

## Research Article

# The Concept of the Remote Devices Content Management

Miroslav Behan and Ondrej Krejcar

Department of Information Technologies, Faculty of Informatics and Management, University of Hradec Kralove, Rokitanskeho 62, 50003 Hradec Kralove, Czech Republic

Correspondence should be addressed to Ondrej Krejcar, ondrej.krejcar@remoteworld.net

Received 20 June 2012; Accepted 31 July 2012

Academic Editor: Peter Brida

Copyright © 2012 M. Behan and O. Krejcar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Modern mobile communication devices which are often used as remote access to information systems bring up many advantages for user. Unfortunately in some cases when user has several different mobile devices for the same remote access, a problem of multiplatform and multivendor environment fragments productivity by user knowledge in principals of approach to possible services, controlling or management of devices as well as features is available. The customizable interface of remote device management benefits by the control of all owned, authorized, or publically accessible devices from single point of user perspective. We propose a concept of such information system which takes into account mobile devices and their content. Concept is suitable not only for Apple iOS or Google Android, but also it covers all mobile platforms as well as the sensors capabilities of mobile devices which can turn such mobile Smart Device to Smart and mobile sensor concentrator.

## 1. Introduction

Future resource-based economy will dramatically shape current daily processes in many parts of human activities. The consumer society will face the effective artificial self-responsible subsystems with behaviour where cost-effectiveness and nature-related responsibility would be on first place. Global acquirements of devices power used for developing smart environments, that could dramatically increase human productivity as a side effect, would be recognized as a middle step of oncoming technology evolution. We acknowledged that nowadays the current market with mobile devices is more and more fragmented cross-vendors or platforms where different users approach could be confusing. As an alternative we would present future vision of Remote Device Management (RDM) which could positively shape human productivity and could be simply used for actual and future user centric multidevice environment as a convenient interface. Our Remote Device Management is designed as a productivity concept for users who prefer centralized management point and confident approach to multiplatform types of devices in comprehensive way. The future user content management challenges will have to take into account multidevice environment where

different User Interfaces (UI) and platform are and device-specific features [1].

*1.1. Multidevice Environment.* As a device, we define all the devices which are able to connect to network, that is, to the internet resources using an online or offline mode. The multidevice environment from single person point of view naturally underlines future realistic scenario where user would own or have to manage more than one device. We acknowledge multivendor environment and multiplatform environment as Android [2], iOS, Mango, and so forth. Every common user has currently at least more than one device which would be as an interface to cyberspace or which would be an extension to visualization of electronic world. The scenario would be about connectivity to cyberspace where user prefers conventional way more and more. The basic idea of the multiple-device management is based on simplify user friendly environment, where the same User Interface (UI) is presented for different devices or types of devices from multiple vendors or manufacturers. What could happen when user reclaims the same type of device interface; for instance, mobile phones, where the same functionality and content exist? The user has to know as many device

interfaces as possible many types of platforms exist. What if there exists one customizable device interface which accesses the most common features of different devices. Is that a good experience in evaluation of human productivity? What if user could be independent on platforms and type of devices and in case of device crash or device lost, it could be easily recovered by one button click? Even more when user realizes that there is a possibility to manipulate content of different devices which is accessible from single interface. Of course it is all about capability of devices which could in the future lead more and more to massive usage of smart solutions and could make mobile device as natural connection between human and groups of devices (e.g., car, fridge, or boat).

*1.2. Mobile Device Apple Platform.* The Apple platform provides for developers fundamental and well-prepared design support with framework named COCOA, which is basically using Object-C as programming language. There are other extensions from point of developer view where Java or other scripting languages could be used. The mobile devices are used as operation systems the iOS and the most convenient way for developing an application is at using common system calls as application interfaces, application services, and core services [3]. The advantages of the Apple platform are basically comprehensive, publishable, and distributive application channels over Internet.

Another plus of this platform architecture [4] is one-vendor device based on a solution where the certainty of proper system calls and their behaviour is well defined and supported. As well as device hardware access in terms of mobile device development the screen resolution where as ratio between height and width constantly 5:3 could be announced another beneficial aspect in Rapid Application Development (RAD). Apple platform establish fundamentals of mobile application ecosystem environment. The increase of usability of mobile device is enormous. Identity of application is consisting of small image and short-term expression with remote update framework possibility known as an application market.

