Recent Advances in Technology and Engineering (RATE-2017)
6th National Conference by TJIT, Bangalore

International Journal of Science, Engineering and Technology
An Open Access Journal

MicroApp: An end-user Environment to Generate Pervasive Mobile Applications

[1]Nadine V Alexandrine, [2]Ramkesh, [3]Nandeesh, [4]Punit Gupta, [5]Dr. D Ramya Dorai

Abstract

We provide a method to allow the client to design the application of their own in their mobile phones by merging the services available in the mobile and of those services provided by the internet. To this aim, we provide a method and graphical annotation which gives freedom to the user to compose their own application. Named MicroApp. Client can develop an application by using a step by step and a repetitive procedure. Client does not worry about the design of user interface. This method mainly deals with the creation of their own applications in their smart phone without making use of a personal computer. The testing results tells us that even though working screen is small in size, the MicroApp generator tool tells us that time management and error management can be done effectively with the use of MIT app Inventor..

Keywords: MicroApp, Pervasive Services, MicroApp Generator etc.

Introduction

Mobile Technology is been advanced and is utilized to the maximum to the benefit of the users. GPS, Sensors, accelerometer, gyro meter and internetworking capabilities are some of the advances. Every application that needs to be built requires programming knowledge and skills, but in this paper we propose a methodology where the user requires zero programming skills to build an app.

Mobile devices are mainly known for three functionalities: the built-in applications, services that are available on the web and applications that can be combined with already developed services on the web. A term, MicroApp Generator is a mobile system that is proposed in this paper, it consists of all the built-in services and the services on the web, through this the user can develop an app and store it in the web. The services for example are services to manage basic user needs like camera and social media, services to manage the user's location like using a GPS, services to manage sensors and temperature based services.

The MicroApp programmer is focused on the app behavior, since the user interface is automatically created for the app starting from the Web Service Description Language (WSDL) [3]. The MicroApp execution can be triggered by various conditions, including the environmental and proximity ones, and gestures.

The MicroApp Generator is compared with a PC based MIT App Inventor , this MIT App Inventor uses the facility of the input devices and wider screens when compared to a smartphone which provides touch operations and sensors as input operations. Though the mapping of a touch to an input device is compared, MicroApp Generator still proves better than an MIT App Inventor in some dimensions. The paper is explained as following Sections:

- Description of the proposed system and the methodology underlying the MicroApp development environment.
- Discussion of the results.
- Analysis of the related work in the field of end-user mobile application development and service composition. Finally, the paper is concluded.

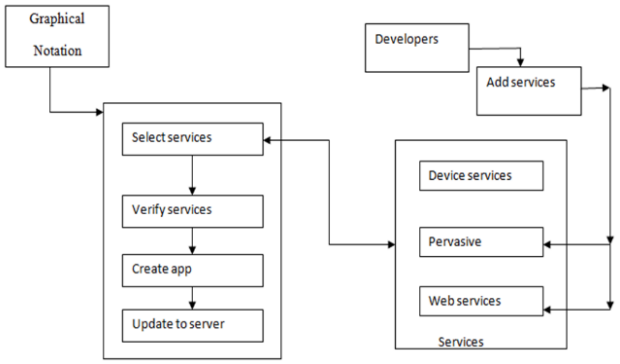

Figure 1: System Architecture

System Design

MicroApp Generator tool enables end-users to interactively compose own applications directly on the mobile device. The user selects services from the repository and builds an app combining the selected services. This tool makes it best for a user in terms of user interactivity and correctness. The architecture and the workflow is described below:

System Architecture

The system architecture follows iterative and incremental process as shown in the figure 1. Here the end-user with the help of graphical notation will have to select the services required for the development of the desired Micro-application. The service selection can be done by selecting the services present on the phone or services can be fetched from the web. Once the selection of services is done it has to be verified whether the selected services match each other and once the services are compatible with each other, the application can be created. After creation of application the user can update the created application on to the server.

On the other side the developers need to keep on updating this application by providing users with new services. The different types of services are device services, pervasive services and web services. The device services are services with which we can access.

The various device sensors such as GPS, proximity sensor etc. The pervasive services are services which are developed and need to be downloaded separately like social networking services such as whatsapp, facebook or the other general services such as alarm, camera etc. The web services are services with which we can link our mobiles with IOT. Again the iterative and incremental process is followed in this architecture.

MicroApp Development Process

The Development of a MicroApp follows an Iterative and Increment method. The MicroApp is developed with a list of services and stored in the Build Application, the user selects those services and generates an .apk file which is saved in the built-in storage of the mobile device.

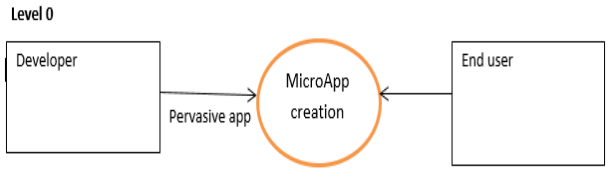

Figure 2: level 0

The Figure 2 shows the interaction between the End-user and the Developer. The Application is seen as a two way approach between the Developer and the End User.

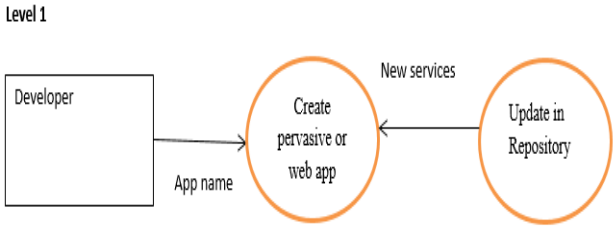

Figure 3: level 1

In figure 3 the Developer develops the predefined services and adds them to a repository each of which is given a unique name and description to be stored in the repository.

