Indian Journal of Science and Technology, *Vol 8(18), DOI: 10.17485/ijst/2015/v8i18/75937, August 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Mobile Application Development Tool based on Object Relational Mapping Solution

Jinhyung Cho[1*], Hwansoo Kang[1] and Seawoo Kim[2]

[1]School of Computer Engineering, Dongyang Mirae University, Seoul - 152-714, South Korea; cjh@dongyang.ac.kr, hskang@dongyang.ac.kr
[2]Department of Digital Media, Soongeui Women's College, Seoul - 100 751, South Korea; nt21jjang@naver.com

Abstract

With the rapid growth of mobile application market for smart device users, an efficient development tool for mobile application software has become important and is essential. In this study, we propose a framework of mobile application development tool using Object Relational Mapping (ORM) solution in front-end domain layer, especially for mobile database-driven application system. The experimental results indicate that the development method using the proposed approach is more efficient compared to the conventional development method.

Keywords: Database Application, Development Tool, Front-End Domain Layer, Object Relational Mapping, Mobile Application

1. Introduction

Mobile application software (called mobile App.) development is the process by which application software is developed for smart devices, such as mobile smart phone, smart tablet computer or smart TV[1,2]. Mobile application development has been steadily growing, both in terms of revenues and jobs created. A 2013 analyst report estimates there are 529,000 direct App. Economy jobs within the EU 28 members, 60% of which are mobile App. Developers[1]. With the rapid growth of mobile application market, an efficient development tool for mobile application software has become important and is essential.

The application development tool is a software system that provides comprehensive facilities to programmers for software development. Most modern development tool offer intelligent code generation features. In this study, we propose a framework of mobile application development tool using Object Relational Mapping (ORM) solution in front-end domain layer. It allows mobile application to be implemented much faster, and makes it easier to change domain requirements especially for mobile database-driven applications system in System Integration (SI) industries.

2. Mobile Application Development and Object Relational Mapping

Object Relational Mapping (ORM) is a programming technique in which a metadata descriptor is used to connect object code to a relational database[3-5]. Object code is written by object-oriented programming languages such as Java or C#. ORM converts data between type systems that are unable to coexist within relational databases and object-oriented programming languages. An ORM is defined by Wikipedia as: "A programming technique for converting data between incompatible type systems in relational databases and object-oriented program-

**Author for correspondence*

ming languages. This creates, in effect, a "virtual object database", which can be used from within the programming language". An ORM has to provide a facility to map database tables to domain objects, usually a design surface or wizard. This mapping is in-between your database and domain model, independent from the source code and the database. The ORM runtime then converts the commands issued by the domain model against the mapping into back end database retrieval and SQL statements. Mapping allows an application to deal seamlessly with several different database models, or even databases[6-10]. When developing mobile applications, one of the important issues is database. Thus, a very large attention is given to good design of the database schema and database tuning techniques. For building database-driven applications the level of independence between the developed source code and the actually used database management system is an important problem[11-14]. One of the possible solutions is to use ORM as an intermediate layer.

3. The Object-Relational Impedance Mismatch

Database normalization theory includes different strategies than object oriented programming theory, which is based on software engineering principles. Database tables model data and prescribe storage techniques. Objects model data and behavior. The problem is that there are subtle differences between the way a database is designed and the way an object model is designed. The approach of doing straight mapping of database tables to objects leads to the famous "object-relational impedance mismatch"[4].

4. Proposed Approach

The overall process of our proposed mobile application development method is divided into four phases as shown in Figure 1. In the first phase, design result is decomposed into the unit and registered to the Unit Builder. In the second phase, the final UI layout is created through designer and developer. In the third phase, the work process of each unit is registered by developer. In the last Back office service phase, the API function is generated from the work process for developer to implement framework in server side.

We have designed the platform architecture of the proposed mobile application development tool and implemented pilot system as Figure 2. It shows the platform architecture of the pilot system. The pilot system operates each agent independently so that the whole system remains stable during experimental substitutions or adjustment of an agent. The proposed solution enables a developer to define the functional framework in the design process with an elegant and intuitive form.

