EUROGRAPHICS Workshop on Multimedia (2004)
N. Correia, J. Jorge, T. Chambel, Z. Pan (Editors)

A Web Services-Based Architecture for Capability-Aware Ubiquitous Media†

Wang Haipeng, Zhou Xingshe, Duan Zongtao, Zhang Tao

School of Computer Science
Northwestern Polytechnical University
Xi'an 710072, Shaanxi, P.R.China
whp_box@hotmail.com

Abstract

Ubiquitous media aims to provide media services anytime and anywhere. To realize it, one challenge is how to provide customized and dynamic services to a variety of computing devices with different capabilities. This paper presents an architecture, which enables customized delivery of multimedia services. A concept of "capability" is used to abstract the adaptation-related attributes of computing devices. Capability concerns both the static and dynamic attributes of the computing devices, such as display resolution and remaining battery power. These two classes of attributes can be combined to provide complementary information for customized and dynamic media delivery. As a proof of concept, we have developed a prototype implementation, which is characterized by Web services-based architecture, and capability-aware feature. Our initial experiments show the effectiveness of this architecture.

Categories and Subject Descriptors (according to ACM CCS): D.4.7 [Operating Systems]: Distributed systems

1. Introduction

The explosive growth of media services and numerous computing devices presents novel requirements for ubiquitous access and usage of these services, called ubiquitous media. Ubiquitous media environment includes a variety of media services, such as audio/video playing, web browsing, and e-learning. Users may access these media services using different computing devices anytime and anywhere. For example, a video playing service has to serve different subscribers with PDA, laptop, and high performance workstation. With PDA's limited display size and low bandwidth network connection, the server has to provide video playing service with degraded quality; with high performance workstation, the server however need to upgrade the quality of service concerning the client's high capability. Therefore, a challenge in ubiquitous media is to deal with the heterogeneous clients' capabilities, and to provide suitable and adaptive media services to different clients [XWN00].

Several projects in ubiquitous media community have explored the implications of media services delivery. The $2K^Q$ [XWN00] explored the resource-aware delivery of ubiquitous multimedia. In $2K^Q$, multimedia services can be delivered by using multiple service configurations, which are chosen based on current available resources, called resource-aware configurations. However, $2K^Q$'s resource-aware feature concerns only the dynamic features of computing devices, such as current CPU usage, bandwidth, and remaining battery power, without consideration of static attributes of display size, resolution, and bit depth, which is important for portable computing devices. Darwin [CFK*98] presented a framework for composing complicated valued-added services, based on a hierarchical service brokerage architecture. Cactus [Tea] proposed a framework for supporting customizable dynamic fine-grain Quality of Service attributes related to dependability, real time, and security in distributed systems. However, both projects focus more on the distributed aspect of the service management issues, while not empha-

† This research was supported by the National High Technology Research and Development Program of China (863 Program) under contract 2003AA1Z2100

sizing the problem of different computing devices' (clients') capabilities heterogeneity. Fitzpatrick [FBC*98] proposed the concept of open binding for the implementation of the service and resource management policies for adaptive multimedia applications. However, as a meta-mechanism, open binding itself does not provide any service management policies. Jini [Wal99] is a software infrastructure to federate networked devices and software components into ubiquitous and dynamically composable distributed services. However, Jini does not emphasize the QoS issue in multimedia services. Poladian [PaPSGS04, SG03] proposed an analytical model and an efficient algorithm for the dynamic configuration of resource-aware services in ubiquitous computing environments. While our work shares some concepts with [PaPSGS04], especially the abstraction of "capability", the problem solved in our work is different. In that work, the objective is to choose among possibly thousands of configurations so as to maximize the objective function of one user. In our work, the objective is to provide a capability-aware mechanism among possibly hundreds of ubiquitous computing devices so as to support adaptive ubiquitous media services.

In this paper, we present an architecture that enables capability-aware delivery of media services, which uses the concept of "capability" to abstract the adaptiveness-related attributes of computing devices. Capability concerns both the static and dynamic attributes of the computing devices, such as display resolution , bit depth, and remaining battery power, which can combined to provide complementary information for the customized and dynamic media delivery. We have developed a prototype, which is Web services-based and capability-aware. Our experiments show the effectiveness of this architecture.