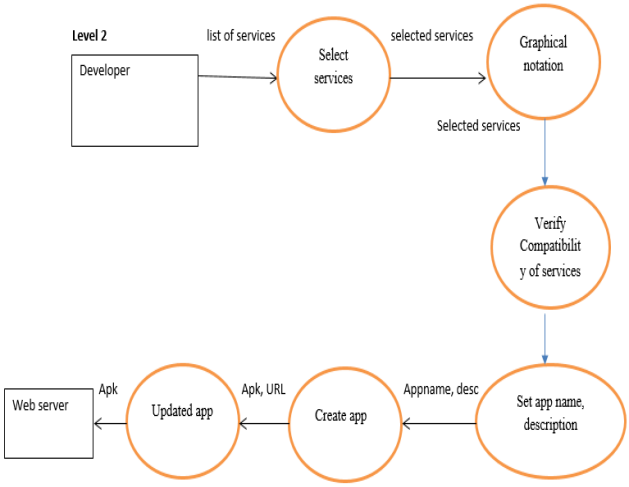

Figure 4: level 2

Figure 4 shows the work flow of the MicroApp in which the End-User selects the services and the selected items are displayed on the Generation Activity. The Items that are selected are checked if they are compatible, if yes the next process is executed else the user is notified of the wrong selection.

Once the selected services are compatible, the user gives a name to the app to be developed and describes it; the apk file is then generated and stored in the built-in storage of the mobile device. On opening the file the app is installed in the mobile device. Once installed, the app is used by the user.

Related Work

MobiOne [5] is a Windows-based tool for creating crossplatform mobile applications which are stored in a cloud repository. The tool creates the GUI of the application by using an emulator on the PC. It then connects to the remote server (App Center Builder) to generate the mobile application which in then loaded on to the PC.

MobiDev allows a user to create and build mobile interface directly on to the mobile device [8].

[7]MIT App Inventor uses a visual programming language that supports various programming constructs and a server is used to generate .apk files.

Danado and Paterno [9], [6] adopts an approach using jigsaw to compose services. They use colour combinations for categorizing data types using the idea presented in [10].

[2], [10], The MicroApp allows the end-user to create customized mobile applications, named MicroApp, which the user can compose directly on to the mobile device. The frame-Work works into two main configuration: model and enactment.

The iCAP system [4] allows the end-users to design context-aware mobile applications where the computers can both sense, and react based on their environment. Devices may have information about the circumstances under which they are able to operate and based on if-then rules.

Table 3 shows the comparison of various tools discussed in this section and their features with respect to their Language, Target user, Development Device, Target Device and the Cloud Service/Remote Server, to the features of the MicroApp Generator.

Tool	Language	Target User	Development Device	Target Device	Cloud Service
MobiOne	TBT	D	PC	SM	Y
MobiDev	Vis	D	SM	SM	N
App Inventor	Vis	EU	PC	SM	Y
iCAP	TB	EU	PC	SM	N
MicroApp	Vis	EU	SM	SM	N
MicroApp Generator	Vis	EU	SM	SM	Y

Table -3: Technological Features of Related Works and Tools.

Conclusion

This paper presents MicroApp Generator, a tool to create apps directly from the mobile device. The services are identified by the name and a graphical notation. This MicroApp Generator introduces a new mobile computing methodology where the users can interact and choose services to build an app, making it a user based app.

The services also list sensors making it grow towards trending technology which was possible only with PC based applications. The End User is facilitated with graphical notation and images making it the best mobile computing scenario.

Acknowledgements

We acknowledge that the content in this paper is prepared by doing a comparison between MIT App Inventor and MicroApp Generator. The MicroApp Generator introduces a new mobile computing scenario for users to access services and creates an app directly on the phone.

References

[1]. Visual Mobile Computing for Mobile End-Users Rita Francese, Michele Risi, Genoveffa Tortora, Senior Member, IEEE, and Maurizio Tucci 2017.

[2] S. Cuccurullo, R. Francese, M. Risi, and G. Tortora, "A visual approach supporting the development of microapps on mobile phones," in Proc. Int. Conf. Distrib. Multimedia Syst., 2011,pp. 171–176.

[3] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," 2001.

[4] A. K. Dey, T. Sohn, S. Streng, and J. Kodama, "iCAP: Interactive prototyping of context-aware applications," in Proc. 4th Int. Conf.Pervasive, 2006, pp. 254–271.

[5] Genuitec. (2014). MobiOne. [Online]. Available: http://www.genuitec.com/mobile

[6] J. Danado and F. Patern _o, "Puzzle: A visual-based environment for end user development in touch-based mobile phones," in Proc. 4th Int. Conf. Human-Centered Softw. Eng., 2012, pp. 199–216.

[7] App Inventor. (2013). MIT Center for Mobile Learning. [Online]. Available: http://appinventor.mit.edu/explore

[8] J. Seifert, B. Pfleging, E. del Carmen Valderrama Baham_ondez,M. Hermes, E. Rukzio, and A. Schmidt, "Mobidev: A tool for creating apps on mobile phones," in Proc. Int. Conf. HCI Mobile Dev. Serv., 2011, pp. 109–112.

[9] J. Danado and F. Patern_o, "A prototype for EUD in touch-based mobile devices," in Proc. IEEE Symp. Vis. Lang. Human-Centric Comput., 2012, pp. 83–86.

[10] S. Cuccurullo, R. Francese, M. Risi, and G. Tortora, "MicroAppsdevelopment on mobile phones," Intl. Symp. End-User Develop.,pp. 289–294, 2011.