Interoperability framework for communication between processes running on different mobile operating systems

A Gal¹, I Filip¹ and F Dragan¹

¹University Politehnica Timisoara, Automation and Applied Informatics Department, Vasile Parvan str., no. 2, 300223 Timisoara, Romania

E-mail: andrei.g88@gmail.com

Abstract. As we live in an era where mobile communication is everywhere around us, the necessity to communicate between the variety of the devices we have available becomes even more of an urge. The major impediment to be able to achieve communication between the available devices is the incompatibility between the operating systems running on these devices. In the present paper we propose a framework that will make possible the ability to inter-operate between processes running on different mobile operating systems. The interoperability process will make use of any communication environment which is made available by the mobile devices where the processes are installed. The communication environment is chosen so as the process is optimal in terms of transferring the data between the mobile devices. The paper defines the architecture of the framework, expanding the functionality and interrelation between modules that make up the framework. For the proof of concept, we propose to use three different mobile operating systems installed on three different types of mobile devices. Depending on the various factors related to the structure of the mobile devices and the data type to be transferred, the framework will establish a data transfer protocol that will be used. The framework automates the interoperability process, user intervention being limited to a simple selection from the options that the framework suggests based on the full analysis of structural and functional elements of the mobile devices used in the process.

1. Introduction

A mobile device is a computational device having a hardware structure similar to that of a personal computer (PC) and also a processing power compared to it. Mobile devices are conceived to be moved very easily from a place to another by one person. Their dimensions are basically compared to that of a human hand. Virtually any device with a hardware structure similar to that of a computer, but much smaller size is considered a mobile device.

In the field of mobile devices manufacturers around the world have noted several manufacturers of such devices such as Nokia, Samsung, Apple, Motorola, BlackBerry. Each of these companies have managed to get noticed by producing mobile devices as sophisticated as they could, meaning as technologically advanced, using both hardware and software equipments to produce next-generation devices.

Very fast technological development has led to the possibility of creating some very technologically advanced mobile devices, giving birth to a new concept, the smart device (smartphone in particular). If from hardware's point of view every company wanted to rapidly improve the structure of the devices they produced, reaching a point where the structure of the devices could be compare to the one of a PC, leading to a "hardware uniformization", from a software point of view things were...
slightly different. If initially each company produced small pieces of software dedicated especially for
the use of phone features, with the development of hardware, companies began to produce complex
software for use of those hardware elements, reaching a point where a mobile device had its own
operating system (OS) installed. The operating systems differed depending on the companies which
produced them, and their use opened the path to the development of OS based applications, thing that
lead to an operational incompatibility between the applications run on different operating systems.
This incompatibility between operating systems was further deepened by alternating dominance of
these systems appeared on the market. Since the development of an application for each operating
system in part involves a complex process that requires time, each company (even ordinary
developers) thought solutions that do not involve communication / interoperability / synchronization
between these applications running on different operating systems, though the applications propose
same functionalities on each operating system.

Thus, with all the applications available on the mobile devices we use every day, it feels the need to
create such a framework that offers the possibility of communicate/ interoperate / synchronize
between applications installed on different mobile operating systems. Though we speak about same
application installed on different operating systems or same application installed on the same
operating system, no matter what the version of the operating system is, the solution is the same. The
framework could handle even different applications noting that the applications interchange data
relevant to each of the applications involved in the process of switching data.

On the market there are currently 3 large dominant mobile operating systems that are installed on
nearly 90% of mobile devices available. Having this in consideration the proof of concept we present
further will be implemented using these 3 large operating systems along with the devices where they
are installed which are: iOS installed on an iPhone 5S, Android OS installed on a Samsung Galaxy S4
and Windows Phone OS installed on a Nokia Lumia 520 [1].

iOS is an operating system developed by Apple Inc. company and it's the second more used mobile
operating system on mobile devices in USA as in the whole world. Apple Inc. being the only company
which distributes iOS on the mobile devices they produce, the company occupies the first place in a
top of most sold mobile devices both in USA as in the whole world [1]. In particular the mobile device
we will use to implement the proof of concept is iPhone 5S which uses iOS 8 as its operating system.
iPhone 5S is a mobile device belonging to the smartphone category which has the following technical
characteristics we are looking for: A7 chip with 64 bit architecture, Wi-Fi module which can use
protocol 802.11 a/b/g/n, Bluetooth 4.0 wireless technology, Assisted GPS and GLONASS (Global
Navigation Satellite System), accelerometer, gyroscope [2].

