The Design and Implementation of Cloud Service Based Customized Mobile Device Application System

Zhu Tong
School of Information Engineering
Wuhan University of Technology
Wuhan, China
e-mail: zhutong68@139.com

Gao Zhuo
Computer School
Beijing Information Science &Technology University
Beijing, China
e-mail: gaodigang@bistu.edu.cn

***Abstract*—In this paper, we propose a system plan for cloud service based local mobile device application customization. The mobile device acquires tailored function configuration from the cloud service and generates the plan for the application with a local configuration interpreting engine. The system enables local applications of mobile devices to be adaptive to the variation of requirements.**

Keywords- cloud services, mobile device, customized application, configuration interpreting engine

I. INTRODUCTION

Cloud computing and mobile computing has already been widely used in every regard of enterprise application and personal application. The new IT service based on cloud computing is called cloud service, with "on demand" and "pay-as-you-go" as its essential characteristics. Cloud service has become the fifth utility after water, electricity, gas and communication. Cloud service and mobile computing will create a new mainstream for the IT industry [1-3].

The cloud computing platform has already formed a mutual promotion, mutual motivation and common rapid development trend with mobile application. On the one hand, cloud service puts the computing implementation in the cloud, overcoming the memory power limitation of mobile computing; on the other hand, the ubiquity, instantaneity and diversity of mobile application provide extensive need and wide space for the development of cloud computing platform.

With the widespread use of mobile devices in enterprise application systems, new problems emerge. Improving the enterprise efficiency and enhancing customer service with the help of cloud service and mobile computing have become significant problems for enterprise application systems. Mobile applications can be categorized to browser and local applications. The former get the work done at the cloud end, and generally depend less on the type of the mobile device. These applications are more adaptable and do not require software update at the client side when the service has to change. However, they suffer from inconvenient integration with local services such as voice, SMS, GPS etc, and also have poorer user experience. On the other hand, the non-browser based local applications implement a majority of it functions at the mobile side, enjoy easy access to local services and better user experience. They can rely on the cloud service by submitting the tasks as well, but updated applications need to be re-developed when the service is changing, which incurs longer update cycles and other inconveniences, especially for those rapidly changing services. This is the background of the widely debated choice between intelligent terminal and dumb terminal. In this paper, a new system plan is proposed. The application is configured at the cloud side. Mobile devices retrieve configuration from the cloud service, and generate applications via local configuration interpreting engines.

The mobile applications belong to C/S mode, or more precisely multiple-layer C/S mode taking cloud service and configuration interpreting engine into consideration. Various technical details and applications of multiple-layer C/S mode were discussed in [4-7] from different perspectives. The security issue was investigated in [8], and the automatic update in [9]. In [10-12], the restricted terminal, mobile terminal and internet service applications were researched on.

In this paper, we first propose the architecture of the system on top of the previous results. Secondly, the triple model of cloud side configuration (functionality, local service type, URI service) and the configuration-transferring protocol are given. Finally, we describe the design and implementation of the configuration acquisition, local configuration interpretation and generation at the mobile application.

II. ARCHITECTURE OF CUSTOMIZED LOCAL APPLICATION BASED ON CLOUD SERVICE

To customize the mobile local application, the server in the cloud needs to be able to configure the client application, and the mobile device need an implementation of the configuration interpreting mechanism and a generator of the local application functionality. A protocol of configuration transmission via cloud service is also needed. The system architecture is shown in Figure 1.

Published by Atlantis Press, Paris, France.
© the authors

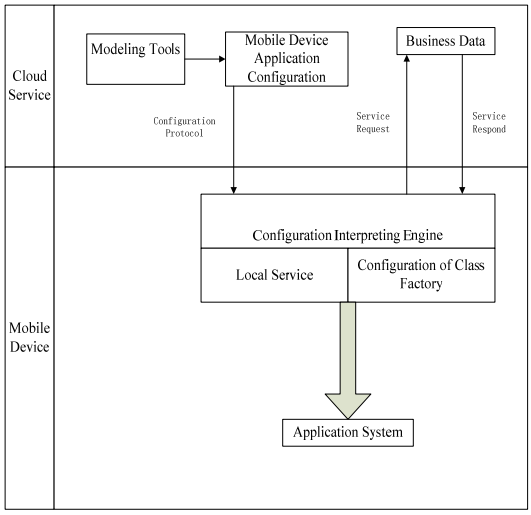

Figure 1. Architecture of cloud service customized mobile application

At the cloud side, a triple of functionality, local service type, URI service is used to represent the functionality, service type and URI service of the data in the application. When the service requirement changes, the functionality can be adjusted with modeling tools.

At the mobile device, a configuration interpreting engine is implemented to acquire the configuration and generate local application system through local service . The local services called by the configuration interpreting engine can be divided into system services of mobile terminal, such as voice, SMS, GPS, etc; local services encapsulated by the engine, such as data list presenting service used to present the data given by the cloud service in lists; and third-party services installed in the mobile terminals. Visiting of designated web pages and cloud services is also supported by the engine.

