

Applying Model View View-Model and Layered Architecture for Mobile Applications

Shashi Prakash Tripathi and Tulika Narang

Centre of Computer Education

Institute of Professional Studies

Science Faculty, University of Allahabad, India

Email: shashi.123.prakash@gmail.com, n.tulika@gmail.com

(Received July 27, 2016)

Abstract: The paper applies the Model View View-Model for mobile applications. The exponentially increasing user of mobile devices is leading to a next generation of applications that exploit user contextual information to provide a richer experience. The activities to perform during the development of application may vary on their context. The paper lays emphasis on the layered architecture of mobile applications. The methodological process followed during the ontology development as well as the ontology platform obtained from this process. Besides, we provide an example of how to extend the ontology for a particular use case in a concrete domain.

Key words: Model view, software architecture, design pattern, software engineering.

1. Introduction

In today's scenario mobile applications consist of wider range of activities than other electronic Wireless Handled devices. As the technology is growing, mobility shifts the global computing infrastructure homogeneous, static, powerful desktop computers to heterogeneous, highly dynamic, resource-constrained handheld and wearable computers. This new edge of computing context requires entirely change and upgraded architectural paradigms that gives the solution of mobile software development and centered for same platform Wireless Handled devices and networks and take advantage of the opportunities afforded by mobile systems. The recent work in this field rapidly changes the architecture and mobile systems. Researchers have created formal models and uses

engineering processes for mobile software. The principles of software architecture are the development of environments and runtime platforms that support mobile software models and processes. Mobile system is consists of mobile software and hardware elements. These mobile applications are distinguished by software architectures that are designed for and essentially facilitate mobility.

The current age is of Wireless Handled devices. Everybody is operating everything on their mobile phones. There are various platforms of mobile phones like android, java, windows and iOS. The architecture of application is an issue for developers. There are various architectural patterns to implement the architectures like MVC, MVVM and MVP. Earlier the MVC is used for web based applications but MVVM [Ken Cooper & Ted Peters, 2005]. A mobile application will generally be structured as a multi-layered application consisting of user interface, application and data layers. When developing a mobile application, basically a Web-based application. If you are developing a client based application, the application and data services layers are likely to be located on the local memory. A Web-Based application requires local processing and must work in an occasionally connected scenario, consider designing a client based application¹. The complexity of application is reduced and reliable by using the MVVM² pattern. Much attention has been focused on using Ontology and MVVM² pattern to design a graphical and Web-based application. The mobile application is need of every mobile user and it should be more users friendly and more responsive in nature. The architecture that is proposed in this paper is more reliable and responsive. A web-based application architecture using thin-client architecture allows application to run on the server and thin page is only part of the application that is transferred to wireless handled devices, thereby removes the need for sophisticated client devices. We use XAML while we are designing our application. XAML, Extensible Application Markup Language [Microsoft, 2008].

2. Background and Related Work

The multi-layered architecture of mobile application has already introduced³. A similar approach has followed to develop an application. The application was based on E-learning⁴. In this approach there are three layers as shown in figure 1 listed below:

1. Presentation layer
2. Application layer

3. Data access layer

The first layer, the presentation layer in application contains the series of form, pages and views to interact with user. Each view contains many fields that display values related to lower layers and also take input and send it to lower layers for processing. This layer mainly focuses on user interface.

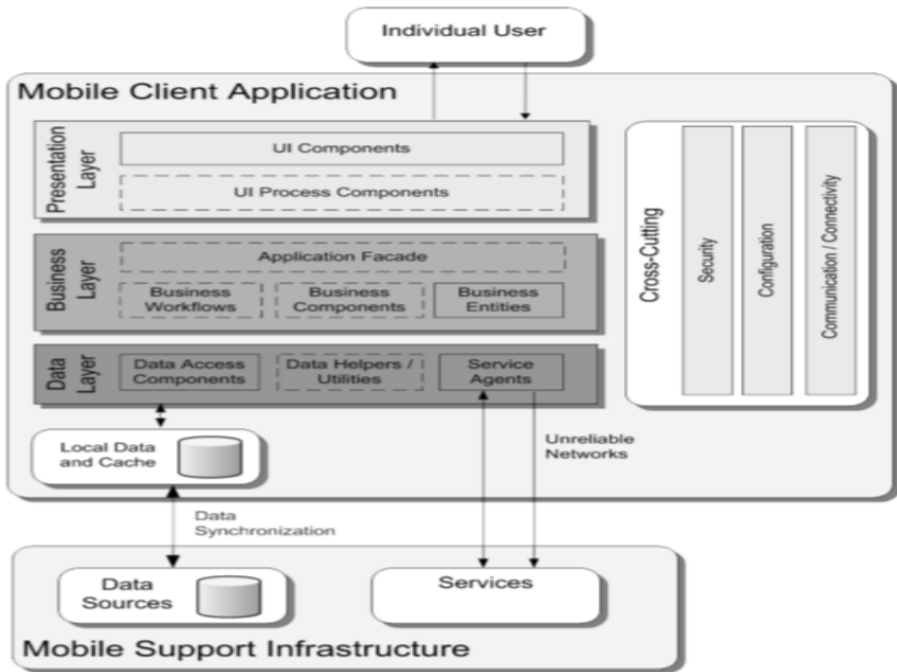


Fig. 1: Layered Architecture of Mobile Application³

The second layer, the application layer contains the logic and functionality that the application is performing. This layer focuses on what and how the tasks are going to happen. The last layer, the data access layer is responsible for exposing the data stored in database. This also consists of data access components and Service gateways.