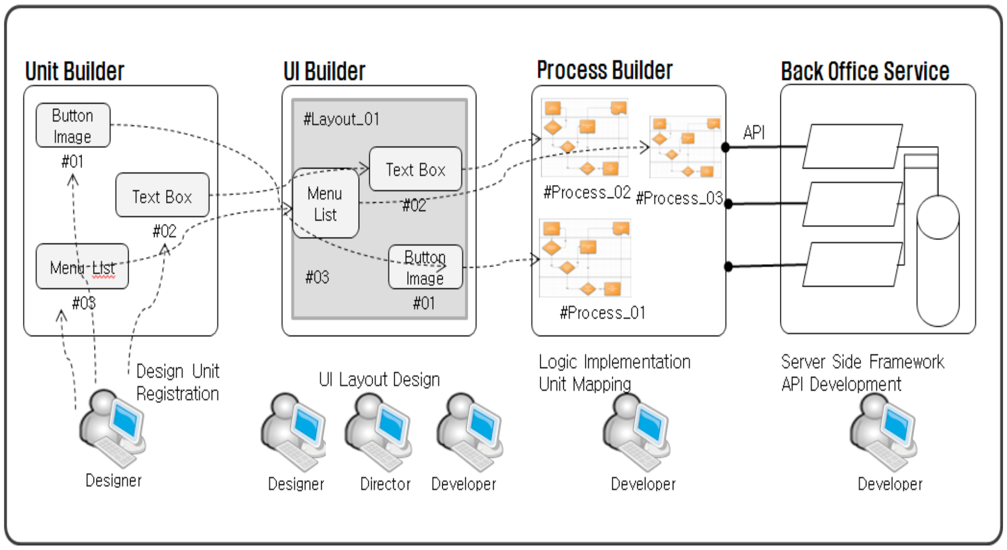


Figure 1. Overall process of the proposed method.

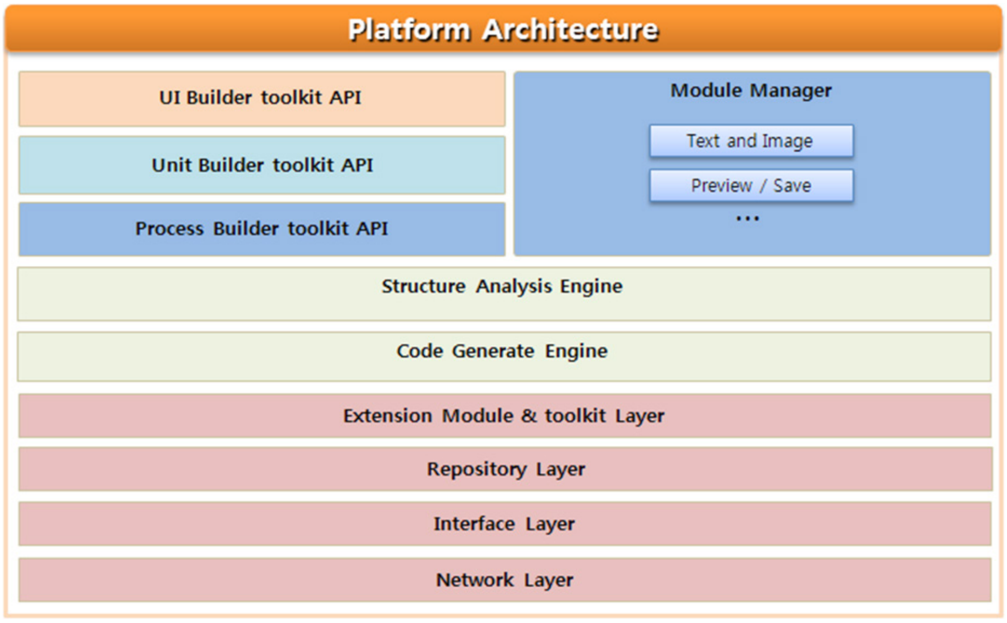
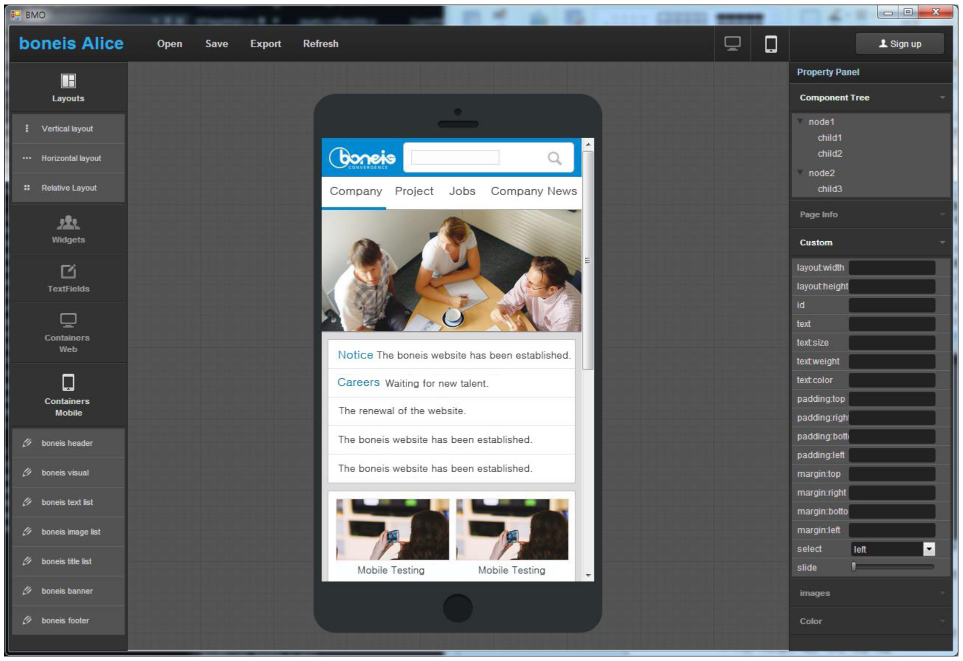


Figure 2. Platform architecture of the pilot system.