*1.3. Mobile Device Android Platform.* The mobile platform as Android is due to self-interopen ability suitable for the 3rd party solutions where partial problems are solved [5]. The security and stability of the system which is based on an Open Source concept is outstanding [6]. A device types which are currently running under Android platform are well-known for smartphones and tablets but also for other device types [7] due to suitability of platform design for, instance laptop, netbooks, smart books, e-book readers, smart TVs, wristwatches, headphones, car players, smart glasses, vehicle navigating systems, refrigerators, home automation systems, games consoles, mirrors, cameras, or portable media players. The architecture of Java-based platform fully provides multi-threading environment where gathering of precise data form sensors are required. The architecture [8] allows services which are running on background as a provider or as a consumer of external services [9].

## 2. Problem Definition of Remote Device Management

In this chapter we summarized problematic areas in device management in consideration of possible remote use. We focused on cross-device features which are mainly based upon the management that provides measurement, controlling, and maintenance over sensors or content of device. Other point of view would consider the platform aspects which are supportive to some key benefits in the Remote Device Management (RDM). At last the focus would consider the network access and its capabilities in terms of usability and sustainable processing. For better overview we outline ideas expressed in the mind map on the following figure (Figure 1) which are described in more details bellow in this chapter.

The aim of Remote Device Management is about to provide consistent cloud service where key benefits are open framework accessibility with simple to use Application Program Interface (API), social connectivity, content consistency, and security policies. Let starts with identity connector (Figure 1) problematic which is bound by correct user identification over native account provided and where all user data are mounted on.

*2.1. Identity Problematic.* The identity is nowadays spread over web application mostly and therefore we include into account the main of them which are divided into Social Networks, Emails, Mobile, and Desktop devices identity providers. The Social Networks are considered only the main ones such as Facebook, G+, Twitter, and LinkedIn. The identity from most of them is provided over inner-defined application which after user authorization by Open Access (OAuth 2.0) process acquired user permission on specific tasks and is able to process user's information and establish secure 3rd side authentication for login session. The other way of user identification used world widely is email authorization where over the provided user email in registration process is authorization link sent and after user activation we assume the email is correct as communication channel or current browser is authorized for login session temporary or permanently. Another case is about mobile devices where the authorization channels are SMS, Voice Call, or Native Application. The identity acquired is Mobile Number or Device Identity where each provides authorized secure channel. The last one case is desktop which uses for access browser or native client in terms of specific platform (iOS, Windows, Linux), and where user identity is acquired over registration process with or without any parts mentioned before.

*2.2. Content Problematic.* The core feature of Remote Device Management is multiply, sync, or backup content over different types of devices. Content is any kind of information related to user and in terms of device management, we consider content as an end-user data which are important to keep safe on devices due to daily usage even in offline mode of the device connectivity and also accessible from

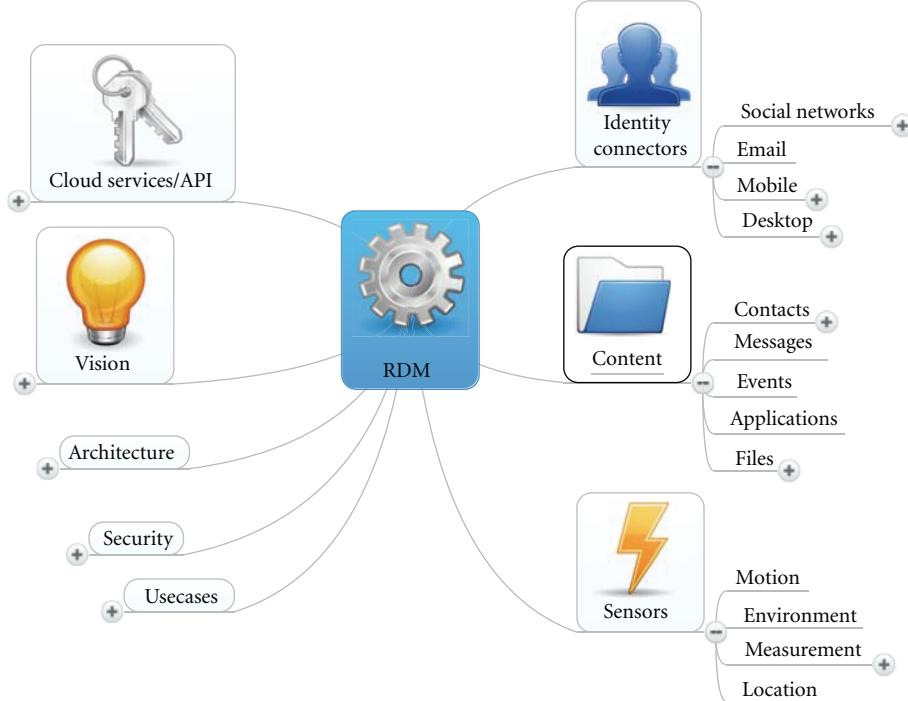


FIGURE 1: Mind map-Remote Device Management.

