RESEARCH PAPERS
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

2013 Special Number

DEALING WITH MOBILE PLATFORMS FRAGMENTATION PROBLEM: ONTOLOGY ORIENTED APPROACH

Zlatko STAPIĆ

ABSTRACT

The available researches that referred mobile platforms fragmentation problem showed that current solutions, although many, converge into several groups that have common advantages and disadvantages and are based on the same principle: code ones, run anywhere. Thus, the new approach that would enable the developers to use native development environments was necessary. This paper introduces an ontology oriented approach that is based on methodological interoperability. Important assumption of the ongoing research presented in this paper is that during the development process the development team should use the same methodology and the same approach while developing for multiple mobile platforms. This will enhance this process with the artifacts reuse possibility according to the ontologically described knowledge.

KEY WORDS

mobile, development, methodology, fragmentation, interoperability, ontology

INTRODUCTION

One of the obstacles to efficient development of mobile applications is the lack of interoperability between different mobile platforms. The teams involved in developing applications for multiple platforms are faced with having to use platform specific development environments for each platform. This leads to the fact that artifacts created during development for one mobile platform, are not useful when developing for the second or any other target platform. This problem has been recognized by the professional and scientific communities which, over the past a few years, proposed several solutions. Although the proposed solutions are different in their implementation, they are based on a similar concept of developing one source code and its automatic transformation for various mobile platforms ("code once, run everywhere"). The results of these approaches, along with the well-known fragmentation problem (Agarwal et al., 2009) (Manjunatha et al., 2010) (Ridene et al., 2010), limit development teams to use only those features that are implemented in automated code generators or automated and target platform dependent adapters of mobile application.

Zlatko STAPIĆ - University of Zagreb, Faculty of Organization and Informatics, Pavlinska 2, HR – 42000 Varazdin, Croatia, zlatko.stapic@foi.hr

This paper will focus on the analysis of this problem and on proposal of a solution in a domain of methodological interoperability. The idea is to allow developer teams to use native development environments (that is, all their advantages for platform specific mobile application development), by raising the reusability and interoperability to a higher, methodological level. The paper is organized in following sections: at the beginning, the overview of existing solutions is given; the second section provides a brief look into a new ontology oriented approach of defining a methodological interoperability; and finally the conclusion with plans for future work is given.

EXISTING SOLUTIONS

The stated problem of fragmentation of mobile platforms, devices and programming languages has been an issue that the scientific and professional community is dealing with during the last several years. As presented in (Stapić et al., 2012) there are several, rather similar approaches that are being used by many authors. These approaches include the usage of mediatory transform engines, the usage of native application adapters, creation of extensions, APIs and middleware frameworks to existing languages and finally the usage of web technologies and web standards.

The usage of mediatory transform engines implies the process of generating the native code to run on multiple platforms (Maaløe and Wiboe, 2011) and is sometimes called a *cross-compilation* approach. As shown in Fig. 1 the generated native code would be packaged into a platform-specific (native) application package and distributed through usual distribution channels such as application stores. Additionally, this approach is sometimes relying on the usage of a domain specific language (DSL) and in some other cases it uses a cross-compilation between existing well-known programming languages.

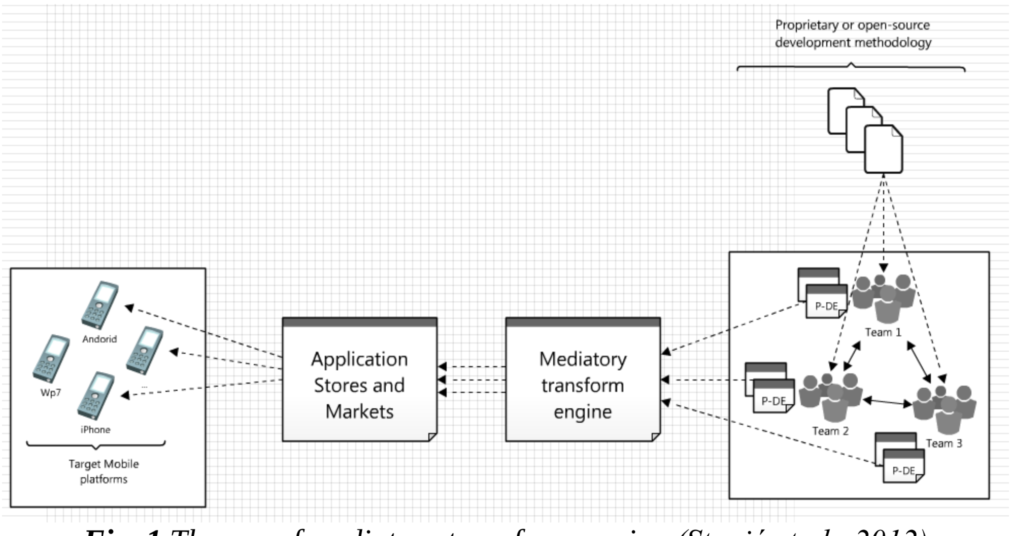
Proprietary or open-source development methodology
Andorid
Wp7
iPhone
Target Mobile platforms
Application Stores and Markets
Mediatory transform engine
P-DE
Team 1
P-DE
Team 2
Team 3
P-DE