The rest of this paper is organized as follows. Section 2 describes the rationale of capability-aware ubiquitous media services, and presents our approach. Section3 describes the architecture of our prototype system. Section 4 discusses the current status of our implementation. Section 5 concludes our work.

2. Approach

The ubiquitous media environment includes numerous media services and computing devices. As the heterogeneity of computing devices and the dynamic nature of environments, it is difficult to deliver a media service to a certain computing device with desired quality of service. For example, consider a scenario, in which a video playing server provides services to a home subscriber with desktop PC and a video-conference subscriber with PDA. With the PC, a high-fidelity MPEG video streaming can be provided; but with the PDA, it is necessary to do an MPEG-to-H.263 transcoding due to PDA's limited CPU cycles, bandwidth, and battery power.

Based on current available resources, multi-fidelity media services are possible. For example, a laptop with wireless network interface would suffer frequent disconnection, so an MPEG-to-bitmap service is suitable. They are called resource-aware applications. However, most resource-aware applications concerns only partial attributes of computing devices, such as CPU cycles, bandwidth, memory, battery power, and so on. The media services still can not provide correct quality of services with these information. Consider a PDA and laptop, both of which may have same remaining CPU cycles, bandwidth, and battery power, but have different display size, if we still provide identical 352×240 services, this is obvious unsuitable.

Capability is an abstraction of computing devices' static and dynamic attributes, includes static display size, resolution, and bit depth, and dynamic CPU cycles, memory, and battery power. Capability can provide more information for making decision about ultimate quality of service. With the previous scenario, it can reduce the resolution to achieve a more reasonable quality of service.

In this paper, capabilities are used to enable fine-grained tuning of quality of service, called two-phase adapting, depicted in Figure 1. In the first phase, static capability information, such as display size, resolution, and bit depth, are used to drive coarse-grained tuning. In the second phase, such as CPU, memory, and bandwidth, dynamic resource information are used to drive fine-grained tuning. With the two-phase adapting, a resulted media service would be delivered to subscribers with the suitable quality of service.

The rationale, underlying the two-phase adapting approach, is in the computation reuse of the first-phase adapting. There are several mapping from one first-phase adapting to multiple second-phase adapting, since the dynamic nature of second phase. For example, the bandwidth and battery power of PDAs vary frequently, but its display size and resolution are relative stable. The two-phase technique can effectively improve the quality of adaptive media services in ubiquitous computing environments.

3. Architecture

The overall architecture is illustrated in Figure 2, which involves three major parts, as follows.

Capability Monitors The capability monitors (CM) are responsible for the monitoring and reporting of devices' capabilities information.

Capability Exchange Protocol The protocol is used to exchange capabilities vectors. Capabilities vector is formally expressed as:

$$C = \{c_1, c_2, \ldots, c_n, r_1, r_2, \ldots, r_n\}$$

. Where c_i denotes capability component, such as resolution, bit depth, and so on; r_i denotes resource component, such as CPU, bandwidth, and so on.

