STATE UNIVERSITY OF MARINGÁ
CENTER OF TECHNOLOGY
DEPARTMENT OF INFORMATICS
GRADUATE PROGRAM IN COMPUTER SCIENCE

LUIS GUSTAVO ARAUJO RODRIGUEZ

Layered architecture and performance metrics for mobile systems: a proposal

Maringá

2017

LUIS GUSTAVO ARAUJO RODRIGUEZ

Layered architecture and performance metrics for mobile systems: a proposal

Dissertation submitted to the Graduate Program in Computer Science of the Department of Informatics, Center of Technology at the State University of Maringá, as a partial requirement for obtaining the Master Degree in Computer Science.

Advisor: Prof. Dr. Anderson Faustino da Silva

Maringá
2017

APPROVAL SHEET

LUIS GUSTAVO ARAUJO RODRIGUEZ

Layered architecture and performance metrics for mobile systems: an proposal

Dissertation submitted to the Graduate Program in Computer Science of the Department of Informatics, Center of Technology at the State University of Maringá, as a partial requirement for obtaining the Master degree in Computer Science by the examination board composed of the following members:

EXAMINATION BOARD

Prof. Dr. Anderson Faustino da Silva
State University of Maringá – DIN/UEM

Prof. Dr. Nardênio Almeida Martins
State University of Maringá – DIN/UEM

Prof. Ph.D. Claudio Luis de Amorim
Federal University of Rio de Janeiro – COPPE/UFRJ

Approved on: February 15, 2017.
Place of defense: Room 101, Block C56, *campus* of the State University of Maringá.

Dados Internacionais de Catalogação na Publicação (CIP)
(Biblioteca Central - UEM, Maringá, PR, Brasil)

Rodriguez, Luis Gustavo Araujo
R696L Layered architecture and performance metrics for mobile systems: a proposal / Luis Gustavo Araujo Rodriguez. -- Maringá, 2017.
80 f. : il. color., figs., tabs.

Advisor: Prof. Dr. Anderson Faustino da Silva.
Dissertation (Master Degree) - State University of Maringá, Center of Technology, Department of Informatics, Graduate Program in Computer Science, 2017.

1. Mobile Systems. 2. Computer Architectures. 3. Mobile Computing. 4. Precision Beekeeping. 5. Sistemas móveis. 6. Arquitetura de computadores. 7. Computação móvel. 8. Apicultura de precisão. I. Silva, Anderson Faustino da, advisor. II. State University of Maringá. Center of Technology. Department of Informatics. Graduate Program in Computer Science. III. Title.

CDD 23.ed. 004.2
AMMA-003413

ACKNOWLEDGEMENTS

First and foremost, I thank God for giving me the opportunity to gain more knowledge at the State University of Maringá.

I thank my family for giving me strength when I had difficulties in the Master's program. For all their sacrifices in order for me to succeed and be someone in life.

I thank my friends, David Posso, Denise Dalcol, Jesus Encinas, João Fabrício Filho, Josue Sousa and Yessica Acosta, for their unconditional support in the last two years. For always wanting the best for me and remaining by my side during difficult times.

I thank my teachers for their patience and knowledge during the graduate courses. For always helping me when I had doubts about a certain topic.

I thank my Portuguese teacher, Dr. Elenir Voi Xavier, for caring about me. Thank you for always being there and giving me your friendship. I will forever value our conversations and classes together.

I deeply thank my teacher and advisor, Dr. Anderson Faustino da Silva, for the advice, support and believing in me. Thank you for always making me feel as a part of the team. It was an amazing experience working together and I hope it is not the last time.

Finally, I thank the Organization of the American States, Coimbra Group of Brazilian Universities, State University of Maringá and CNPq for giving me the opportunity to study a Master's course in Brazil. Thank you for making this phase one of the best and unforgettable experiences that I will ever have.

Thank you to all those people who always supported me and stood by my side. You made me move forward, do my best and have the motivation and strength to practice my profession properly. I will always be grateful for everything you did for me. Thank you for being in my life and letting me be a part of yours. I am eternally grateful with everyone.

Layered architecture and performance metrics for mobile systems: a proposal

ABSTRACT

Mobile Computing is evolving into an important and indispensable area. In the forthcoming years, it is expected that the number of people interacting with mobile systems will increase significantly. A sector where Mobile Computing is applied is *Precision Beekeeping* (PB), which can be defined as the process of monitoring bees in their hives in order to detect the state of the colony and prevent undesirable events. This is done through mobile systems, which provide parameters such as temperature or humidity to learn more about the conditions of the colony. However, the development of these applications is complex due to them running in various heterogeneous environments. Furthermore, technologies and network protocols are constantly being updated, which is why the Mobile Computing environment is characterized with variability. The objective of this project is to propose a layered architecture that would ease the development and maximize the lifespan of mobile systems. The layers of the proposed infrastructure are: Service Layer, Network Layer and Hardware Layer. In addition, performance metrics are proposed in order to evaluate mobile systems and their infrastructures. The expected contributions of this research include a *flexible* architecture to simplify the development and maintenance of mobile systems, and performance metrics, which will support validating portable applications. Furthermore, it is expected that this project will support the growth and evolution of Mobile Computing and PB, and serve as a basis for future developments of mobile systems.

Keywords: Mobile Systems. Computer Architectures. Mobile Computing. Precision Beekeeping.

Arquitetura baseada em camadas e métricas de desempenho para sistemas móveis: uma proposta

RESUMO

A Computação Móvel está evoluindo numa área importante e indispensável. Nos próximos anos espera-se que o número de pessoas interagindo com sistemas móveis aumente de uma forma considerável. Um setor onde é aplicada a Computação Móvel é *Apicultura de Precisão* (AP), a qual pode ser definida como o monitoramento das abelhas em colméias para identificar o estado da colônia e assim prever eventos indesejáveis. Isso é feito por meio de sistemas móveis, os quais fornecem parâmetros tais como a temperatura ou umidade para conhecer sobre as condições da colônia. O desenvolvimento de aplicações portáveis é complexo porque são executadas em ambientes heterogêneos. Além disso, as tecnologias e protocolos de rede são atualizados constantemente pelo qual o entorno da Computação Móvel é caracterizado por mudanças. Os objetivos deste trabalho são propor uma arquitetura flexível baseada em camadas que permita facilitar o desenvolvimento e maximizar a vida útil de sistemas móveis, e propor métricas de desempenho para avaliar sistemas portáveis. As camadas da infraestrutura proposta são: Camada de Negócio, Camada de Rede e Camada de Hardware. As contribuições esperadas deste trabalho incluem: uma infraestrutura *flexível*, a qual permita simplificar o desenvolvimento, atualização e manutenção de sistemas móveis, e métricas de desempenho para avaliar sistemas portáveis. Além disso, espera-se apoiar no crescimento e evolução da Computação Móvel e AP, e ser uma base para futuros desenvolvimentos de sistemas móveis.

Palavras-chave: Sistemas Móveis. Arquitetura de Computadores. Computação Móvel. Apicultura de Precisão.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

3G: Third Generation
4G: Fourth Generation
5G: Fifth Generation
CASHE: Content Adaptive System for Heterogeneous Environments
CDPD: Cellular Digital Packet Data
EDGE: Enhanced Data for Global Evolution
EFI: Experimental Farm of Iguatemi
GPRS: General Packet Radio Service
GSM: Global System for Mobile Communications
HTML5: Hyper Text Markup Language 5
IDE: Integrated Development Environment
IT: Information Technology
LAN: Local Area Network
LTE: Long Term Evolution
PB: Precision Beekeeping
PC: Personal Computer
PCS: Personal Communication Systems
PDAs: Personal Digital Assistants
SDR: Software Defined Radios
SOA: Service Oriented Architecture
TV: Television
UEM: State University of Maringá
UFRJ: Federal University of Rio de Janeiro
VPN: Virtual Private Network
WiFi: Wireless Internet
WLAN: Wireless Local Area Network
WSN: Wireless Sensor Network

TABLE OF CONTENTS

1 Introduction

Nowadays, Mobile Computing is widely utilized in several fields, including Agriculture. Mobile devices are constantly increasing and offering modern services due to emerging technologies.

