Reconfiguration for Wireless Web Access in Proceedings of the Twelve International World Wide Web Conference, (May 2003), ACM Press, 58-67.

[2] S.N. Chuang, A.T.S. Chan, J. Cao, Ronnie Cheung, Actively Deployable Mobile Services for Adaptive Web Access, IEEE Internet Computing, March 2004.