different mobile or desktop devices. As representative types of user content we recognized Contacts, Messages, Events, Applications, and Files. These kinds of content would be smart distributed and synchronized over user owned devices or be backed up automatically in terms of lost or broke accident. Management of content would be more convenient on desktop device rather than small screen mobile device. On the other hand in time or in location tasks are required event from nonconforming user interfaces for specific tasks of content management. Contacts are very specific content which requires correct personal identification and actual communication channels availability resolving therefore supportive part is Social Networks, Email Contacts and Phone Contact list in terms of distributive social knowledge. Messages are consists of SMS, Voice mail or IMs which provides open access to 3rd sides. Also the problematic is mounted by personal identification based on contacts. Events represents all activity provided by device or user action. Applications are content which in mobile point of view is mostly provided over platforms dedicated applications web or desktop (iTunes-iOS, Play-Android, MarketPlace-Windows Phone) therefore at least the list of application resources with time spent monitoring is considerable for sync. The last one kind of content we assume as files where the requirements are limited to amount of date and share purposes. The file content is achieved by implemented Camcorder, Camera, Microphone, or Universal Serial Bus (USB) for other types file content. The main reason for defined file content is to share further postprocessing on desktop.

**2.3. Sensor Problematic.** As the sensors we define all measurable informational providers of devices which are accessible and are enabled for gathering motion or environment data such as wireless signals, sensorial data, or long-term indicators. There are more types of sensors and therefore we divide them into Motional, Environmental, and Locational-based sensors [1, 10–12]. The motional types are connected to movements of device or its position in space for instance Accelerometer, Magnetic field, or Gyroscope sensor (Table 1).

The environmental types are associated with conditions of current location such as pressure or temperature and also knowledge-based measurable information which are gathered over long-term time period where for instance the home environment is recognizable as the most occupied place of device where the sleeping time of user is recognized by motion or microphone sensor inactivity [13, 14].

And the location-based sensors we consider as Global Positioning System (GPS) sensor, Global System for Mobile communication (GSM) signal sensor, or Wireless Local Area Networks (WLAN) signal receiver which in cooperation establish precise measurement of device location accordingly to energy efficiency. The way of gathering sensor data is over allowed platform system calls where access is authorized by an end-user or by device management provided for platform or by overriding manufactured firmware within dedicated customized distribution of opened platforms which would be available as an Open Source. We are focusing on allowed 3rd party sensor access which is available through framework API, for example, in Android (Table 1).

TABLE 1: Android sensors [18] for smart and ambient environments—overview.

Type	Functionality
Accelerometer	Sensor calculate acceleration without gravity acceleration $g = 9.81 \text{ m/s}^2$ as following equitation: $Ad = -g - \sum F/\text{mass}$ <p>The reason is user centric-suspected behaviour for an end-user where device in a stable position placed on the table should have acceleration 0 instead of +9.81</p>
Ambient temperature	Ambient (room) temperature in degree Celsius
Gravity	A three-dimensional vector indicating the direction and magnitude of gravity. Units are $\text{m/s}^2$ . The coordinate system is the same as is used by the acceleration sensor
Gyroscope	Values are in radians/second and measure the rate of rotation around the device's local X-, Y-, and Z-axis
Light	Ambient light level in SI lux units
Linear acceleration	A three-dimensional vector indicating acceleration along each device axis, not including gravity. All values have units of $\text{m/s}^2$ . The coordinate system is the same as is used by the acceleration sensor. The output of the accelerometer, gravity and linear-acceleration sensors must obey the following relation: $\text{acceleration} = \text{gravity} + \text{linear-acceleration}$
Magnetic field	Values are in micro-Tesla ( $\mu\text{T}$ ) and measure the ambient magnetic field in the X-, Y-, and Z-axis
Orientation	All values are angles in degrees
Pressure	Atmospheric pressure in hPa (millibar)
Proximity	Proximity sensor distance measured in centimeters
Relative humidity	Relative ambient air humidity in percent
Rotation vector	The rotation vector represents the orientation of the device as a combination of an angle and an axis, in which the device has rotated through an angle $\theta$ around an axis $\langle x, y, z \rangle$

The measurement requires at least separate thread to perform precise measured result therefore architecture suites to producer and consumer concept. In case of remote consumer the results would not be influenced by dilation of time of transport or transaction. With consideration of network latency the result would be notified or expected in correct time form. We recognized two groups of sensors where one of them is real-time changed and the other one consists of state long-time sensor changes.