Such gadgets are utilized in a wide range of scenarios including *Precision Beekeeping* (PB). PB refers to the process of monitoring bees in their hives, utilizing *Information Technology* (IT) tools, such as temperature or humidity sensors, microphones and/or infrared cameras. The use of such devices is done in order to accurately maintain bee colonies. Bees are indispensable because of their ability to pollinate, providing nutrients for humans and animals, as well as aiding in ecosystem health.

A mobile system called myBee, developed by the *State University of Maringá* (UEM) and *Federal University of Rio de Janeiro* (UFRJ), was implemented to monitor and control the internal temperature of hives belonging to the *Apis Mellifera* species [114]. myBee utilizes a two-layered architecture to manage services and hardware components.

Based on this system, it was acknowledged that the development of mobile systems is complex because they are executed in heterogeneous environments. Furthermore, network technologies and protocols are constantly being updated, which is why Mobile Computing is evolving considerably. Another challenge is maximizing the lifespan of a mobile system, which can be defined as the period when efficiency and availability are at their maximum capacity.

1.1 Objectives and Contributions

Since `myBee` is considered a primary source of motivation for this master's project, the objectives are to:

- Propose a flexible and layered architecture that allows for simplicity in developing and maximizing the lifespan of any mobile system; and
- Propose performance metrics in order to evaluate mobile systems and its infrastructures.

The expected contributions of this work include: a *flexible* architecture, that provides simplicity in developing, updating and maintaining mobile systems; and performance metrics to evaluate a portable application. In addition, `myBee` was documented in order to better comprehend it. Thus, this master's project offers technical and scientific contributions. Furthermore, this thesis will support the growth and evolution of Mobile Computing and `PB`, as well as be a basis for future developments of portable systems.

1.2 Text Structure

The text is organized as follows. Chapter 2 describes general concepts related to mobile systems and architectures. Chapter 3 details about `PB`. Chapter 4 describes `myBee`, the experiments that were executed and works related to `PB` systems. Chapter 5 presents the proposals, which consist of a layered-based architecture and performance metrics for any type of mobile system. Chapter 6 describes related works in regards to mobile and/or monitoring systems, architectures and metrics. Finally, Chapter 7 establishes the conclusions of this project.

2

Mobile Systems

Over the last few years, desktop computing has seen a major reduction, in terms of usage, due to portable systems and gadgets. Mobile computing can be defined as information access via non-stationary devices [63, 89]. Portable systems are easily operable from any location and typically use wireless connections to perform their operations [89].

Figure 2.1 displays, in more detail, the concept of mobile systems. In this figure, the user is manipulating a mobile device. These electronic gadgets are connected, via a network, to access data that is requested by the user.

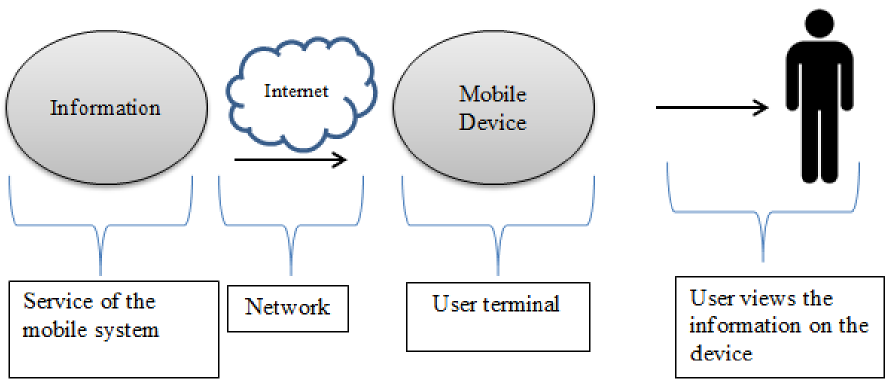


Figure 2.1: Concept of a Mobile System

Portable devices offer services such as information. The data, user interface and operations may vary depending on the environment. In an *adaptable* mobile system, user requirements can alter dramatically and, consequently, require *adjusting* them to new demands. Furthermore, user preferences may constantly differ, such as, for example, new system interactions or different types of service quality [89].

Typically, mobile systems are composed of the following elements [117]: mobile devices; users; wireless networks; portable applications; databases; and *middleware*.

2.1 Difference between Mobile Systems and Traditional Systems

According to John Krogstie *et al.* (2004) [51], the differences between mobile systems and traditional systems are in the following areas:

- **User guidance and customization**: It is important to design mobile systems, from the initial software-development phases, to be simple to any type of user. In fact, mobile systems should be adaptable to user preferences. The system interface has to be comprehensible and intuitive in order to perform queries easily and rapidly. Furthermore, it is essential to consider the input and output, such as a keyboard, voice recognition or touch screen.

- **Convergence and multichannel support**: A characteristic of mobile systems is *technology convergence*, which permits integrating functions that were previously offered in separate devices. This allows applications to provide additional services than those originally conceived.

  Mobile technologies can provide different alternatives to access information. However, developers should consider the limitations of mobile devices, such as processing power, memory and communication capabilities.

- **Customization-oriented**: Mobile systems generate initiatives to modify traditional information systems and, consequently, produce additional ideas in the early development stages. However, these initial ideas must not be limited to current technology. Mobile systems should have *flexibility* and *scalability*.

2.2 Overview of Mobile Systems

Mobile systems can benefit users in various types of environments. However, it is important to consider some of their shortcomings. Thus, the following sections present objectives, advantages and limitations of mobile systems.

2.2.1 Objectives

The objectives of mobile systems are the following [20]:

- **Provide mobility**: A *key* characteristic of mobile systems is that they can be utilized *anytime* and *anywhere*.
- **Improve operations**: Portable systems provide benefits that can simplify services.
- **Overcome business obstacles**: Mobile systems benefit institutions to expand their customer service and not limit themselves to a single area.
- **Improve information quality**: Portable systems, in conjunction with traditional information systems, are utilized in order to enhance data quality.
- **Reduce transaction delays**: Institutions use mobile systems to reduce activities and increase productivity.
- **Improve efficiency**: Portable systems provide opportunities to utilize few resources while preserving time.

2.2.2 Benefits

Mobile systems provide modern services because of their portability and flexibility. Services can be offered in situations where traditional access is hampered by the lack of a stable and reliable network infrastructure [89].

Mobile computing provides strategies to increase efficiency, productivity and profitability. Low-cost mobile devices currently exist due to various hardware options. Additional benefits of mobile systems are [33]:

- **Improved information access**: Mobile devices can transmit data to a base station in order to store and share it with users. However, this process depends on the hardware and communication components of mobile devices.
- **Increased operational efficiency**: Mobile computers possess capabilities to improve operations. In addition, they increase efficiency in several forms, including preserving time or reducing operations.
- **Increased management effectiveness**: Portable computing permits access to the most current information.

2.2.3 Limitations

Mobile devices possess a number of hardware and software limitations that should be taken into consideration in order to ensure accessibility. Furthermore, they have small screens; limited input interaction, bandwidth and computational resources. These shortcomings should be considered when implementing *reliable* mobile systems [89].

The limitations of mobile systems are the following [20]:

- **Insufficient or limited bandwidth**: Wireless Internet in mobile devices is generally slower when compared to a wired connection. Technologies that provide Internet-access include *General Packet Radio Service* (GPRS) [63], *Enhanced Data Rates for Global System for Mobile Communication Evolution* (EDGE) [121], *Third Generation Networks* (3G) and its successor, the *Fourth Generation* (4G) [21, 100], etc.

- **Energy consumption**: Portable devices rely on a power source, such as a battery, when an electrical outlet is not available. Thus, they have to be designed emphasizing energy conservation and efficiency.

- **Transmission interference**: Weather, buildings, long distances and other obstacles can interfere with the device's signal reception.

- **Security**: There are security risks when a portable device is used in a public network. Thus, it is recommended to utilize a *Virtual Private Network* (VPN) [67] for increased security. However, VPNs, despite being secure, have risks as well. Security is a major concern in the Mobile Computing field [21].

2.3 Components and Architectures of Mobile Systems

The following sections delve into greater depth about components and architectures of mobile systems, as well as their characteristics, benefits and importance.

