

Design Modules for the Application Software Development for Mobiles

Navyashree T M

Assistant professor, navyashree.tm@gmail.com
Dept. of Computer Science and Engineering,
DSCE, Bangalore – 560078

Spurthi BS

Android Developer, spurthik@gmail.com
Bangalore – 560078

Abstract— Application development for the mobiles is continuing to grow at a greater rate. A research is done and implemented for few design modules in which multi-platform mobile applications software can be developed. Designing a module to build software application play important role in developing and deploying applications. To subsist with the changing aspects, deploying cost and maintenance cost, a module of separation of objects through which application can be easily sustained and ported to non-identical platforms.

Keywords— Mobile Application, Design module, Mobile platform.

I. INTRODUCTION

Mobile software development is the process of creating software which can be used on a mobile device.

Mobile application development is the process by which application software is developed for small low-power handheld devices such as personal digital assistants, enterprise digital assistants or mobile phones. These applications are either pre-installed on phones during manufacture, or downloaded by customers from app stores and other mobile software distribution platforms. Users of mobile applications have diverse preferences, so extensive improvements to traditional system development methodologies are needed to keep up with this demand. A mobile application should be developed within short period to meet time-to-market. A developer for mobile application software has no time to consider the architecture of software because they have to meet the deadline for software launching. Mobile application framework is based on MVC (Model View Controller) architecture, but the developer of the application need not follow the architecture of the framework. The bad design of the application includes jumbling of the model, view and controller of the application in a single class. This design will certainly affect the performance of the application and decreases the maintenance cost.

In the Fig. 1, the UI related functionalities (OnDraw(), call Graphic()), Data Content related functionalities (Parse Content(), SendHttp()) and Data Observers related Contents (HandleErrors() ,ConnectTOServer()) are mixed in the Main Class. A change from requirement might affect entire source codes in this structure. The cost of maintenance will be increased because of unstructured mobile application software.

Class MainClass

```
Public:  
MainClass ();  
virtual ~MainClass ();  
void CallGameEngine ();  
bool Initalize();  
void OnDraw();  
void Parse_Content();  
void callGraphic();  
void DrawBitmap();  
void Send_HTTP();  
void InitalizeObjects();  
void ConnectToServer();  
void HandleHttpRelatedErrors();  
void HandleParsingRelatedErrors ();  
void HandleUIRelatedErrors();
```

Fig. 1 Mobile Application Code Structure

To solve similar related issues we came out with a design pattern through which the development and maintenance of the application is made easy. Each defined component in the design pattern has its own transformation rules corresponding to each mobile platform.

The paper is organized as follows. In Section II, we discuss the related work, In Section III we define a design module for the application development. Section IV we explain (with case study) concrete mobile application using the Design Modules explained and effectiveness of Module . The last section highlights the main contributions and points to ongoing and potential future work.

II. RELATED WORK

At the early days of mobile technologies, the typical utilization of handheld devices was mainly basic personal communication purposes (e.g. initiating calls, exchanging messages, etc.). Nowadays, providers of these devices

exploded the market with a large variety of mobile devices such as portable media players, PDAs, Smart phones, and Pocket PCs. Besides, with new adopted networking and telecommunication technologies such as Wi-Fi and GPRS, the

use of handheld devices becomes very popular and these devices are used for a wide range of services including internet navigation, emailing, location-based and context aware services, and mobile learning. State of the art shows that many mobile applications have been developed to enrich the appliance of these devices and to benefit from their features. Most of these applications are very sensitive to performance (Quality of Service), security, and availability. Examples of these applications use video streaming, image processing and mobile learning services (m-Learning).

Nowadays, very few mature works have been done on the development and management of mobile applications and mobile Web services. However, considering usability and quality of these applications will play a key element in using them. On the one hand, providers of these applications will need to guarantee a better usability of mobile applications and their corresponding mobile Web services to remain competitive and maximize the return on investment from their businesses. On the other hand, mobile clients will have the possibility to look for mobile applications and mobile Web services with higher usability, quality, and lower cost. Achieving such requirements necessitates a concise development approach of both mobile Web services and mobile applications. State of the art in this issue still lack solid and proven development approaches.

In [1], authors propose an approach for developing mobile applications; they call this structure as *Application Framework*. This framework is based on MVC pattern to help maintenance. It can support various mobile platforms if there is a transformation rules for specific mobile platform or new mobile platform. The component also has a code template using XSLT (Extensible Style sheet Language Transformations) for code generation.

The work proposed is good for the complex projects and can be useful for similar application of the *Application framework* is built. To build upon their proposed work we came up with the Design modules which can be applied to any application irrespective of the platform .Mobile Applications keep on updating by adding new features to the existing one. To support this and to reduce the maintenance cost Design modules are proposed.

III. DESIGN MODULES

Major Features

In Fig. 2, the Design Module defined in this chapter consists of four major elements - Data Modeller, Data Observers, Data displays, Data Containers.