The following article [15], where the domain of sensors data gathering is well defined, was considered as contributory for our solution design. The informational system as extension would provide user status resolution over sets of gathered sensor data, where sleeping, sitting, running, walking, or driving have informational value in current context point of view. Also the environment context is valuable in terms of user productivity for instance vacation, work, or distance movement.

**2.4. Universal Platform Approach.** The most significant aspect which influences usability of mobile devices is platform based solution. All positive and negative user experience leads to platform evolution where useless platforms are terminated. Therefore platform survival dependents on scalability and open mind accessibility where openness to new solutions and approach both technical and future ideas predetermine the platform success. The accesses to device features as well as technical capabilities which are not user invasive are essential to be included in our universal solution where native client on device is realized as connector between user and platform.

### 3. Solution Design Concept

This paper describes solution concept of remote device management focused mainly on server side architecture. Designed concept is variable in terms of technology use case. We suppose to use as development framework all Java based technology because of effectiveness in productivity, scalability and reuse of available Open Source components. The following figure (Figure 2) highlights important parts of architecture which are required for specific needs especially from network connectivity characteristics [16]. The informational system consists of three main parts which are remote client part, core system part and end-user interface part. We start to describe remote part where all possible devices potentially could be connected to the system. The devices are highly fragmented hardware area due to competition of manufactures and vendors about end-user's goodwill. We recognize basically two sets of devices from system point of view.

The first contains all mobile devices which have to care about power supply management and without power-saving management the unnecessarily draining battery would lead to uselessness of developed application. The second group of devices is an independent of power supply where for instance we classify cars because of their external power supply. For these reasons each group would behave differently in terms of kind of connectivity mode. The devices which could be connected to the system over any kind of network and could provide peer-peer internet connectivity instantly or for exact amount of time we called as active devices. The others we called as passive where which would not be all time online or connected to a dedicated server. The passive mode would respect user's defined network connectivity due to cost effectiveness or power management. Next figure

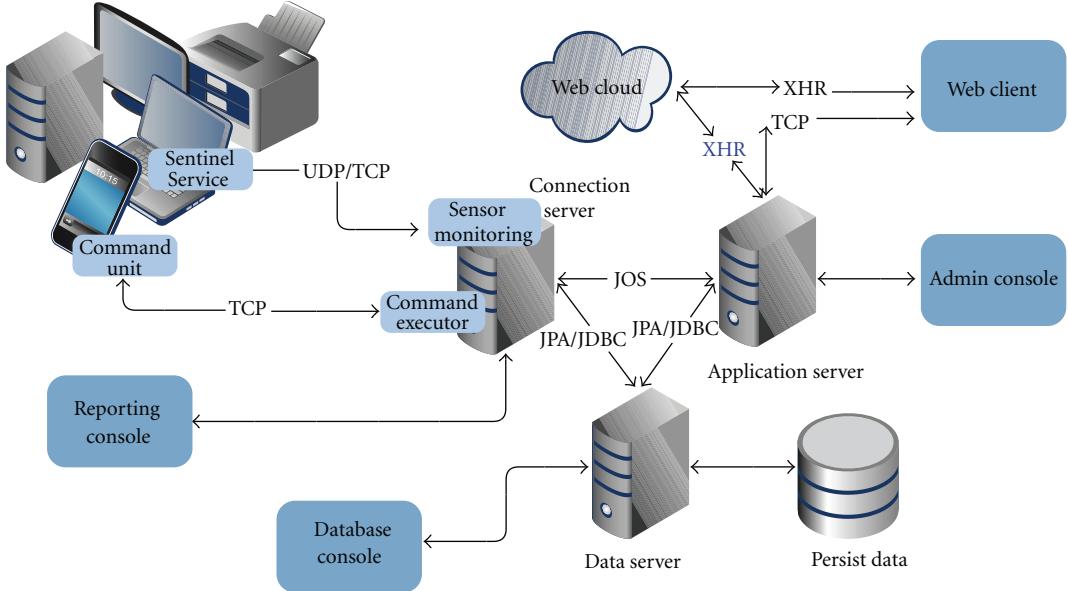


FIGURE 2: System architecture-Remote Device Management (using gliffy UML).