2.3.1 Components

According to Gupta Deepak *et al.* (2012) [20], the components of a mobile system can be classified into three categories:

1. Hardware;

2. Software; and
3. Communication.

Hardware

The characteristics, in terms of hardware, are classified by the size and form of the device, microprocessor, primary and secondary storage, screen size, input and output, battery capacity, communication capability, and device durability.

Software

Mobile computers utilize an operating system to perform tasks. The most common operating systems are: *Android* [96], *iOS* [96] and *Windows* [96]. These systems provide an *Integrated Development Environment* (`IDE`) [115] to develop applications. The majority of operating systems offer a wide variety of `IDEs`.

Communication

This category can be considered as a *key* feature of Mobile Computing. The forms of communication between a mobile system and base station are the following [33]:

- **Connected**: Implies a connection that is continuously available.
- **Weakly Connected**: Implies a continuous communication that has lower speeds.
- **Batch**: Implies a periodic or random communication, consequently reducing time and resources.
- **Disconnected**: Implies a mobile device that does not have the ability to communicate or exchange data. The said process can only be executed by manually inserting data into the system.

Currently, there are several technologies that enable mobile devices to communicate over a network. The most common are: *Wireless Local Area Network* (`WLAN`) [15], *Communication Satellite* [52], *Cellular Digital Packet Data* (`CDPD`) [102], *Personal Communication Systems* (`PCS`) [52], *Global System for Mobile Communications* (`GSM`) [52] or *Internet* [20].

2.3.2 Architectures

An infrastructure (architecture) can be defined as properties of a system in an environment involving its elements, relationships, objectives, design and evolution [41, 75]. A clear and well-developed infrastructure is indispensable in a mobile system due to the following reasons [17, 44]: establishes a clear structure of the system and modules; determines its qualities; and serves as a basis for the system life cycle and its future.

The components of an architecture possess [101]:

- a defined set of responsibilities; and
- a well-defined structure that clearly establishes the relationship between each element.

The most common type of architecture is based on layers, which is shown in Figure 2.2. In this infrastructure, the components are arranged in horizontal blocks, and each layer has a specific objective. The components have responsibilities based on the level to which they belong, providing simplicity to manage effective tasks. Additionally, it facilitates the development, management and maintenance process [97].

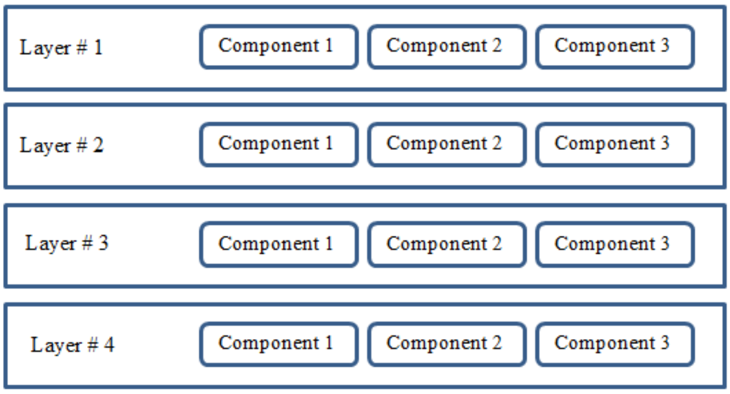


Figure 2.2: Layered Architecture

Importance

Architectures play a significant role in the success of a system and are indispensable due to the following reasons [21]:

- **Better comprehension**: enable a better understanding of systems.
- **Reuse**: provide an opportunity to reuse the same structure for other systems. Thus, the development process is rapidly executed.
- **Evolution**: permit a forecast of how the system will evolve and what elements it will require.

- **Analysis**: provide new opportunities to improve the system.
- **Management**: each system component can be managed in order for them to work in conjunction.

Benefits

The advantages of utilizing architectures for mobile systems are the following:

- **Flexibility**: This refers to a system that can be utilized in areas where it was not initially conceived [41, 75]. An architecture permits adding or modifying services depending on the users' needs.
- **Organization**: allows all system components to work together and satisfy user requirements.
- **Simplifies mobile system development**: offers flexibility and, thus, simplifies the development process. Developers have greater control in each layer, which permits improving, modifying or adding system services.
- **Simplifies the maintenance process**: each component has a clear motive and, thus, the maintenance process is simplified. In addition, it permits a better understanding of the system and what components must be improved.
- **Increases the lifespan of a mobile system**: The lifespan of a mobile system depends on a well-designed infrastructure that simplifies modifications and enables its evolution.

2.4 Current State and Future of Mobile Systems

Mobile devices have capabilities to communicate through wireless networks, and as technology evolves, they will progressively improve. There has been strong evidence that mobile computing is becoming more dominating. Currently, users utilize mobile systems daily, and a higher interaction is expected in the forthcoming years. Since such area is expanding considerably [26], developers are constantly implementing applications in order to satisfy users' needs and requirements [98]. Furthermore, these systems are widely considered to be an integral part of the future of computing [123]. Several types of systems can be expected in the future, using new technologies to provide a modern service [89].

3 Precision Beekeeping

A healthy ecosystem is indispensable for human life, animals and natural resources [5]. An ecosystem can be defined as a group of organisms interacting with each other and the environment. Thus, ecosystems are composed of two elements: *biotic* [36], which are the living components such as plants or animals, and *abiotic* [104], the non-living components, for example nutrients, temperature, or humidity. There are two types of ecosystems: terrestrial and aquatic.

Ecosystems are essential for the following reasons:

- permit soil maintenance [118];
- provide plants, medicine, food, water and air [31];
- provide nutrients [62]; and
- prevent erosion [4].

Since ecosystems are indispensable, a field called *Precision Agriculture* [116] emerged, which consists in monitoring and controlling agricultural elements [124]. This is done through *Information Technologies* (`IT`), which allow the user to collect data more precisely and accurately. There are several branches in this field, such as precision livestock [61]; precision horticulture [37]; precision viticulture [113]; and precision beekeeping [129]. All of the aforementioned fields monitor various elements such as land, crops, plants or organisms.

3.1 Overview

Honey Bees are natural sensors of ecosystems and one of the most important insects. They possess the ability to pollinate [80, 81], consequently providing nutrients for humans and animals, as well as aiding in ecosystem health. Thus, it is indispensable to preserve them, as they play a major contribution to ecosystems, as well as to the global economy [126].

Bees are insects that emerged at least forty thousand years ago [18]. Their development consists in four stages: egg, larva, pupa and adult [120]. Pollen is indispensable to feed young bees and, consequently, permits an increase in honey production. This food originates from nectar, which is converted by bees and used as an energy source to perform tasks. Bees often communicate in two ways: dancing or using pheromone [77]. The colony consists in three types of members [18, 19, 83]:

- **Queen**: The most important member of the colony due to producing eggs. She is the only female that has a properly-developed reproductive system. The queens are chosen, at the larval stage, by the worker bees and are constantly fed with royal jelly, which is rich in protein and consumed by humans for: improving overall health and increasing longevity for the elderly [72]. The queens live up to four or five years, however, they must be replaced between a year and a half to two years, as their ability to lay eggs decreases with time.

- **Worker**: female bees, being the most numerous and hard-working, responsible for: honeycomb construction; feeding the queen and drones; defending the colony; and hive hygiene. Unlike the queen, they lack a complete reproductive system. Life expectancy varies depending on the season: between fifteen to thirty-eight days in the summer; or one hundred and fifty to two hundred days in the winter [94].

- **Drones**: male bees, which its primary function is to fertilize the queen to produce eggs [120]. Drones live between eighty to ninety days; or until the fertilization process is concluded. They depend solely on food provided by the workers.

Bees live in a hive, which is utilized to work, store honey, and produce eggs. These insects generate high-quality honey regardless of the type of beehive. A hive can host approximately sixty-thousand workers and five-hundred drones [18]. Its size depends on the bee population. The weight of the beehive is considered an important factor to determine the state and productivity of the colony [29].