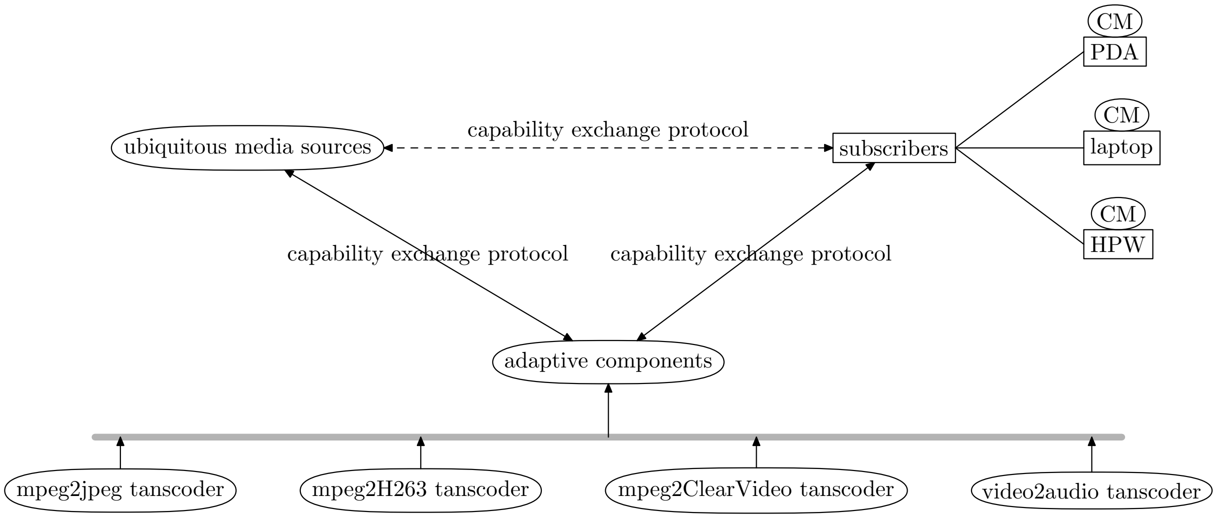


Figure 2: *Architecture*

Adaptive Components The adaptive components provide transformation functions, such as video-to-audio, MPEG-to-ClearVideo, and so on.

This architecture operates in a two-phase adapting mode, described in section 2. Firstly, the clients report their capabilities vectors to media servers, then the media servers extract c_i part, $\{c_1, c_2, \ldots, c_n\}$, and make decision how to deliver suitable quality of media services to clients by using selected adaptive subcomponents. Secondly, whenever the clients' resource availability vary, the media servers perform the second-phase adapting based on the r_i part, $\{r_1, r_2, \ldots, r_n\}$, of capabilities vectors. The latter phase usually includes a reconfiguring and rebinding processes [K*01, PaPSGS04].

4. Implementation

We have developed a prototype of this architecture, which is Web services-based, depicted in Figure 3. The Web services environment includes Apache Tomcat (version 5.0.25), Apache Axis (version 1.1) with its simple stand-alone server, and J2SE (Version 1.4.2) platform. The CMs and adaptive components are implemented as Web services. The clients include an IBM thinkpad laptop with wireless network interface, and a desktop PC with a 100M network interface. The media server is an IBM x-series-235. In addition, there are two adaptive components: a video-to-audio transcoder and an MPEG-to-ClearVideo transcoder, running on general-purpose desktop PCs respectively.

This prototype is used to execute a video playing application. The media server can serve two clients with different capabilities. It supports a wide variety of media services with certain quality of services matching with target computing devices. Whenever the laptop's wireless connection degrades, the media server gets the notification pushed by the client CM, and react with a switching to video-to-audio. Finally, the laptop gets a more suitable media service with server side's best effort.

Our long-term goals for this architecture includes a building of a private UDDI system and a wide variety of adaptive components wrapped with Web services, which will satisfy the requirements of more ubiquitous computing devices with different capabilities, and will provide an automatic searching and selecting mechanisms. We are working on a number of these challenges now.

5. Conclusions

To address the challenges of ubiquitous media, we present an architecture that enables customized delivery of multimedia services, which uses the concept of "capability" to abstract the addition-related attributes of computing devices. Capability encapsulates both the static and dynamic attributes of the computing devices, which are combined to provide complementary information for the customized and dynamic media delivery. This architecture operates in a two-phase adapting mode, which effectively improved the quality of adaptive media services in ubiquitous computing environments. We have developed a prototype, which is Web services-based and capability-aware. Our experiments show the effectiveness of this architecture.