The local services called by the configuration interpreting engine are implemented with local XML configuration file in class factory pattern to guarantee the flexibility and expandability of the system.

III. CLOUD SIDE AND ITS PROTOCOL OF CUSTOMIZED LOCAL APPLICATION

At the cloud side, a triple of functionality, local service type, URI service is used to represent the functionality, service type and URI service of the data in the application. Corresponding to traditional menu style and the group navigation style, a layered structure is used to represent the relations of functions. We implement a cloud server side group navigation interface model with an exemplary two-layer navigation interface. (Note the implementation of multi-level tree structure is similar except that a metadata model is in need. We omit the implementation details of the tree structure to focus on the core ideas of customized applications.) The model is shown in Table 1.

TABLE I. TRUCTURE OF MOBILE DEVICE APPLICATION CONFIGURATION MODEL

Level 1 business code	Level 1 Business name	Level 2 Business code	Level 2 business name
Icon	Service type	URI service	

In the model, the code and name of two levels can be used to construct two-level menus or two-level group navigation interface. (The specific interface will be implemented by the local configuration engine according to the type of mobile operating system) .The Icon is given to make sure the same application has the same representation under different mobile systems. The service type is used to distinguish the service types called by the local engine, including cloud services and local services. URI service defines the cloud service call used to acquire business data.

The system model is configured by modeling tools at the cloud service side. The mobile device acquires the configuration from cloud service at first logon and maintains a local copy. At following logons, the local configuration is verified and updated to guarantee generation of newest version of the mobile device application. The XML representation of the protocol is shown in Table 2.

TABLE II. AN EXAMPLE OF THE XML REPRESENTATION OF THE CONFIGURATION PROTOCOL

```
<returnInfo>
        <isok>1</isok>
        <msg>TJRSPUSR2370</msg>
        <mnav count="16" pagesize="5">
                <page index="0" count="5" expand="true"
title="Basic Information">
                        <item showdes="1">
                                <id>IDC1</id>
                                <text>Product</text>

        <icon>/com/cntomorrow/mobile/pngs/idc/cplb.png</icon>
                                <type>contextapp</type>

        <uri>[{'package':'com.cntomorrow.mobile.framework.Page
ListDetail'}]</uri>
                <params>
                  [{'class':'com.cntomorrow.mobile.framework.f
actory.listdatasource.DefaultDetailSource',
                        'pageSize':'15','canSearch':'0','baseUri':'http:/
/ Service IP:Service Por/MobileIDC',
                        'action':'mobile.do','params':'dispatch=getBa
sicalInfo&amp;
                          accountName=MAPPING-
                          USERID&amp;userName=MAPPING-
                          USERNAME&amp;passWord=MAPPIN
                          G-PASSWORD &amp;imei=MAPPING-
                          IMEI &amp;imsi=MAPPING-IMSI
                          &amp;model=MAPPING-MODEL'}]
                </params>
                                <description>Get
Product</description>
                                <smstag/>

        <needupdate>0</needupdate>
                        </item>
                        .....
                </page>
        </mnav>
</returnInfo>
```


© the authors

The configuration protocol of mobile device application has its XML node description shown in Table 3. PAGE is for the tree structure of the functionality. ITEM and TEXT are tags for functionality. TYPE indicates the local service to implement the business logic and URI is the cloud service call to serve data.

TABLE III. MOBILE DEVICE APPLICATION PROTOCOL NODE

Node Name	Node Description
PAGE	Group ID INDEX:Group Index EXPAND:If Expanded TITLE:Group Name
ITEM	Function ID
TEXT	Function Name
ICON	Icon Path of Function
TYPE	Service Type of Function WEB:Web Page SYSTEM:Local Service of Mobile Device(e.g.:Voice,SMS,GPS) CONTENTAPP:Interior Service PLUGAPP:Exterior Service WEBSERVICE:Cloud Service
URI	URI by Type WEB:URL of Web Page SYSTEM: Local Service of Mobile Device CONTENTAPP: Interior Service PLUGAPP: Exterior Service WEBSERVICE: Cloud Service
PARAMS	Parameters for URI
DESCRIPTION	Description of Function

IV. THE CLIENT IMPLEMENTATION OF CUSTOMIZED LOCAL APPLICATION

By the aforementioned configuration model and protocol, the mobile device generate and submit the system with the configuration interpreting engine. With the combination of user authentication in the cloud and mobile device functionality configuration, at user logon, the cloud service sends customized configuration in XML defined by the protocol, to the mobile device client. The client interprets the response XML file and constructs the navigation interface according to the group, functionality name and icon in the XML file. Each node is the entry point of the function.

The system navigation interface represents all function points with a grouped list and users click a function point to exercise the function. Obviously, the configuration style of the interface make it possible to accustom to the changing requirements, and thus improve responsiveness of the system. Essentially it solves the problem of frequent update. The generation algorithm is shown in Figure 2.