An apiary is a set of beehives placed in a single location [9, 19], which its goal is to allow honey production; pollination; and bee maintenance. Beekeepers place the hives in specifically-designed boxes, which protects them from possible threats [10]. The bees will abandon the hive in case they do not perceive benefits for the colony and their honey production. Thus, it is essential to confirm that the apiary, beehive and tools provide advantages for both beekeepers and bees. The elements that need to be considered to establish an apiary are the following [10]:

- should be located near trees and water;
- should be distant to prevent possible threats;
- must not receive strong sunlight, wind, or rain. This can harm the equipment and bees; and
- must allow for accessibility, in other words, simplify the inspection and maintenance process.

It is necessary for beekeepers to analyze bees with the following tools:

- **Fumigator**: its objective is to calm bees, allowing beekeepers to better analyze the said insect and hive.
- **Chisel**: its purpose is to open, remove and clean beehive frames.
- **Protective clothing**: aims to cover the body and face of the beekeeper.

Recently, beekeepers have perceived the necessity of adopting `IT` into the agricultural field [57]. This occurred not only because of the contribution provided by Honey Bees, but also due to the fact that bee population has been decreasing over the last few years [60, 80]. Therefore, monitoring these insects has become a crucial activity [81, 126].

In this context, *Precision Beekeeping* (`PB`) emerged, which is a subdivision of Precision Agriculture [130] that applies `IT` in order to determine the state of the bee colony and improve its preservation [58, 129].

3.2 Importance

Bees are pollinators [10, 80] and, thus, considered indispensable contributors to agricultural fields [10]. Therefore, several institutions were established to protect bees [80]. `PB` permits the following [39]:

- collect data to analyze and maintain bee colonies;

- a collaboration between agronomists and computer scientists to manage ecosystems;
- an educational contribution about bees, their hives and importance;
- a better understanding of these insects and their habits;
- obtain nutrients such as pollen, nectar, and honey, which contain essential properties for the medical field; and
- ensure an enhanced method to maintain bees and, consequently, increase pollination, which will provide: plant fertilization; medicines; nutrients; greater crop manipulation; and environmental benefits.

Research regarding PB began in 1926, when W.E. Dunham [23] utilized eight thermoelectric pairs to measure the temperature of a beehive. Throughout the years, many investigations were made with modern tools, such as, thermal or infrared cameras, microphones and weight scales.

3.3 Precision Beekeeping and Computer Science

Beekeepers obtain a better understanding of bees through IT. Beekeepers apply IT for several reasons, including to: ensure that their crops are pollinated properly; and harvest and produce honey. IT are utilized in order to support, but not replace, the beekeeper [125]. There are several parameters that beekeepers evaluate, such as: temperature [127], humidity [95], audio [80, 81], images [103] and weight of the beehive [29].

Temperature is one the most popular parameters because of its relatively low cost and simplicity [131]. This variable provides data regarding food consumption, bee development, hive abandonment, and death of the colony [130, 131]. The collected results must not only be presented in real-time to the user, but also stored for future analysis.

Recently, the necessity of improving PB has been discussed [80]. According to Aleksejs Zacepins & Jurijs Meitalovs (2014) [127], a PB monitoring system has to:

- reduce time and costs; and
- minimize manual inspections and maximize bee maintenance.

4

Precision Beekeeping Information System

This chapter describes a *Precision Beekeeping Information System* (PBIS) called myBee, which served as a motivation for this master's project.

4.1 Introduction

A well-developed *Information System* (IS) has the following characteristics:

- simplifies the development and maintenance process;
- provides reusability of modules and subsystems;
- modules and subsystems are plug-and-play;
- provides compatibility for different devices;
- provides compatibility for stationary and mobile systems;
- provides security; as well as,
- provides quality of service.

Taking all of this into consideration, it can be stated that a flexible [93], scalable and efficient architecture is required in order to implement a reliable IS. In fact, Aleksejs

Zacepins *et al.* (2012) [125] claim that infrastructures with sub-elements are needed for PB.

myBee is based on two approaches described by Armands Kviesis & Aleksejs Zacepins (2015) [56]:

1. using an interface device for each beehive; and

2. sending data to a remote computational center.

Therefore, myBee offers a more detailed analysis of the bee colonies, resulting in a better maintenance and preservation of Honey Bees.

Figure 4.1 outlines myBee, which is divided into two subsystems:

1. the *Stationary System*; and

2. the *Mobile System*.

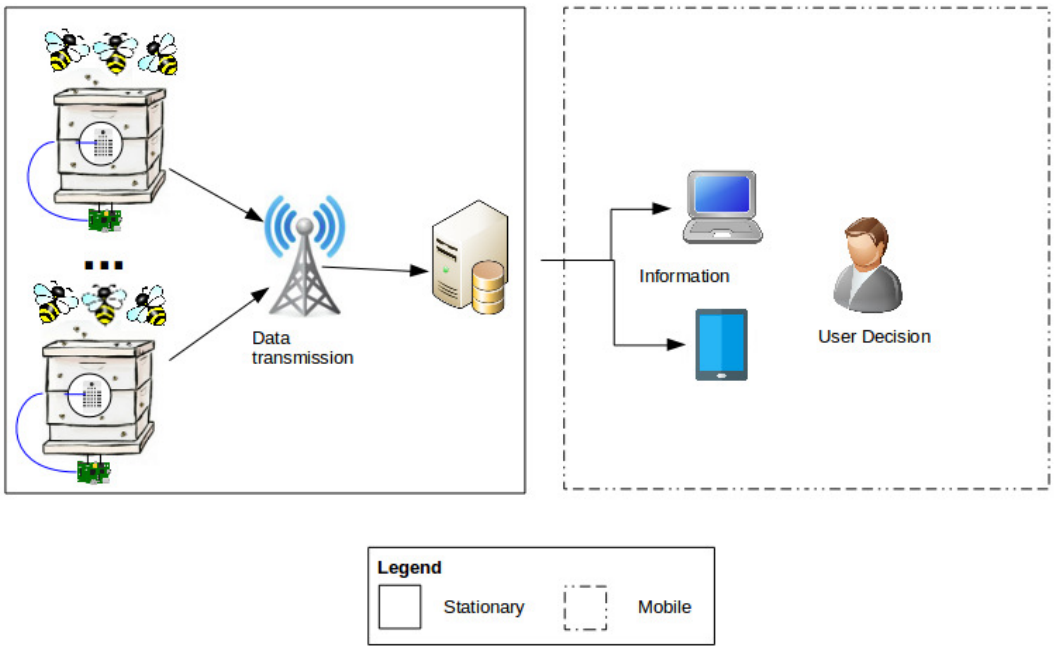


Figure 4.1: PB System Platform

The *Stationary System* consists in collecting and storing the data. The *Mobile System* consists in the software to monitor the collected data. A brief description of myBee is as follows: the sensors located at the center of the beehives monitor the conditions, collecting pieces of information, which are sent through a wireless network to a server. Therefore, by using the *Mobile System*, beekeepers can monitor via a web interface on a computer or mobile device the conditions of the bee colonies.

4.2 The Stationary System

The *Stationary System*, whose objective is to provide a well-defined and developed `IS`, is an architecture and can be viewed in Figure 4.2.

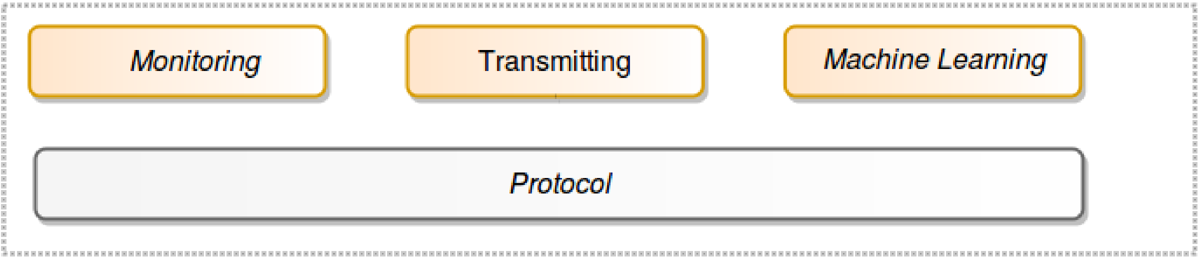


Figure 4.2: Stationary System Architecture

The infrastructure has the following classes:

- Monitoring: provides the functionality to monitor a certain condition of the environment.
- Transmitting: provides the functionality to receive and transmit the data.
- Machine Learning: estimates future conditions of the bee colony.
- Protocol: provides the functionality for communication between devices.