Android OS is an operating system based on the Linux Kernel and it's developed by Google. Its
user interface, much as like iOS, is using direct manipulation to provide interaction between the user
and the operating system. Taking this in consideration, Android operating system was designed only to
be used by devices such as touchscreens, smartphones, tablets or smart TV's, as this type of devices
offer the possibility to handle the objects provided by the operating system directly [3]. Android OS is
the most used mobile operating system in the world, and being an open source operating system it's
distributed by a lot of companies on the mobile devices they produce [1]. For the proof of concept
further presented, the chosen mobile device is Samsung I9500 Galaxy S4 which uses Android OS
version 4.2.2 and has the following technical specifications: Cortex-A7 chip, Wi-Fi module which can
use protocol 802.11 a/b/g/n/ac, Bluetooth 4.0 wireless technology, Assisted GPS and GLONASS, NFC (Near Field Communication), accelerometer, gyroscope.

Windows Phone OS is an operating system developed by Microsoft exclusively for smartphones. It
is an operating system based on Windows Kernel. Windows Phone OS replaced the first mobile
operating system developed by Microsoft, called Windows Mobile and all the devices that had
Windows Mobile installed are now compatible with Windows Phone. Like both Android and iOS,
Windows Phone is using direct manipulation to let the users interact with the objects it provides. It's
the third used mobile operating system in the whole world after the first two presented above. Though
it's not an open source mobile operating system Windows Phone unlike iOS it's distributed by many
smartphone manufacturers on their devices. We chose to use for implementing the proof of concept the smartphone Nokia Lumia 520 which uses Windows Phone 8 OS and technical specifications like: Snapdragon S4 chip (similar to Cortex-A8 chip) having a dual-core 1.2 GHz, Wi-Fi module which uses protocol 802.11 b/g/n, Bluetooth 4.0 wireless technology, GPS and GLONASS, accelerometer, gyroscope [4].

The technical characteristics of the mobile device on where the operating system is installed present a lot of interest as the framework will make use of them as communication channels with which data is transmitted between applications, thus we have to know the technological limits they have.

2. Framework architecture

The framework is intended for working on the three different mobile operating systems presented above by being installed on each of the mobile operating system. The structure of the framework is represented in the figure below (Figure 1). The core of the framework is represented by the 6 modules that altogether work for offering the solution to communicate easily and optimal between the applications which use it. The architecture of the framework may vary a little bit depending on the mobile operating system where it is installed. The changes in the architectural structure will be made so that the framework is able to use properly the hardware available on the device using that operating system. Any additional modules that may be needed in the implementation could be added as new modules communicating with the core of the framework [5].

![Framework Architecture](image)

**Figure 1. Framework architecture**

2.1. Graphical User Interface module

Given that the framework offers the possibility to the users to communicate/synchronize between processes installed on the mobile device they use, the Graphical User Interface module is the module responsible for making available all the necessary information to the users in terms of selection of processes that will start the communication, the data chosen to be synchronize/switched between the processes, the way in which the communication/synchronization is done (selection of communication environment). The module is situated at the top of the framework's architecture and it communicates either directly or indirectly with all the others module below it.

2.2. Operating system identification module

As the main aim of the framework is to make possible the communication between processes running on different mobile operating systems, a first step in making this possible is the identification of the operating system with which the process has to communicate. In order to identify this, the communicating processes have to switch information that will point to the type of the operating system where they are running, any information related to the version of that operating system in order to have a better understanding in how the communication will be performed being necessary [5].
whole process of switching operating system information will be done by the operating system identification module (Figure 2). In the identification process the module will initiate a handshake type communication that will basically negotiate the rules under which the data will be exchanged between processes involved. The set of rules that the framework will establish will be encapsulated in a protocol for manipulating the data switched between the communicating processes, according to the standards set by the operating systems on which the processes are running. The information gathered by the module will also be used to implement the communication strategy between the processes involved in exchanging the data [6].

![Figure 2. Operating system identification module](image)

### 2.3. Analysis, selection and initialization of communication environment module

As every mobile device used for implementing the proof of concept provides several hardware capabilities that can be used as communication environments, the selection of the optimal environment in each case is a problem that has to be solved in order to have an efficient exchange data process. The framework module responsible for this is the **analysis, selection and initialization of communication environment module** (Figure 3). This module will search hardware communication capabilities of the mobile device to find possible communication environment to be used by processes that use the framework. Each possible communication environment will be examined and analyzed and as a result of the analysis there will be selected the optimal communication environment which will be used by the processes during the data exchange [7].

The analysis will take into account several parameters such as the communication environment data transfer speed, accuracy transfer, the possibility of losing data, the maximum amount of data that can be transferred, possibility of transferring blocks of data that can be sent through this medium of communication. An important thing to mention is that the result of the analysis will provide the optimal communication environment as being the main communication environment between those processes, but this does not mean that other communication environments cannot be used in the exchange process. Even more than that, through this module the framework will select, if needed, based on the analysis of all the processes involved in exchanging the data, a communication environment for each type of data the processes commits to switch between them [8], [9].
2.4. Process communication module

The module is responsible for initializing and performing the communication/data exchange between processes which will use the framework. When initializing the communication process, a new handshake type communication is initiated, and communication rules between processes will be negotiated (Figure 4). The set of rules established will control the format of the data transferred, the communication protocol the processes will use, timeout periods, limitation of bandwidth or limitation of data packages length sent along the communication channels. This module will effectively make the transfer of data from one process to another. An important thing to mention about the module is that it will maintain a live link with the Analysis, selection and initialization of communication environment module (Figure 2) as per having a fast switch from a communication environment to another in case of failure of sending data or reaching an impasse. If the module cannot make the switch to another communication environment, the communication is interrupted without altering the already exchanged data. Maintaining the data integrity has high priority when the communication environment is changed not only when the communication is interrupted [9], [10].