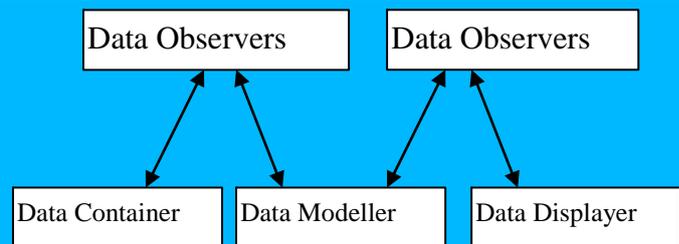


Fig. 2: Design Modules for the Mobile application

Data Containers hold the actual data of the application, all the application needed for the application are present in the Data Containers. Here Data Container is the abstract way to represent the data; it may hold many classes through which data can be communicated depending on the application. Data modeller is the similar to wrapper classes'. All the data to the Containers and Displays are communicated through the Data Modeller through the Observers. Data Displays are the UI specific classes which include the Complete UI creation, switching forms and displaying the data along with the graphics. Data observers act like interfaces for communication. Single observer can be used through the application.

A. Data Displayers

Data Displayers consists of classes which is used to display the graphics along with the data. In the framework we worked on each screen is a form .It is the frame first upon which we keep on adding forms. Depending on the application each form can be a represented as the class and each class contain display related UI and corresponding data can be fetched form the Data modellers through the observers.

Data Displayers also contain the form management classes through which through which corresponding class objects can be fetched and displayed. From the Fig. 3, only functions related to UI are present in the Display Class. Form management class can be the controller object for the UI. Depending on the application Form manager can be made as the controller class for the entire design modules. The design modules certainly depend upon the complexity of the application that is being developed.

B. Data Observers

Data Observers are like interfaces through which the communication to the different modules is possible. Depending on the application complexity required number of observers used. The most common observers used in the application includes,

```

Class Data_Displayer

public BaseClass,
public FormManagerClass,
public TouchEventListenerClass,
public SlidableListEventListenerClass,
public CustomItemEventListenerClass
public:
    Data_Displayer ();
    virtual ~Data_Displayer();
    bool InitializeForm ();
    result OnInitializing ();
    result OnDraw ();
    result OnTerminating ();
    void OnTouchPressed ();
    void OnTouchDoublePressed ()
    void OnTouchFocusIn ()
    void OnTouchLongPressed ()
    void OnTouchMoved()
    void OnListPropertyRequested()
    void OnLoadToTopRequested()
    void OnLoadToBottomRequested()
    
```

Fig. 3 Design Modules of the Data Displayers

1) *Download Notifiers*

If the application is using the network to access the data content, the calls to the network will be asynchronous. If the application is threaded model, each thread may call the network asynchronously.

```

Class IDownloadNotifier

virtual void NotifyDownloadCompleted (char * buf)=0;
virtual void NotifyImageUpdate (int i , int j) = 0;
    
```

Fig. 4 Design Modules of the Data Observers

There has be Notifiers to each download done by the each thread .Multiple call-backs from the server are handled by the Notifiers and through the application logic the data is passed to the Data modeller. Download Notifiers are more like the Network Managers, managing all the network related contents.

2) *Notifiers to UI*

All data in the application are displayed through the UI screens. If the data is not present Data need to be fetched need to be fetched form the Data Containers. Call back to the UI about the request can be synchronous and asynchronous. So it's a good practice to use Notifiers to UI.

C. *Data Modellers*

Data Modellers are the wrapper class, it contain functionality that are needed by the UI module and also the Container module. Every communication between the Data Container and Data Displayer happen through the Data

Modeller. All the Classes present in the Data container are initiated with the single instance of the Data Modeller. Application specific design objects can initiate only once and used through the application. The need for the Data Modeller here is to provide a layer of separation. This design follows the layer architecture and help in layer to layer communication. All the requested data from the UI are sent to Containers through the modeller and corresponding call-backs containing data is sent from Container to the UI through the Modeller.

D. *Data Containers*

Depending on the application the Data Containers can be designed. If the application is stand alone application let's say a simple game where network access is not the criteria then all the game related data, might be the game engine should be present in the Data Container. If the application has the network support then Data Containers contain class related to the HTTP, WEB, SOCKET etc. The need for the Data Containers is, if the application is complex and each module is designed and coded separately .Data container module contain only raw data and this developers need not to know the way data is displayed by the Displayers. They are only concerned about sending data to the Data Modeller .Any change to the Data Containers can be handled separately.

IV. CASE STUDY

We developed simple mobile application software, a game to validate the Design Modules. The game is Chess; it includes the local play in the standalone mode with the Artificial intelligence and also the network mode where player plays against other through Internet. Game has many screens to be specific more forms; each form has a different UI. The player has options to choose the different modes of play like local or the network mode. Local play takes the player to the playing screen and additionally provide options for different game modes, difficulty etc. Network play includes logging with the valid user ID and getting the list of available players. Selecting a player and playing against each other in the network.

Fig.5 shows the Design modules through which the game is implemented. Since the game is developed for both stand alone and also the Network, Data Container holds the entire data. Figure 5 shows the abstract view of the Data Container. For the local play it need move validation functionalities which is present in Container itself or the can access through the Game engines through functions.