Notice The boneis website has been established.
Careers Waiting for new talent.
The renewal of the website.
The boneis website has been established.
The boneis website has been established.

Figure 3. UI builder of the pilot system.

Figure 3 shows the UI builder of the pilot system. The pilot system operates each agent independently so that the whole system remains stable during experimental substitutions or adjustment of an agent.

5. Evaluation and Conclusion

We simulated the overall process of our proposed solution through mobile and web application development project. The purpose of the simulation test is to prove the

Table 1. Performance gain over NA model of ORM model

Benchmark Model	Number of developer			Average
	5	10	20	
[ORM Model] On Web Non DB App.	6.72	10.02	13.09	9.94
[ORM Model] On Mobile Non DB App.	10.23	14.10	17.29	13.87
[ORM Model] On Web DB App.	10.48	16.81	18.42	15.24
[ORM Model] On Mobile DB App.	14.67	19.51	23.22	19.13

usability and economic efficiency of the solution in mobile database-driven application development. In the simulation, we have experimentally evaluated the performance of the proposed solution [ORM model] by comparing with the conventional development process model [NA model] on various development projects, which are related to mobile/Web and Database/Non-database applications. The simulation results provide evidence that the proposed solution can provide the performance gains in terms of development cost. The proposed solution could save development cost especially in mobile database-driven application development. Therefore, our study offers a very meaningful, practical implication to mobile database application developers and/or marketers. As the measure for performance evaluation, this study adopted the saving rate of development cost and time widely used in researches related to the software development tool verification[2]. The experimental results provide evidence that the proposed ORM model is generally better than the conventional model as Table 1. In addition, it was proven that the proposed ORM model could improve the development performance to a greater extent than could the conventional model by experiments.

Future work in this area should be along the following directions. First, the UI builder agent should incorporate more refined user interface. Second, the performance of our proposed solution should be experimentally evaluated by comparing with other commercial development tools in the next consequent study.

6. Acknowledgement

This work was supported by 2013 Industry-Academy-Research Collaboration Project for Technology Development funded by Small and Medium Business Administration in the Republic of Korea.

7. References

1. Vision Mobile, Plum Consulting. European App Economy; 2013.
2. Boniewicz A, Gawarkiewicz M, Wisniewski P. Software Development Discussion Paper: An overview of mobile development in the context of current technology. International Journal of Software Engineering and Its Applications. 2013; 7(4):189–96.
3. Amber S. Agile database techniques: effective strategies for the agile software developer. USA: John Wiley and Sons; 2011. Available from: http://www.agiledata.org/
4. Amber S. Mapping objects to relational databases: O/R mapping in detail. 2006. Available from: http:///www.ambysoft.com/mappingObjectsTut.html
5. Bauer C, King G. Hibernate in action. Manning publishers Co. Boston/Massachusetts; 2006.
6. Blaha MR, Premerlani WJ, Rumbaugh JE. Relational database design using an object-oriented methodology. Communications of the ACM. 1988; 31(4): 414–27.
7. Catell RGG. Object data management: Object-oriented and extended relational database systems. Addison-Wesley Publishing Company; 1991.

8. Catell RGG, Barry D, Barler M, Eastman J, Jordan D, Rusell C, Schadow O, Starienda T, Velez F. The object data standard: ODMG 3.0.San Franscisco: Morgan Kaufmann Publishers; 2000.
9. Begg C, Connolly T. Database systems: A practical approach to design. Implementation and management, 5th ed. Addison Wesley; 2010.
10. Cho W, Hong K, Loh W. Estimating nested selectivity in object-oriented and object-relational databases. Information and Software Technology. 2007; 49(7):806–16.
11. He Z, Jerome D. Evaluating the Dynamic Behavior of Database Applications. Journal of Database Management. 2005; 16(2):21–45.
12. Hoffer J, Prescott M, Topi H. Modern Database Management. 9th ed. Pearson Prentice Hall; 2009.
13. Pardede E, Rahayu J, Wenny T, Taniar D. Object-relational complex structures for XML. Information and Software Technology. 2006; 48(6):370–84.
14. Philippi S. Model driven generation and testing of object-relational mappings. Journal of Systems and Software. 2004; 77(2):193–207.