Fig. 1 *The use of mediatory transform engine (Stapić et al., 2012)*

There is an imerging number of projects that are based on this approach, and according to Stapić et al. (2012), some of the most influencial projects are MobiCloud (Manjunatha et al., 2010) from Kno.e.sis Research Group (2011), Rhodes from Rhomobile Inc. (2011), Amanquah & Eporwei code generator (Amanquah and Eporwei, 2009) et cetera.

The use of native application adapters is according to Agarwal et al. (2009) one of the two main techniques for handling the fragmentation in which the interfaces calls are wrapped in a distinct modules which are then ported across the platforms. In this scenario (see Fig. 2), optimistically speaking, the interfaces calls in the original source code could be the same for all target platforms, but as argued in (Stapić et al., 2012) this is usually not the case.

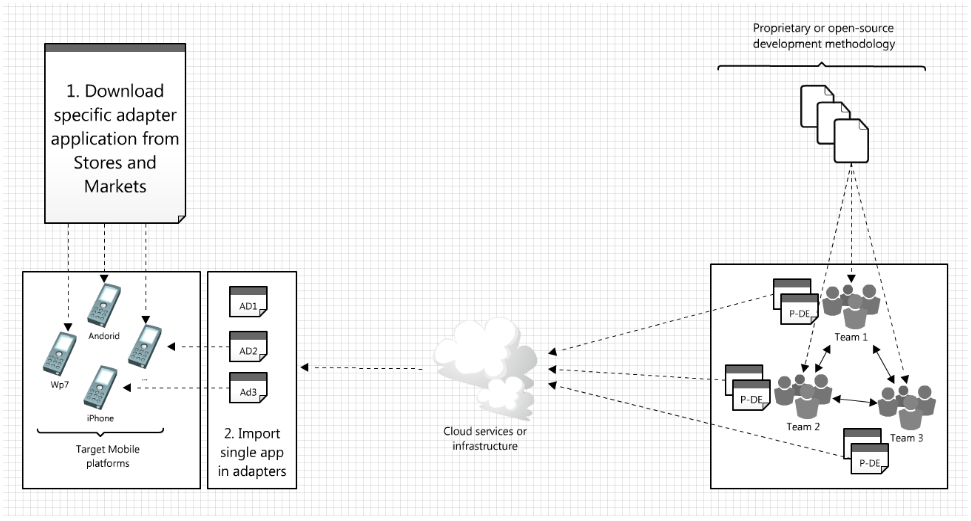
1. Download specific adapter application from Stores and Markets
Andorid
Wp7
iPhone
Target Mobile platforms
AD1
AD2
Ad3
2. Import single app in adapters
Cloud services or infrastructure
Proprietary or open-source development methodology
P-DE
Team 1
P-DE
Team 2
Team 3
P-DE

Fig. 2 *The use of native application adapters (Stapić et al., 2012)*

The influential projects from this group are MobiVine (Agarwal et al., 2009), Corona SDK (Fernandez, 2012), Sencha Touch (Rao, 2012), jQuery mobile (Bai, 2011) and for example PhoneGap which uses HTML5, CSS3 and JavaScript code and wraps it with platform specific adapter capable of calling the native target APIs. On the other side, there are numerous middlewares and frameworks developed in the simillar manner and some of them are presented in (Baloian et al., 2007), (Choi, 2012), (Choi et al., 2009), (Miravet et al., 2009), (Maaløe and Wiboe, 2011) et cetera. These extensions to existing languages aims to improve and make faster the overall multi-platform development process but are faced with the same disadvantages as other previously stated approaches.