4.2.1 The Implementation

`myBee` has the following functionality:

- uses low-cost devices and a database system manager;
- monitors the temperature and humidity of the beehives;
- organizes monitoring-devices in a mesh network;
- provides reports and statistics; and
- anticipates future behaviour.

Hardware Currently, the Stationary System supports a Raspberry Pi, *DHT22* sensor, and the microcontrollers *GPIO7* and *GPIO18*. This indicates that the said system uses low-cost hardware, which is able to: deactivate the sensor *DHT22* for a certain time; collect the temperature and humidity of the beehive; store data; and transmit/receive data.

Database The Stationary System supports the database management system, MySQL. The data is stored on a database as shown in Figure 4.3.

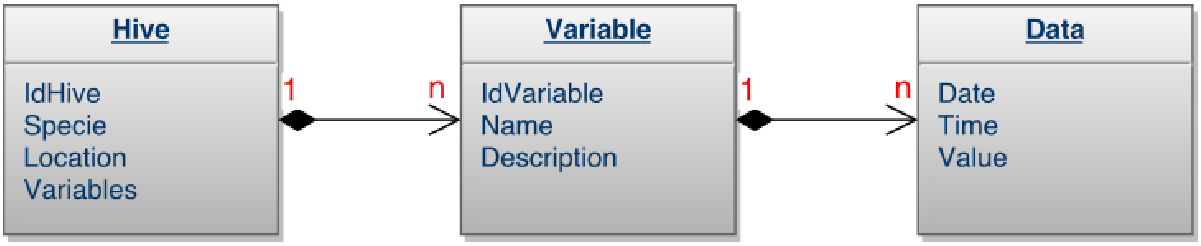


Figure 4.3: Database

Protocol The monitoring-devices that compose the Stationary System automatically organize themselves as a mesh network, which differs from a traditional network because each node serves as an access point. Thus, it provides a higher fault-tolerance and simpler maintenance because the network adapts automatically to the number of nodes available [65]. Data is sent from one node to the other until it reaches its target-location.

A technique to deliver messages is by utilizing *Interest Ad-Hoc Networks* (Radnets), which is based on user interests and characteristics. Radnets permit the following [32]:

- a collaboration between the network nodes; and
- a message-delivery approach for users who have the same interests. This approach consists in an asynchronous model, called Publisher/Subscriber, in which a publisher node sends messages to subscribers with the same interests.

The address of network devices and users is performed through the Active Prefix [25, 32], which is divided into two fields: Prefix; and Interest. The Prefix represents user characteristics and is used as an *Internet Protocol* (IP) address. However, unlike the IP, the Prefix is linked to the application and not to the device. In addition, several users can have exactly the same Prefix. The Interest is a field that stores and represents an interest of the application [32].

The Radnet Protocol can return messages to a node and, consequently, provide an opportunity for *reprocessing* to occur. In order to avoid such an event, whenever a message is received it is inserted into a hash table for verification purposes [32].

The aforementioned protocol was developed for mobile and low-power applications. Radnet's main features include: energy-saving; sending data through multiple nodes that have common interests; and adapting to node additions, failures or removals [25, 32]. Thus, the aforementioned protocol is ideal for mobile and low-power applications.

It is worth highlighting that Radnet was developed with security in mind, attaining Active Prefix messages by either cryptographic signatures or passwords [25]. Based on these advantages, this protocol was chosen for the PBIS.

Computational Intelligence myBee provides possible-future values of the monitored elements. This action is performed by the Machine Learning module. Basically, when the said functionality is triggered, the Stationary System activates artificial intelligence algorithms to estimate values of the monitored elements, from the existing database. Whenever an inference is made, the data will be validated as soon as the system obtains it. If the inference is wrong, a warning is issued to the user. As an example, myBee anticipates, based on the collected data, whether the temperature or humidity will reach unfavorable values and, thus, notify the user.

Client/Server Side The characteristics of the Stationary System architecture allow both clients and servers to be created. The former monitors the environment and sends the data to the server. The latter permits viewing the data.

4.3 The Mobile System

The Mobile System has three objectives:

1. visualize the data monitored by the Stationary System;
2. provide data and statistics; and
3. provide notifications about undesirable behaviours.

Similar to its counterpart, the Mobile System is composed of an architecture. Figure 4.4 displays the infrastructure of the Mobile System platform.

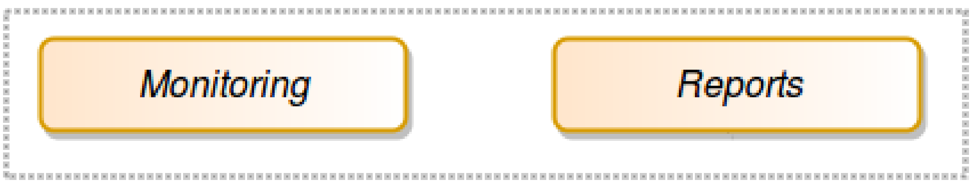


Figure 4.4: Mobile Architecture

The components of the architecture are the following:

- **Monitoring**: consists in monitoring real-time data on a mobile device.

- **Reports**: provide information based on the collected data.

4.3.1 The Implementation

The Mobile System consists in a Web interface to monitor the bee colonies. Figure 4.5 displays the Web System.

The Web System has the following functionality:

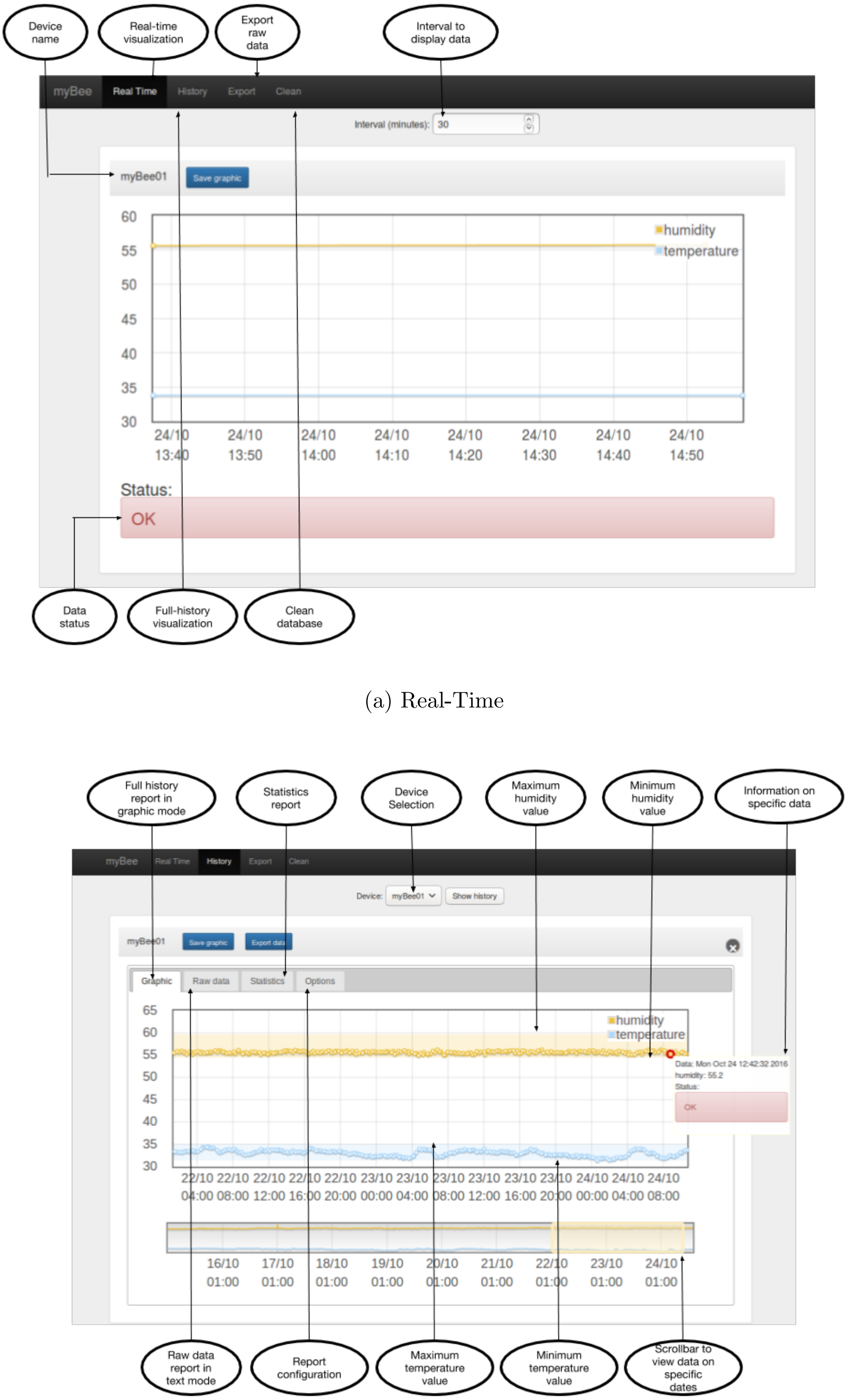
- displays the monitored data;
- generates graphs;
- provides reports;
- provides statistics; and
- provides notifications;