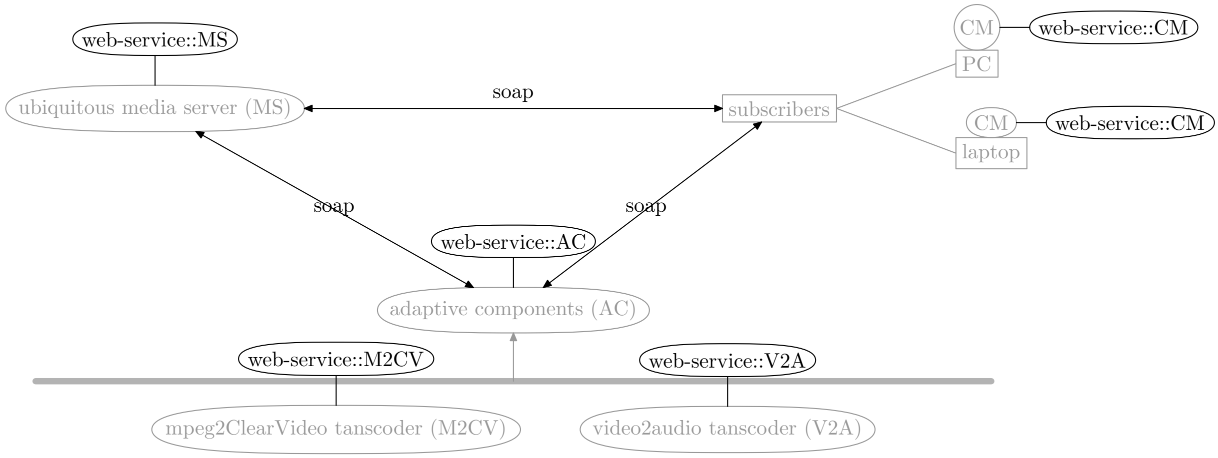
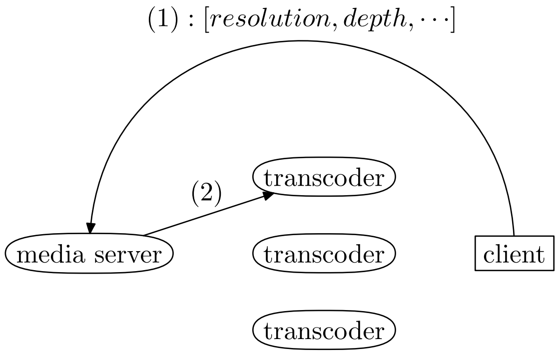
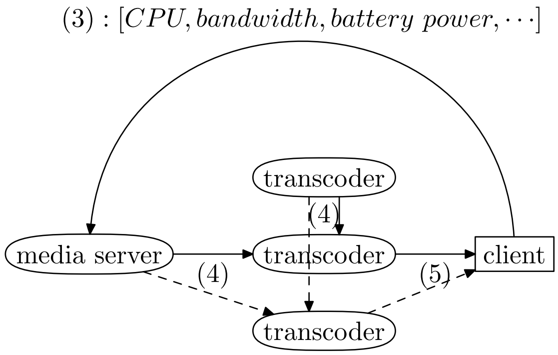


Figure 3: *Web Services-Based Implementation*



(a) First-Phase Adapting



(b) Second-Phase Adapting

Figure 1: *Two-Phase Adapting*

References

[CFK*98] CHANDRA P., FISHER A., KOSAK C., NG T. S. E., STEENKISTE P., TAKAHASHI E., ZHANG H.: Darwin: Resource management for value-added customizable network service. In *Sixth IEEE International Conference on Network Protocols* (October 1998).

[FBC*98] FITZPATRICK T., BLAIR G. S., COULSON G., DAVIES N., ROBIN P.: Supporting adaptive multimedia applications through open bindings. In *International Conference on Configurable Distributed Systems* (May 1998), pp. 128–135.

[K*01] KON F., ET AL.: Dynamic resource management and automatic configuration of distributed component systems. In *USENIX Conference on OO Technologies and Systems (COOTS)* (2001).

[PaPSGS04] POLADIAN V., AO PEDRO SOUSA J., GARLAN D., SHAW M.: Dynamic configuration of resource-aware services. In *26th International Conference on Software Engineering* (May 2004).

[SG03] SOUSA J., GARLAN D.: *The Aura Software Architecture: an Infrastructure for Ubiquitous Computing*. Tech. Rep. CMU-CS-03-183, Carnegie Mellon, 2003.

[Tea] TEAM T. C. P.: The cactus project. http://www.cs.arizona.edu/cactus.

[Wal99] WALDO J.: The jini architecture for network-centric computing. *CACM* (July 1999).

[XWN00] XU D., WICHADAKUL D., NAHRSTEDT K.:

Resource-aware configuration of ubiquitous multimedia service. In *Proceeding of IEEE International Conference on Multimedia and Expo 2000(ICME2000)* (July 2000).