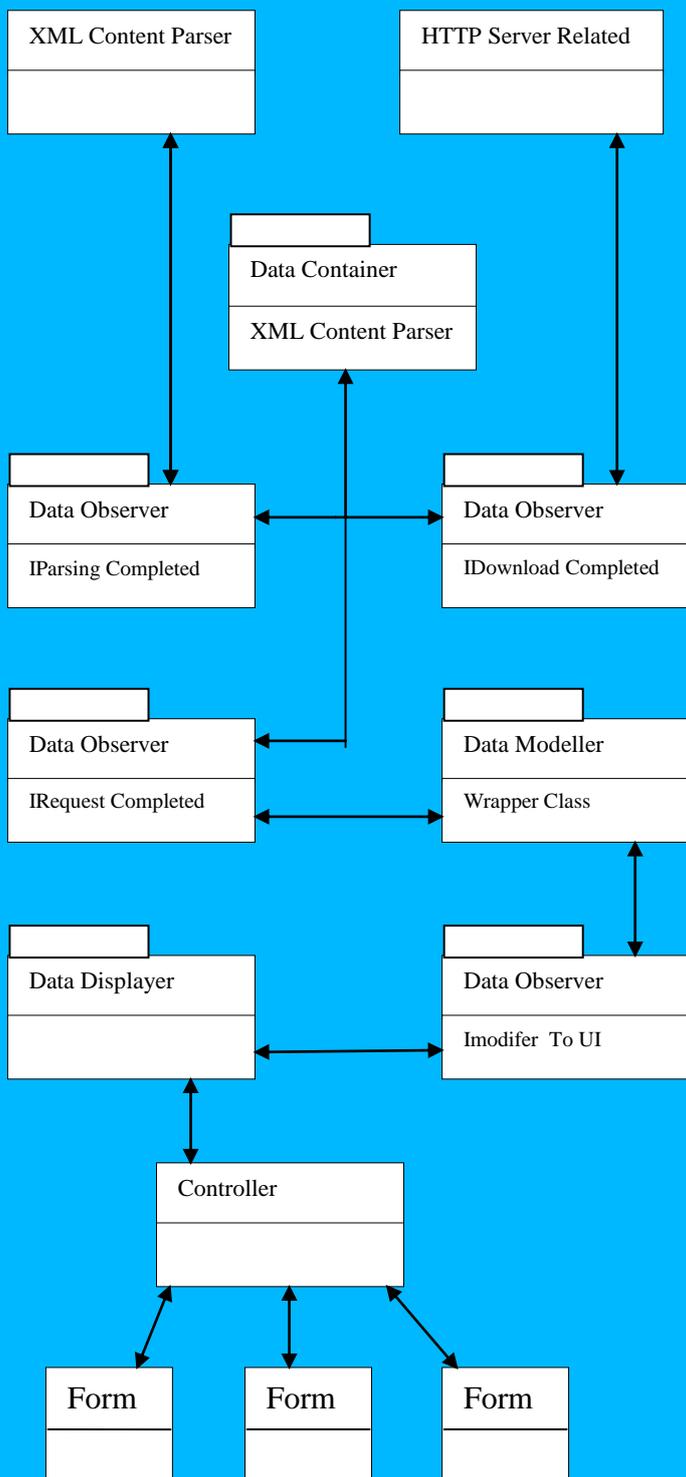


Fig. 5 Design Modules for the Game Implemented

For the network play Http request is posted to the server and data is retrieved. The contents are parsed and passed to the corresponding Displayers. Anything related to the Game Engine or the network issues is present in the Data Container. So the feature enhancement or the maintaining the backend code is made simple. Data modellers provide the wrapper, ie it contain all functionalities that are needed in the Backend code and also in the Displayer part. The need for the Data Modeller

here is to provide a layer of separation. The game chess has more than 15 screens and each screen with different UI. There is Form Manager (Controller class) which helps in switching between the forms. Any change in the UI is done in the displayer unit and will not affect the backend data part. Any changes to the application can be done easily.

V. CONCLUSION

In this research, we worked on the few designs with respect to mobile application development considering the general structure of the application. We also test a case study in which Design Modules applied and the study proves that along with Application Framework mentioned in the (Related works) Design modules help to manage the application well and provides an efficient way to maintain and manage application by reducing time and cost. In the future works we will be implementing this to very complex project and comparing result and the maintenance issues and looking more into the Design Modules and for the WAC (Whole sale Application Community). It is basically a top level technology that is meant to allow developer to write code once and deploy to different platforms and devices. Coding is done in HTML5, JavaScript and CSS.

REFERENCES

- [1] Choi, Y, Yang, J.S, Jeong, J "Application Framework for Multi Platform Mobile Application Software Development ,ICACT ,2009
- [2] Object Management Group: "Model Driven Architecture (MDA)",OMG Document ormsc / 01-07-01
- [3] World Wide Web Consortium, "XSL Transformations (XSLT)", W3C Recommendation, 1999
- [4] Choi, yoonseok, "A Meta Model for Mobile Application Development", KCSE 2008, 2008-02-20
- [5] Hojin Cho, Jin-Seok Yang, "Architecture Patterns for Mobile Games Product Lines", ICACT, 2008