The system-visualization module graphically displays the collected data, allowing the user to select the time period. The report module provides detailed reports, including: description of each beehive, location, monitoring time and date. Both graphs and reports can be exported, with different extensions, and, thus, used separately.

The statistics module provides numeric values of the monitored data, which are: general mean, standard deviation, variance, minimum and maximum value. The first three can be viewed in reports, while the remaining two are displayed graphically.

The said functionality can be applied to each Stationary System individually. Thus, a single Mobile System is used to monitor all Stationary Systems installed on different beehives. It is worth mentioning that the Web System provides five data filters, which are applied when viewing the data, and they are:

- all data monitored thus far;
- data monitored between 12:00 AM and 6:00 AM (0h - 6h);
- data monitored between 6:00 AM and 12:00 PM (6h - 12h);
- data monitored between 12:00 PM and 6:00 PM (12h - 18h);
- data monitored between 6:00 PM and 12:00 AM (18h - 24h);



(a) Real-Time

(b) History

Figure 4.5: The Web Interface

The notification module provides warnings to the user, indicating that the conditions of the bee colony are undesirable. Thus, the user can make appropriate decisions depending on the issued notifications.

Finally, the Web System provides the user with information estimated by the Machine Learning module in the Stationary System.

4.4 Experiments

Beekeeping is a sustainable activity that generates positive impacts on social, economic and environmental areas. This activity provides: income to beekeepers by commercializing their products; and benefits to the environment. These results favor the balance and maintenance of biodiversity [11].

Controlling temperature and humidity is essential because biological processes can be modified and/or altered by high variations. Thus, it is important to implement technologies in order to maintain adequate beehive conditions.

Therefore, `myBee` was validated by monitoring the internal temperature and humidity of a beehive corresponding to the *Apis Mellifera* species.

Since `myBee` is based on the infrastructures described in Figure 4.2 and Figure 4.4, it possesses the following characteristics:

1. **Flexibility**: the `PBIS` has to be based on simplicity. As a result, `myBee` can be easily maintained and modified to better suit the beekeeper's needs.
2. **Fault tolerant**: the `PBIS` has to efficiently handle potential errors. Thus, `myBee` was implemented with several precautionary measures including data redundancy.
3. **Security**: the `PBIS` has to provide security mechanisms to ensure that data is not violated. Thus, `myBee` uses a secure protocol.
4. **Efficiency in decision-making**: the `PBIS` has to simplify the maintenance of bee colonies. As a result, `myBee` provides several reports, estimates future conditions and warns about undesirable behaviours.

4.4.1 Experiment Area

`myBee` is currently used in the *Experimental Farm of Iguatemi* (`EFI`), viewed in Figure 4.6. The `EFI` is located at a latitude of 23°25' S; 51°57' O, an altitude of 550 meters and area of 170 hectares. This location provides a suitable environment to develop projects on agriculture and animal husbandry.

Figure 4.6: Experimental Farm of Iguatemi, Source: Google Earth

The apiary is composed of 10 beehives, which are arranged into two types of boxes:

- Styrofoam and
- Wood.

Different materials are used in order to evaluate the conditions of the beehives, each with distinctive treatments.

The boxes were arranged in contrast to one another and directly exposed to the weathering of the climate, reducing interference of non-climatological factors in the experiment. The experiment area is surrounded by an eucalyptus plantation.

The placement of the wooden and styrofoam boxes can be visualized in Figure 4.7.

The red-dotted line represents data transmission. This means that each collected information is sent to every node, through `Radnet`, for backup purposes. This data-redundancy guarantees that the data will be stored regardless of equipment failure.

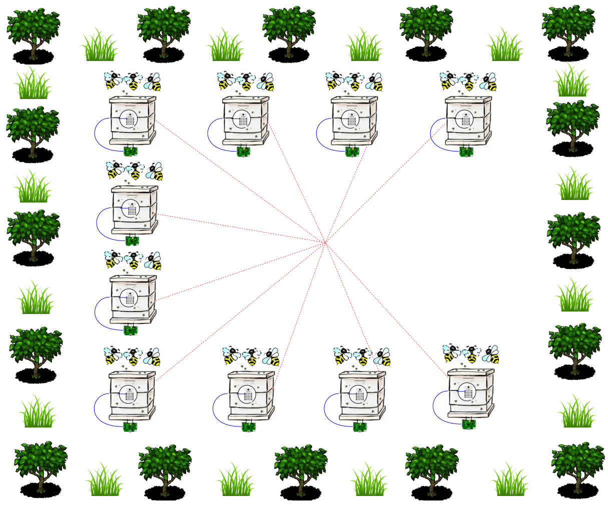


Figure 4.7: Apiary

4.4.2 Observations

This section presents data collected over a period of 10 days. The values shown in Figure 4.8 prove that both materials did not interfere with the temperature of the swarm.

Figure 4.8: Temperature

The humidity, as shown in Figure 4.9 was significant, achieving a higher stability for swarms placed in a Styrofoam box.

According to Thomas Seeley (2006) [105], controlling the temperature of a beehive can be seen as one of the greatest innovations.

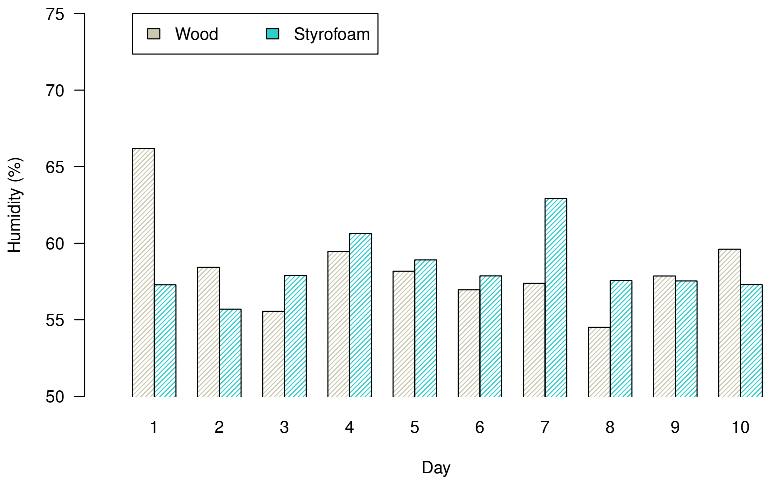
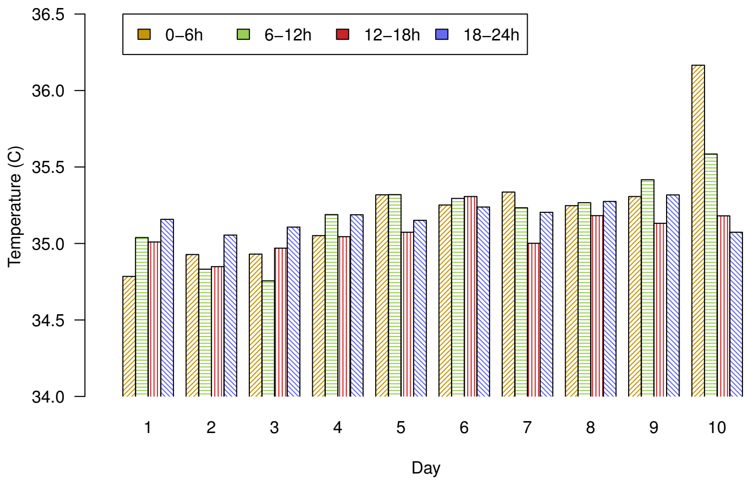
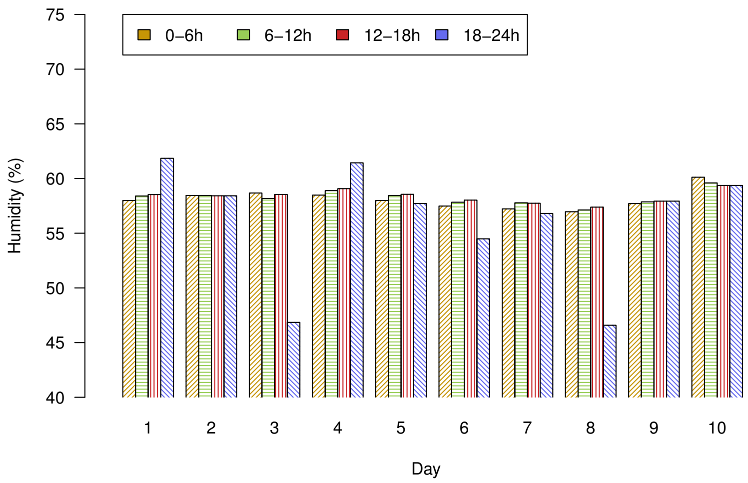