As argued and discussed in (Stapić et al., 2012) all mentioned approaches have some advantages and some disadvantages and thus should be used only if project boundaries are clearly defined and state that a particular approach would result in acceptable solution. The conclusion from mentioned authors is that in specific cases none of the mentioned approaches is applicable and that new approaches that will provide the teams developing the multi-platform mobile applications with the full advantages of using the native APIs, the native test environments and the native generators of the executable code are necessary.

ONTOLOGY ORIENTED APPROACH

We can identify at least three dimensions in the space (S) of the possible approaches that could be taken while developing *multi-platform* mobile applications. Those dimensions are defined by methodologies (M), approaches (A) and target platforms (P). Thus, the space can be defined as $S = \{M, A, P\}$ where each dimension can have several instances: $M = \{m_1, m_2, \ldots m_n\}$; $A = \{a_1, a_2, \ldots a_m\}$; $P = \{p_1, p_2, \ldots p_o\}$. Assuming that development team would use SAME methodology and SAME development approach, cardinality of defined sets could be defined as $|M| = 1$; $|A| = 1$; $|P| > 1$, and consequently the cardinality of set S as $|S| = \{(1, 1, n): n > 1\}$. Finally, the development process (DP) could be defined as set of sub-processes (ordered triplets), where each sub-process would result in mobile application developed for specific target platform, as $DP = \{SP_1, SP_2, \ldots SP_n : SP_i \in S; SP_i = (m, a, p_i); 1 < n \leq |P|; i =$

$\{1, 2, \ldots, n\}$; $m \in M$; $a \in A$; $p \in P\}$. The important information given here is that the methodology and approach are NOT going to be changed during the execution of *n* development sub processes.

The research that is currently being performed is constructed by taking all mentioned constraints (of the existing solutions and of the domain defined in previous chapter) into consideration and is aiming to propose a different approach to solve the issues of multi-platform mobile applications development. The idea is to propose a solution based on a methodological interoperability by utilizing the ontologies and relying on previously stated assumption of usage of single methodology and single approach ($|M| = 1$; $|A| = 1$) during the development process.

While talking about interoperability, in this ontology oriented approach, the IEEE definition of interoperability (IEEE Computer Society., 1990) is adopted and extended and the interoperability is considered as “the ability of two or more systems, components, teams or team members to use and exchange the information and methodological artifacts that have been created during the mobile application development process”. Additionally, although there are different types of interoperability, *semantic interoperability* is the knowledge-level interoperability which provides the interoperable systems with possibility to bridge the semantic conflicts (Park and Ram, 2004) and only semantic interoperability is in focus of this approach.

Finally, as ontology is defined as specification of a representational vocabulary for a shared domain of discourse and as it includes definitions of classes, relations, functions and other objects (Gruber, 1993), the ontology is considered as most appropriate tool to describe the mentioned methodological interoperability. Important concept related to ontology mapping of knowledge is *domain ontology* and it can be defined as a network of domain model concepts (topics, knowledge elements) that defines the elements and the semantic relationships between them (Brusilovsky et al., 2005). The use of domain ontologies is suitable to index all content regarding development methodology and this creates the solid bedrock in this approach. In the literature of ontology development, a number of ontology development methodologies (Lumsden et al., 2011) (Corcho et al., 2003) and ontology representation languages like LOOM, OCML or OWL are proposed, but the usage of specific ontology description language or specific ontology development methodology is not required.

CONCLUSION

The results of the efforts of the scientific and professional community to solve the mobile platforms fragmentation problem still have the important drawbacks which makes the paradigm “code ones – run anywhere” useless for mobile application development. The previous papers on this subject as well as short review given in this paper showed that new approach in solving this reusability-interoperability-and-development-efficiency problem is needed and also might be based on enabling the developers the usage of native development environments.

Combining the semantic similarities of artifacts that arise in such development process and taking into consideration the meaning of created artifacts should result in reusability and interoperability enhanced knowledge which is described with proper ontological description. In that way the future work based on this results could end up with an adaptive Web-based system, which will be able to select and recommend the most relevant content during the multi-platform mobile applications development process. Such system would in different aspects significantly improve the mentioned development process.

ACKNOWLEDGMENT

The research presented in this paper was conducted as a part of a PhD research and is partially founded by the scholarship granted to the first author from Croatian Science Foundation.