(see Table 2) overviewed connectivity of devices and their statuses according to an active or passive device policy.

Explicitly defined states of connectivity would not get rid of some cases where the uncertain behaviour of network connectivity could appear. The problem would be eliminated by control queuing management based on messages. The messages of events and data would be handled on a client side as well as on a server side in correct time frame. In case of network unavailability the queue substitute consumer and when network status changed and reconnects the message queue processed all First In First Out (FIFO) messages in time order. The sentinel measurement data which are with low level importance and are not supposed for real-time processing are lastly consumed. From technical point of view the messages are sent from client to server by User Datagram Protocol (UDP) in such low level importance cases and where high measurement precision is not required but is preferred speed of delivery or amount of transmitted data. Otherwise the Transport Control Protocol (TCP) is used for confident types of measurement or data delivery of content [17]. The connection server consists of datagram resolver and socket transport object resolver where socket resolver due to usage with sensor monitoring and also with command executor is practically core functionality of connection server. The socket resolver either being client or server is providing communication between device and system. The main responsibilities are maintenance of connection and transportation of data objects between both sides. Objects are transported over serialization Java technology where content is fast serialized or de-serialize on binary code and sent over network.

The second part called as system part basically handles core features of informational system from device monitoring, device command execution and device management with predefined device or user policies up to the system

external data storing, system content providing, and system authorized access ability for all kinds of requests from front or background processing. Also as a connection server we would use Java programming language for implementation due to object interserver exchange. The application server and client application service decides what and when would be transported. Device execution commands are initiated also by application server where are authorizations of requests dispatched from an end-user actions or device routines with associated permissions. Application server is responsible for all other types of requests from the device or the end-user. The calls consist of group with data visualization calls, group of background routine calls based on the time triggering settings and maintenance group or administration group of calls. All calls are related to specific tasks or concrete device or group of devices therefore also security and authorizations are part of responsibility of application server. Last component is data management of informational system based on Java Persistence Application Programming Interface (JPA) technology which is being used due to extraordinary developing capability and time development saving. The data objects are defined in Java classes and relations between entities are expressed as a member of concrete class with specific annotation which specific cardinality and type of relation. The objects are transformed to database through persistent commands and after the commitment are saved to hard file on disk.

The last part of the system is focused on web content delivery and interactions with an end-user. The web content is hosted on external web server as a cloud solution where the user identity and cloud services could be used. The web client is connected to the web cloud services either to the application server over secure channel. Web cloud is used due to implicit network traffic monitoring tool and cost-effective load balancing for web clients with minimal

TABLE 2: Table of connection modes.

Device mode	Connectivity/status	Initiator	Purpose
Active	Persistent/live	Server	After client registration server establishes peer-peer connection for live command requests channel triggered by web user interface Client sentinels could send event change statuses
Active	Interval/command routine	Server/client	In defined interval on server side based on configuration or policy system connects and performs commands on background Client sends interval defined sentinel updates
Active	Zero-base/online	Server	Heart beat protocol for devices in active mode to maintain stable persistent connection
Active/Passive	Session/online	Server/client	Requested commands related to established session by web user interface where in passive mode explicit authorization is required Registration request of client with meta data for system connection Batch updates of sentinels in passive mode
Active/Passive	Not available/offline	Server/client	Connection with client or server is lost or could not be established. Data or tasks are queuing and waiting for the connection to be established

impact on the maintenance. Web cloud is basically used as secured fast traffic response container for web client which mainly communicates with application server in global world scale where continental redistribution is a case. Web client itself implements data visualization and requests posting and corresponding response handling. The client is based on HyperText Markup Language (HTML) version 5 and JavaScript (JS) concept. Communication with an application server is performed over Transmission Control Protocol (TCP) by Web Socket technology due to convenient and fast responsive way in comparisons to classical Asynchronous JavaScript and Xml (AJAX) technology. The Web Socket technology provides persist communication channel over a well-known port 80 with advantages of socket connectivity. Therefore the reaction time of committed commands in live online mode increases usability of the entire system where round trip time (RTT) to server is multiple times faster than common Xml Http Request (XHR) for short messages.