Figure 4.9: Humidity

The values obtained in the four periods of 6 hours/day, as shown in Figure 4.10 and Figure 4.11, were within the values of homeostasis. In addition, it revealed that a higher temperature is expected during the night.

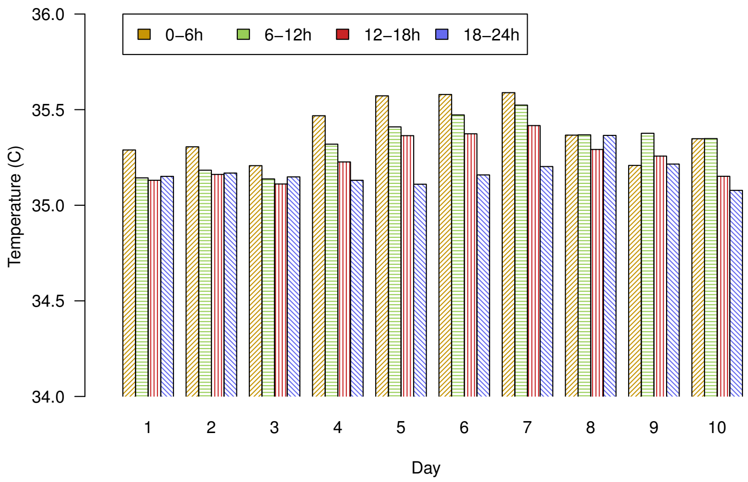


(a) Temperature

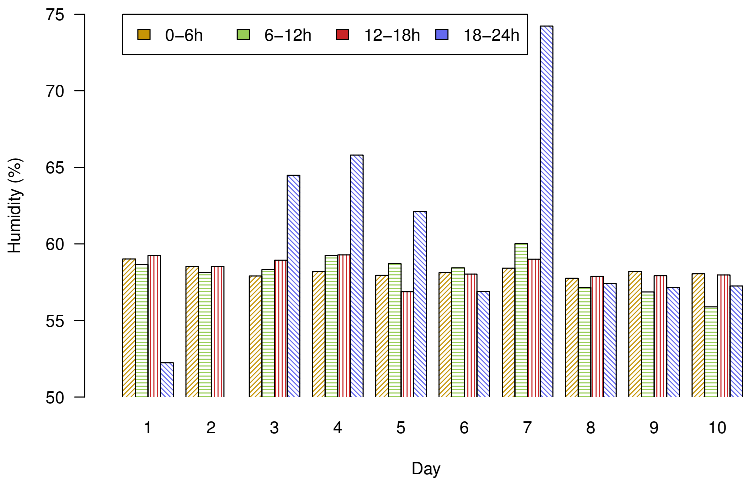


(b) Humidity

Figure 4.10: Wood



(a) Temperature



(b) Humidity

Figure 4.11: Styrofoam

4.4.3 Validation

An infrared camera-system was utilized to analyze the internal temperature of the beehive, as shown in Figure 4.12.

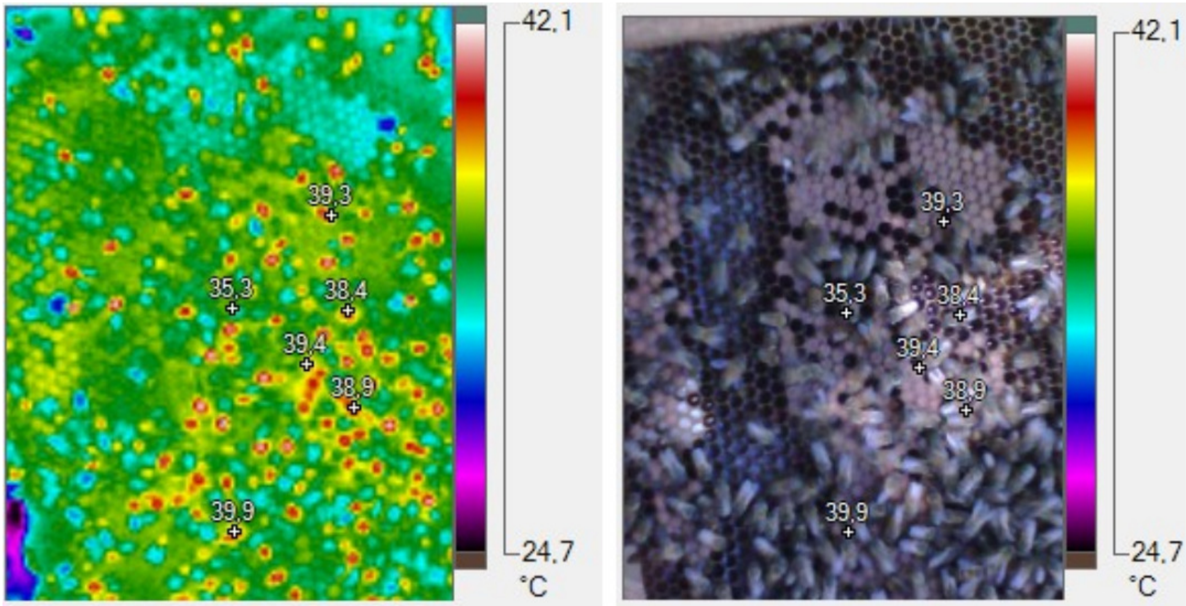


Figure 4.12: Infrared Image of a Part of the Beehive.

The use of such a strategy has two problems:

1. the use of infrared images is susceptible to a known error of 2%; and
2. the user plans to handle each beehive to identify its condition.

Comparing the data monitored by `myBee` with the data collected by the infrared camera, it is demonstrated that `myBee` has the following advantages:

- the data monitored by `myBee` is statistically the same as that obtained by the infrared camera;
- the collected data is not susceptible to an error percentage;
- there is no need to handle each beehive to identify its condition;
- there is no need to process the data to obtain reports and statistics;
- an unfavorable condition in the beehive is known in real-time.

Using `myBee` in a real-world environment demonstrates its ability to monitor the conditions of the bee colony in order to perceive alterations and to analyze the effects of environmental variables (temperature, humidity) on different boxes.

4.5 Related Works

This section presents related works in regards to Precision Beekeeping and Generic Information Systems. The works were divided into the following topics: Visualizing the results via a web interface; Visualizing the results offline; and Proposals made in recent years.