3. Problem Statement

The mobile architecture that was provided earlier is tightly coupled and lack of modularity that creates the listed problems:

1. Maintaining, upgrading the application.
2. Failing to use a message-based interface between layers.
3. Not so secure you can directly access data from Application layer⁵.

- 4. Relying on validation that occurs in the presentation layer.

3. Solution

Here we have given solution for the problem

1. Using Design Pattern Model View View-model With Layered Architecture

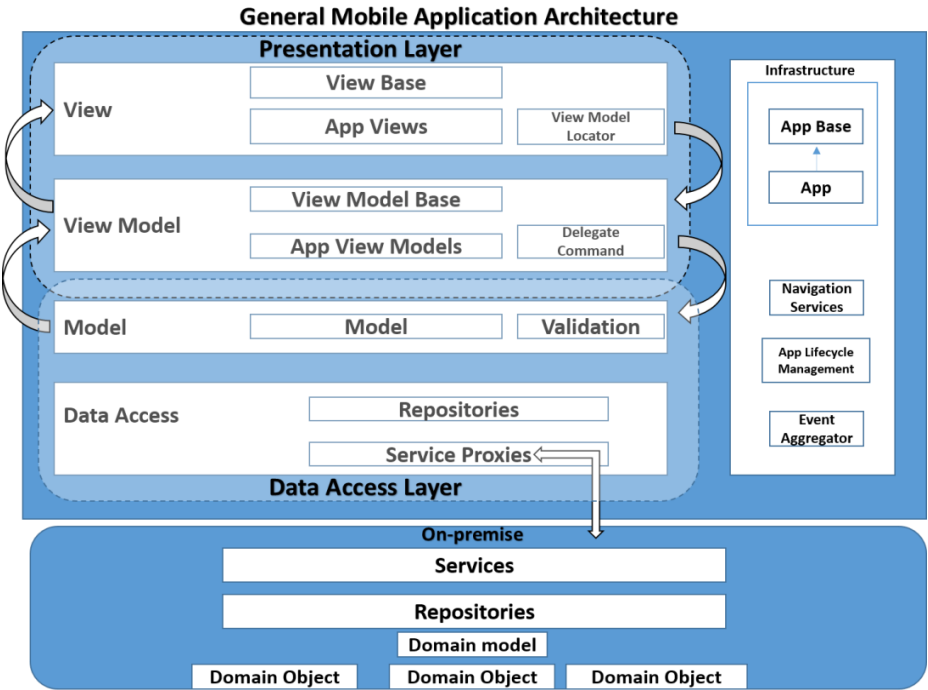


Fig. 2: Multi Layered Architecture with Model View View-Model Design Pattern

When we use Model View View-Model with the layered architecture it increases the modularity of application this helps to maintain and upgrade software easily^{6, 7}. In the figure 2, the presentation layer consists of many modules termed as view. The view module contains view base, application views and View-Model locator.

The view base contains the first view when an application starts and the Application view contains all views that will Application have and the last View-Model locator contains the relation between views and if the view is dynamically accessed then the location rom where it can be accessed.

The Application layer contains View-Model that has view model base and application view models and delegate commands. The delegate command is there for accessing different models for the application.

The Data access layer contains the model and database. The model contains model and validation. The validation resides at data access layer that get rid from validating at presentation layer. And the database has its own local repository if requires then on-premise it can request for remote database or cloud one.

2. Ontological Approach for the Mobile Application

Figure 3 is representing how we can develop an application ontologically with design pattern. It shows the different layer as the classes and attribute as the content of Model View View-Model design pattern.

The Application class containing the attribute Requirement, objective and the task. The requirement of the Application is full filled by the function class that as attributes Logical Structure, Interfaces and Constraint.