#### 4. Conclusions

Over the current multiple device environmental interfaces and subsidized functionality we would decrease human time spent with maintenance of an authorized, owned or public group of devices in terms of content, settings, policy or gathering sensor information and to increase human productivity accordingly to comprehensive user's interface for multiple devices and within comfortable working space. The fragmentation of devices is more increasing in time due to the technological innovations and therefore Remote Content Device Management would be convenient also for nongeek personalities. The advantages of proposed system concept are single user interface customizable by user-centric behaviour undependable on platforms or vendors settings and recovery optionality over uniform or nonuniform devices with a user synchronized content delivery.

#### Acknowledgments

The work and the contribution were partially supported by the Project (1) "SMEW-Smart Environments at Workplaces," the Grant Agency of the Czech Republic, GACR P403/10/1310; (2) "Smart Solutions in Ambient Intelligent Environments," University of Hradec Kralove under the Project SP/2012/6.

#### References

- [1] P. Mikulecky, "Remarks on ubiquitous intelligent supportive spaces," in *Proceedings of the 15th American Conference on Applied Mathematics/International Conference on Computational and Information Science*, pp. 523–528, University of Houston, Houston, Tex, USA, 2009.
- [2] A. Loukas, D. Damopoulos, S. A. Menesidou, M. E. Skarkala, G. Kambourakis, and S. Gritzalis, "MILC: a secure and privacy-preserving mobile instant locator with chatting," *Information Systems Frontiers*, vol. 14, no. 3, pp. 481–497, 2012.
- [3] S. P. Hall and E. Anderson, "Operating systems for mobile computing," *Journal of Computing Sciences in Colleges*, vol. 25, no. 2, pp. 64–71, 2009.
- [4] Apple Developer Site, iOS, <http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/IPhoneOSOverview/IPhoneOSOverview.html>.
- [5] M. L. Murphy, *Android Programming Tutorials*, CommonsWare, 2009.
- [6] T. Yamakami, "Foundation-based mobile platform software engineering: implications to convergence to open source software," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human (ICIS'09)*, pp. 206–211, November 2009.
- [7] D. Vybiral, M. Augustynek, and M. Penhaker, "Devices for position detection," *Journal of Vibroengineering*, vol. 13, no. 3, pp. 531–535, 2011.
- [8] J. P. Espada, R. G. Crespo, O. S. Martínez, B. Cristina Pelayo G-Bustelo, and J. M. C. Lovelle, "Extensible architecture for context-aware mobile web applications," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9686–9694, 2012.

- [9] N. Radia, Y. Zhang, M. Tatipamula, and V. K. Madisetti, "Next-generation applications on cellular networks: trends, challenges, and solutions," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 841–854.
- [10] P. Brida, J. Machaj, J. Benikovsky, and J. Duha, "An experimental evaluation of AGA algorithm for RSS positioning in GSM networks," *Elektronika ir Elektrotechnika*, no. 8, pp. 113–118, 2010.
- [11] N. Chilamkurti, S. Zeadally, A. Jamalipour, and S. K. Das, "Enabling wireless technologies for green pervasive computing," *Eurasip Journal on Wireless Communications and Networking*, vol. 2009, Article ID 230912, 2 pages, 2009.
- [12] N. Chilamkurti, S. Zeadally, and F. Mentiplay, "Green networking for major components of information communication technology systems," *Eurasip Journal on Wireless Communications and Networking*, vol. 2009, Article ID 656785, 7 pages, 2009.
- [13] R. Brad, "Satellite image enhancement by controlled statistical differentiation," in *Innovations and Advances Techniques in Systems, Computing Sciences and Software Engineering*, Proceedings of the International Conference on Systems, Computing Science and Software Engineering, ELECTR NETWORK, pp. 32–36, 2007.
- [14] O. Krejcar, J. Jirka, and D. Janckulik, "Use of mobile phones as intelligent sensors for sound input analysis and sleep state detection," *Sensors*, vol. 11, no. 6, pp. 6037–6055, 2011.
- [15] G. G. Blazquez, A. Berlanga, and J. M. Molina, "InContexto: multisensor architecture to obtain people context from smartphones," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 758789, 15 pages, 2012.
- [16] S. Conder and L. Darcey, *Android Wireless Application Development*, Addison-Wesley, 2009.
- [17] V. Kasik, M. Penhaker, V. Novák, R. Bridzik, and J. Krawiec, "User interactive biomedical data web services application," *Communications in Computer and Information Science*, vol. 171, pp. 223–237, 2011.
- [18] Android Developer Site, Sensors, <http://developer.android.com/reference/android/hardware/Sensor.html>.