Figure 3. Analysis, selection and initialization of communication environment module

Figure 4. Process communication module
2.5. Automation module
The most important part of the data flow that will be manipulated by the framework will not be manipulated directly by the users from within the processes they are using, as the framework will have an automation module that will automatically manipulate the data so that a user's implication is reduced to accepting/choosing protocols that will be used by the processes when exchanging the data. The automation module will act as a wrapper over the data flow coordinating it from one process to another. The automation part of the framework can be integrated in a separated module of the framework, or it could be added separately in each of the existing modules in the framework, in this particular case creating the automation at the module level. In each case the result will be the same: data flow will be handled automatically by the framework with a minimum of user input. As depicted in Figure 1 the automation module is an independent module of the framework communicating directly with all the modules that deal with data manipulation [9].

2.6. Mobile device investigation module
The idea being that of a framework that offer users the opportunity to switch data between processes they run on different mobile devices, it has to be taken in consideration a way in which the process ran on a mobile device is able to find another device which runs a process with which it can switch relevant data for each of the process. In fact the process itself does not find the mobile device, but the mobile device finds another mobile device. The framework provides a module that handles this type of investigation, a module that will automatically search for all available devices that are able to communicate with it, using the framework and any of the communication environments it has available. The module will both get and provide data relevant for starting to communicate between processes run on mobile devices using the framework [8], [9].

3. Conclusion
In this paper we have presented the proof of concept of a framework that provides the ability to interoperate between processes running on different mobile operating systems installed on smartphones (smart devices) that are different in terms of technical communication capabilities that they offer. Each of the mobile operating systems being different in terms of kernel structure, the framework will have to be installed on each mobile operating system, so that the process running on those operating systems could have access to it. Processes that will use the framework will initiate any communication by using the mobile device investigation module in order to gather information about other available processes.

The framework provides information about the type of the operating system where the partner process is running is by initiating a handshake type communication using the operating system identification module. All the information acquired by this module is passed to all the other module so that a proper data transfer protocol can be established. The data transfer protocol will be composed not only of operating system information but will also be taken in consideration communication environment used, format of the data to be transferred, and technical limits that the devices involved in the data transfer have. As manipulation of data could reach a high level of complexity depending on the processes involved, the framework will use an automation module that will take care of data packages switched between processes so that the user's intervention is reduced to only apply or select from options made available by the framework.

So far the framework is in a proof on concept state but in the future we want to consider a real implementation on each of the three mobile operating systems presented. Of course the proof on concept might suffer structural changes when reaching the implementation level as there might be limitations imposed by the operating systems, and more particular by the companies which developed the operating systems.
References

[1] ***Worldwide Smartphone OS Market Share. http://www.idc.com/prodserv/smartphone-os-market-share.jsp (accessed July 2015)

[2] ***Apple Inc. iOS. http://web.archive.org/web/20071006005308/http://www.apple.com/iphone/features/index.html (accessed July 2015)

[3] ***Google. Android OS. http://www.android.com/history/ (accessed July 2015)

[4] ***Nokia. Windows Phone OS. http://www.windowsphone.com/en-us/phones (accessed July 2015)

[5] Xuan X, Wang Y and Shanping L 2014 Privacy requirements patterns for mobile operating systems, IEEE 4th International Workshop on Requirements Patterns, Karlskrona, Sweden, August 26, pp 39-42

[6] Enck W, Gilbert P, Chun B, Cox L P, Jung J, McDaniel P and Sheth A N 2010 TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones, Proceedings of the USENIX Symposium on Operating Systems Design and Implementation, Vancouver, Canada, October 4-6, pp 99-106

[7] Ahmad M S, Musa N E, Nadarajah R, Hassan R and Othman N E 2013 Comparison between Android and iOS Operating System in terms of security, IEEE 8th International Conference on Information Technology, Kota Samarahan, Malaysia, July 1-4, pp 1-4

[8] Burns J 2008 Developing secure mobile application for Android, ISEC Partners

[9] Lashkari A H and Moradhaseli M 2011, Mobile Operating Systems and Programming: Mobile Communications, VDM Verlag

[10] Enck W, Ongtang M and McDaniel P 2009 On lightweight mobile phone application certification, Proceedings of the 16th ACM Conference on Computer and Communications Security, Chicago, USA, November 9-13, pp 235-245