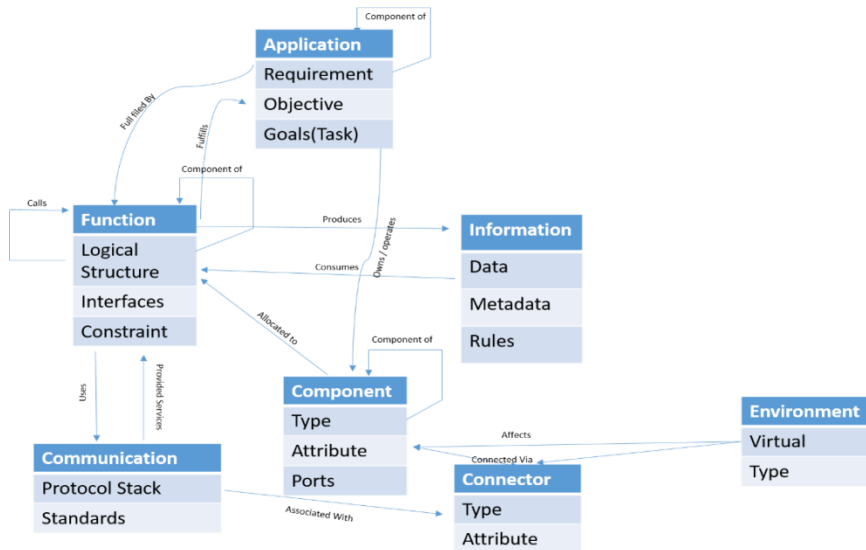


Fig. 3: Ontological Representation of Mobile Architecture

The functionality of the application is completing the objective of the application. The function class produces the data that stores in information class that has attributes like Data, Metadata and rules. The interface attribute of function class is used providing interface between other classes like communication class, environment class^{8, 9}. The component class that explained in figure 3, consists of component type, ports and other attributes. The components that require performing any action allocated to logical structures of function class. The connector class is used to connect the

classes using their attributes. If the application requires the data from network then it can communicate with server using communication class that has protocol stack to establish connection, fetch and transfer the data. The application class and function class are self-referential structure because they have to perform some operations that are related with their attributes. The ontological platform also provides the multi-platform architecture for any application¹⁰. This also helps us to increase the performance of the application¹¹.

In both the solution either the multi layered architecture with design pattern or the ontological approach gives the application modularity. The application that is provided is highly cohesive and loosely coupled¹²⁻¹⁶.

4. Conclusion and Future Work

In this paper, we have applied a design pattern with multi-layered architecture, Model, View and View-Model. We believe this is useful for new mobile application developer to understand the different type of concerns for mobile application. However, this compilation by no means is comprehensive to cover all the possible concerns of mobile application domain. We provide a modular view for application that is more secure and easy to maintain. Future work will be to continue exploration of possible concerns and design patterns.

We will try to make it dynamic on contextual basis.

References

1. Nenad Medvidovic and George Edwards, Software architecture and mobility: A roadmap, *The Journal of Systems and Software*, **83** (2010) 885–898.
2. F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*, Wiley, 1996.
3. J. D. Meier, Alex Homer, David Hill, Jason Taylor, Prashant Bansode, Lonnie Wall, Rob Boucher Jr and Akshay Bogawat, *Mobile Application Architecture Guide*, Pattern & Practices, Microsoft, 2008.
4. Manouchehr ZadahmadJafarlou, Bahman Arasteh, A pattern-oriented web based architecture to support mobile learning software development, *Procedia - Social and Behavioral Sciences*, **28** (2011) 194-199.
5. Swarnpreet Singh, Ritu Bagga, Devinder Singh and Tarun Jangwal, Architecture of mobile application, Security issues and services involved in mobile cloud computing environment, *International Journal of Computer and Electronic Research*, **1(2)** (2012).

6. E. Gamma, R. Helm, R. Johnson and I. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, *Pearson Education*, 1994.
7. Li Liu, Analysis and Application of MVVM Design Pattern, *Application of Micro Computer*, **28(12)** (2012) 57 -60.
8. Medvidovic, Nenad, Mikic-Rakic, Marija, Mehta, Nikunj, Malek, Sam, 2003. Software architectural support for handheld computing, *IEEE Computer – Special Issue on Handheld Computing*, **36 (9)** 66–73.
9. Malek, Sam, Edwards, George, Brun, Yuriy, Tajalli, Hossein, Garcia, Joshua, Krka, Ivo, Medvidovic, Nenad, Mikic-Rakic, Marija, Sukhatme and Gaurav, An architecture-driven software mobility framework, *The Journal of Systems and Software*, **83(6)** (2010) 972–989.
10. Y. Choi, J. S. Yang, and J. Jeong, Application Framework for Multi-Platform Mobile Application Software Development, *International Conference on Advanced Communication Technology*, (2009) 208-213.
11. F. Aquilani, S. Balsamo, and P. Inverardi, Performance analysis at the software architectural design level, *Performance Evaluation*, **45(2-3)** (2001) 147-178.
12. R. Pressman, *Software Engineering: A Practitioner's Approach*, 7th Edition, McGraw-Hill Science/Engineering/Math, 2009.
13. B. König-Ries and F. Jena, Challenges in Mobile Application Development, *Information Technology*, **52(2)** (2009) 69-71.
14. Y. Natchetoi, V. Kaufman and A. Shapiro, Service-Oriented Architecture for Mobile Applications, *Proceedings of the 1st international workshop on Software architectures and mobility (SAM '08)*, (2008) 27-32.
15. P. Clements, et al., *Documenting Software Architectures Views and Beyond*, Addison-Wesley, 2003.
16. Haihong Pan, A scalable graphics user interface architecture for CNC application based - On WPF and MVVM, *Advanced Materials Research*, 1931 -1935.