REFERENCES

1. AGARWAL, V., GOYAL, S., MITTAL, S., MUKHERJEA, S., 2009. MobiVine: a middleware layer to handle fragmentation of platform interfaces for mobile applications. In: *Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*, *Middleware ’09*. Springer-Verlag New York, Inc., New York, NY, USA, pp. 24:1–24:10.
2. AMANQUAH, N., EPORWEI, O.T.. 2009. Rapid application development for mobile terminals. In: *2nd International Conference on Adaptive Science & Technology (ICAST)*. Presented at the Technology (ICAST), Accra, Ghana, pp. 410–417.
3. BAI, G. 2011. *Jquery mobile first look*. Packt Pub Ltd, S.l.
4. BALOIAN, N., ZURITA, G., ANTUNEZ, P., BAYTELMAN, F. 2007. A Flexible, Lightweight Middleware Supporting the Development of Distributed Applications across Platforms, in: Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference On. pp. 92–97.
5. BRUSILOVSKY, P., SOSNOVSKY, S., YUDELSON, M. 2005. Ontology-based Framework for User Model Interoperability in Distributed Learning Environments, in: World Conference on ELearning, E-Learn. AACE, pp. 2851–2855.
6. Choi, M. 2012. A Platform-Independent Smartphone Application Development Framework. Computer Science and Convergence 787–794.
7. CHOI, Y., YANG, J.S., JEONG, J. 2009. Application framework for multi platform mobile application software development. In: *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference On*. pp. 208–213.
8. CORCHO, O., FERNANDEZ-LOPEZ, M., GOMEZ-PEREZ, A. 2003. *Methodologies, tools and languages for building ontologies. Where is their meeting point?* Data & Knowledge Engineering 46, 41–64.
9. FERNANDEZ, M.M. 2012. Corona SDK mobile game development beginner’s guide: create monetized games for iOS and Android with minimum cost and code. Packt Pub., Olton, Birmingham [England].
10. GRUBER, T.R. 1993. A translation approach to portable ontology specifications. *KNOWLEDGE ACQUISITION* 5, 199–220.
11. IEEE Computer Society. 1990. IEEE standard computer dictionary : a compilation of IEEE standard computer glossaries, 610. Institute of Electrical and Electronics Engineers, New York NY USA.
12. Kno.e.sis Research Group, 2011. Welcome to Kno.e.sis [WWW Document]. URL http://knoesis.wright.edu/ (accessed 8.27.11).
13. LUMSDEN, J., HALL, H., CRUICKSHANK, P. 2011. Ontology definition and construction, and epistemological adequacy for systems interoperability: A practitioner analysis. *Journal of Information Science*, 37, 246–253.
14. MAALØE, L., WIBOE, M. 2011. Kamili - A Platform-Independent Framework for Application Development for Smart Phones (Bachelor thesis). Technical University of Denmark, Lyngby, Denmark.
15. MANJUNATHA, A., RANABAHU, A., SHETH, A., THIRUNARAYAN, K. 2010. Power of Clouds in Your Pocket: An Efficient Approach for Cloud Mobile Hybrid

Application Development. In: *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, 496–503.

16. MIRAVET, P., MARÍN, I., ORTÍN, F., RIONDA, A. 2009. DIMAG: a framework for automatic generation of mobile applications for multiple platforms. In: *Proceedings of the 6th International Conference on Mobile Technology, Application &#38; Systems, Mobility '09*. ACM, New York, NY, USA, pp. 23:1–23:8.
17. PARK, J., RAM, S. 2004. Information systems interoperability: What lies beneath? ACM TRANSACTIONS ON INFORMATION SYSTEMS 22, 595–632.
18. RAO, N. 2012. Sencha touch 1.0 mobile JavaScript framework: build web applications for Apple iOS and Google Android touchscreen devices with this first HTML5 mobile framework. Packt Publ., Birmingham; Mumbai.
19. Rhomobile, Inc. 2011. Smartphone Enterprise Application Integration, White paper [WWW Document]. URL http://tiny.cc/rhomobile (accessed 8.20.11).
20. RIDENE, Y., BELLOIR, N., BARBIER, F., COUTURE, N. 2010. A DSML For Mobile Phone Applications Testing, in: Proceedings of 10th Workshop on Domain-Specific Modeling in SPLASH. France.
21. STAPIĆ, Z., de-MARCOS, L., GUTIÉRREZ MARTÍNEZ, J.M. 2012. Approaches in Development of Multi-platform Mobile Applications: State of the Art. In: *Proceedings of IV International Conference on Application of Advanced Information and Communication Technologies. Presented at the ATICA 2012*, Universidad Técnica Particular de Loja, Loja, Ecuador, pp. 429–436.