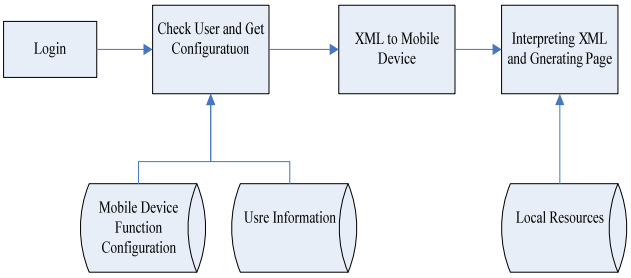

Figure 2. System navigation interface generation algorithm

When users choose a function point, the system employs a class factory pattern, looks up the associated processor from the local configuration XML file according to the TYPE (WEB, SYSTEM, CONTENTAPP, PLUGAPP, WEBSERVICE). Then the service is called by invoking ACTIONDO method. The implementation algorithm is shown in Figure 3.

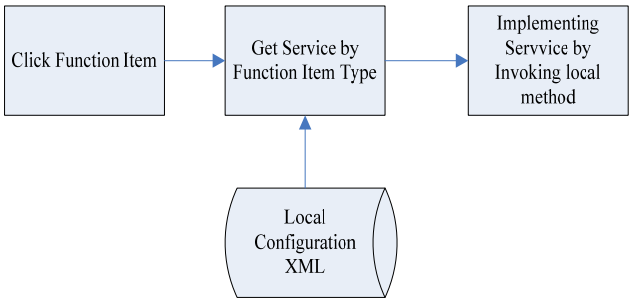

Figure 3. System functionality implementation algorithm

Application system generated by mobile device is showed in Figure4.

Figure 4. Application system generated by configuration interpreting engine

© the authors

V. CONCLUSION

Comparing with browser-based applications, the local applications have the advantage of better integration with local services such as voice, SMS and GPS etc, and better user experiences. They also have the problem of updating softwares once the service is changed. This problem is particularly exacerbated with frequently varying requirements. Frequent redevelopment delays deployment of new service and affects the market competitiveness. Moreover, frequent updates bring inconvenience to users and introduce the challenging problem of version control.

In this paper, a solution based on cloud service is proposed. The local functionality of the mobile devices is governed at the cloud side. Users obtain the up-to-date configuration from cloud services at log on. The local application system is then automatically generated according the configuration via a local configuration interpreting engine. This system model avoids frequent redevelopment and update and is able to help enterprises accelerate deployment of new services and improve user experiences, which, ultimately, strengthen the competitiveness of the enterprise.

ACKNOWLEDGMENT

The work of Zhuo Gao is supported by Funding Project for Academic Human Re-sources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality (PHR201108268).

REFERENCES

[1] Chen Donglin,Ma Mingming,Lü Qiuyun. Cloud Service Collaboration Market Transaction Model [J]. Journal of Wuhan University of Technology(Information & Management Enginerring),2011,33(3):456-459.DOI:10.3963/j.issn.1007-144X.2011.03.032.

[2] Dong Xiaoxia, Lü Tingjie. Review of the Cloud Computing and Its Future Development [J]. Journal of Beijing University of Pots and TelecommunicatIions(Social Science Edition),2010,12(5):76-81.

[3] Rajkumar Buyya,Chee Shin Yeo,Srikumar Venugopal.Cloud computing and emerging IT platforms: Vision, hype, and reality fordelivering computing as the 5th utility[J].Future generations computer systems,2009,25(6):599-616.DOI:10.1053/j.gastro.2007.03.107.

[4] Zong Mei,Ma Xiaoping. Structure and Application of Three-tier Client/Server Structure Based on .Net [J].Computer Engineering and Design,2005,26(2):520-522.

[5] Fan Yinting,He Hongyun. The Research of 2-Tier and 3-Tier Structure Based on the Client/Server Architecture [J].Application Research of Computers,2001,18(12):23-24,40.

[6] Yu Chongchong. Management Information System on Three Layers Client/Server [J]. Application Research of Computers,2000,17(7):93-95.

[7] Hatton, L..Empirical Test Observations in Client-Server Systems[J].Computer,2007,40(5):24-29.DOI:10.1109/MC.2007.166.

[8] Iliev, A.,Smith, S.W..Protecting client privacy with trusted computing at the server[J].IEEE security & privacy,2005,3(2):20-28.

[9] Yue Guohua. Design and implementation of client software automatic upgrade for C/S pattern in distributed environment [J]. Journal of Xi'an University of Science and Technology,2011,31(1):72-76.

[10] Canfora, G.,Di Santo, G.,Zimeo, E. et al.Developing Java-AWT thin-client applications for limited devices[J].IEEE internet computing,2005,9(5):55-63.DOI:10.1109/MIC.2005.99.

[11] Ilias Michalarias,Arkadiy Omelchenko,Hans-J. Lenz et al.Fclos: A Client-server Architecture For Mobile Olap[J].Data & Knowledge Engineering,2009,68(2):192-220.DOI:10.1016/j.datak.2008.09.003.

[12] Haibo, B.L..Future internet services and applications[J].IEEE Network,2010,24(4):4-.

© the authors