4.5.1 Visualizing the Results via a Web Interface

Aleksejs Zacepins *et al.* (2011) [128] implemented temperature sensors, during the winter season, in case bee-development occurs. The system consisted in: transferring, storing and visualizing data in a personal computer; a database implemented in *Microsoft Access* to store temperature results; a web interface for the beekeeper to quickly access the information; notifications in case the temperature is not adequate; and data backup.

Aleksejs Zacepins *et al.* (2012) [125] developed a web system to monitor, in real time, a bee colony. The software architecture consists in: a configuration file which sets the system parameters; a database implemented in *Microsoft Access*; an application which stores temperature data for future analysis, and a web system which displays the temperature of the bee colony in real-time.

Aleksejs Zacepins *et al.* (2014) [127] implemented a temperature-measurement system in the PB field, which consists of two phases: real-time temperature measurement of the beehive and visualizing the parameter results in a web server. The researchers implemented the system using a *Microsoft Access* database, which permits sharing the collected results through *Dropbox*. The temperature sensors were *DS18S20*.

Marco Giammarini *et al.* (2015) [30] developed a monitoring system to collect temperature and humidity of two beehives in the summer. One beehive was placed in a wooden box, while the other in a plastic box. A GSM modem was implemented for sharing and downloading data; remote monitoring; and software debugging.

Fiona Edwards Murphy *et al.* (2015) [79] implemented a system, utilizing *Wireless Sensor Networks* (WSN), to monitor a bee colony. The system collected data such as temperature, carbon and nitrogen dioxide, pollutants and battery percentage of the device being utilized. The researchers applied WSN due to being a non-intrusive technology, thus, allowing more accurate data collection. The information can be accessed through a web interface or mobile device.

4.5.2 Visualizing the Results Offline

Octavio A. Márquez Reyes *et al.* (2012) [95] implemented a monitoring module based on five variables: humidity, temperature, population, movement and water. The platform evaluated these five elements and the results were visualized in a personal computer or stored in a memory card. The sensors utilized for the experiment belong to the *Sensirion SHTxx* series, which have a response time of approximately four seconds and low energy consumption (30 μ W).

Fiona Edwards Murphy *et al.* (2015) [80] implemented a prototype to collect images and audio within a beehive. The platform utilizes a *Libelium Waspmote* and *Raspberry Pi* in order to process and store data. Additionally, microphones, accelerometers, thermal and infrared cameras were used, along with emergency notifications to the user in case an undesirable event occurs to the beehive. The goal was to utilize non-intrusive equipments to collect data.

4.5.3 Proposals Made in Recent Years

There have been several approaches for implementing monitoring-system architectures in the `PB` branch.

Douglas S. Kridi *et al.* (2014) [50] proposed an algorithm to anticipate beehive abandonment based on high temperatures, lack of food, and humidity. The algorithm was validated with a monitoring system. The objective was to identify the behavior of bees when they abandon the hive and, at the same time, minimize energy consumption and data transmission.

Armands Kviesis *et al.* (2015) [58] detailed six different approaches for `PB` system platforms and proposed an algorithm to select a method based on the beekeeper's needs. The researchers stated that, currently, several platforms for `PB` have been implemented and therefore, the algorithm will simplify decision-making and reduce time.

Armands Kviesis *et al.* (2015) [59] proposed a PB system platform that utilized a *SHT15* sensor to measure temperature and humidity. A total of eight hives were placed outdoors, each with measurement nodes in closed boxes and protected with waterproof material. All temperature and humidity data were stored in a `SQL` Database and can be visualized in a web application.

Armands Kviesis *et al.* (2015) [60] proposed a decision-support module to better-comprehend the collected data and, consequently, execute reliable actions. The sensor utilized in the experiment was *DS18S20*, which measures temperature. The sensors were connected to a *Raspberry Pi* and data was sent to a server and stored in a `MySQL` database. Two systems were developed, the first monitored ten honey bee colonies inside a wintering building, while the second monitored ten honey bee colonies outside. The data can be visualized in either a web or desktop application, providing the option to view the maximum, minimum, median and average temperatures, per day, for all installed sensors. Both systems have a decision-support and analysis module.

Table 4.1 offers a comparison between the aforementioned `PB` related works and `myBee`. Based on this table, there exists several `PB` information systems that collect temperature/humidity; provide a web interface; and offer data security. However, `myBee` has Machine-Learning capabilities, which permits it to predict future or possible temperature/humidity values.

Table 4.1: Comparisons of Related Works with `myBee`

Related Work	Temperature/Humidity	Web Interface	Data Backup	Machine-Learning
[128]	**Yes**	**Yes**	**Yes**	No
[125]	**Yes**	**Yes**	**Yes**	No
[127]	**Yes**	**Yes**	**Yes**	No
[30]	**Yes**	**Yes**	**Yes**	No
[79]	**Yes**	**Yes**	**Yes**	No
[95]	**Yes**	No	**Yes**	No
[80]	**Yes**	No	**Yes**	No
[59]	**Yes**	**Yes**	**Yes**	No
[60]	**Yes**	**Yes**	**Yes**	No
`myBee`	**Yes**	**Yes**	**Yes**	**Yes**

4.5.4 Generic Information Systems

Wilson Goudalo *et al.* (2016) [34] mention how, currently, Information Systems play a significant role for Enterprises. Thus, they proposed various methods to provide simple user interfaces for managing security. The authors applied seven principles of the ISO 9241-11 [42], which are: *clarity*, *discriminability*, *brevity*, *consistency*, *detectability*, *readibility* and *comprehensiveness*. All these principles, including security concerns, were all taken into consideration for `myBee`.

Delfina Soares *et al.* (2014) [109] defined *interoperability* as a characteristic in which entities preserve autonomy and independence. Thus, `Radnet` can be categorized as *interoperable* due to exchanging information while being independent from one another. The authors also recalled the importance of a social-technical perspective for information systems, which applies to `myBee` as well, due to offering data reports that influence user-decisions and system operations.

Hadi Kandjani *et al.* (2013) [47] proposed a framework to classify system-planning methodologies. The authors stated that selecting a proper methodology to develop an information system is a *key* factor for its success. The methodology utilized to implement `myBee` was a success due to the obtained data and comparisons.

Ovidiu Noran (2013) [85] proposed enhancements, based on interoperability, to Disaster Management Information Systems, using an enterprise architecture perspective and artifacts. The author claimed that these types of systems are important due to allowing a collaboration for environmental incidents. This statement can also be applied to `myBee`, which provides crucial data for beekeepers and, therefore, collaborates and supports the environment.

Jorge Aguiar *et al.* (2013) [2] proposed an improvement for Decision Support Systems corresponding to *Intensive Care Units* (`ICU`) based on *Technology Acceptance Model* (`TAM`). The architecture for an `ICU` information system can be divided into two subsystems: one to collect the data and another to process and display data. This statement can also be applied to `myBee`, which is divided into two subsystems (stationary and mobile).

5

Proposal of a Layered Architecture and Performance Metrics for Mobile Systems

A mobile system can be executed anytime and anywhere in order to access information [64, 90]. Recently, mobile gadgets have evolved into powerful portable computing devices with modern wireless technologies that enables them to be utilized in a wide range of scenarios [110]. However, their development is complex because they are executed in dynamic and heterogeneous environments. The following section delves into more detail about portable systems.

5.1 Overview

The Mobile Computing industry is evolving at a rapid pace [64] and, thus, developers must keep up-to-date with recent technologies in order to deliver a modern service to customers. Additionally, the number of users utilizing mobile devices is increasing each year. Figure 5.1 displays a general overview of Mobile Computing. The layers were designed based on the components of portable computing: software, hardware and communication [20, 86]. These components are described in more detail below:

- **Software**: refers to the mobile applications executed on different types of portable devices.

- **Hardware**: refers to the different types of mobile devices. According to Gupta Deepak *et al.* (2012) [20], the characteristics in terms of hardware are defined by the size, physical components and shape of the portable device.

- **Communication**: refers to protocols and network infrastructures utilized for data transmission. According to Gupta Deepak *et al.* (2012) [20], communication is an important characteristic of Mobile Computing. Thus, the last four layers: *Connectivity*, *Network*, *Protocols*, and *End Host* belong to the said component. Communication is related to routers, servers, cloud computing, networks and the Internet.

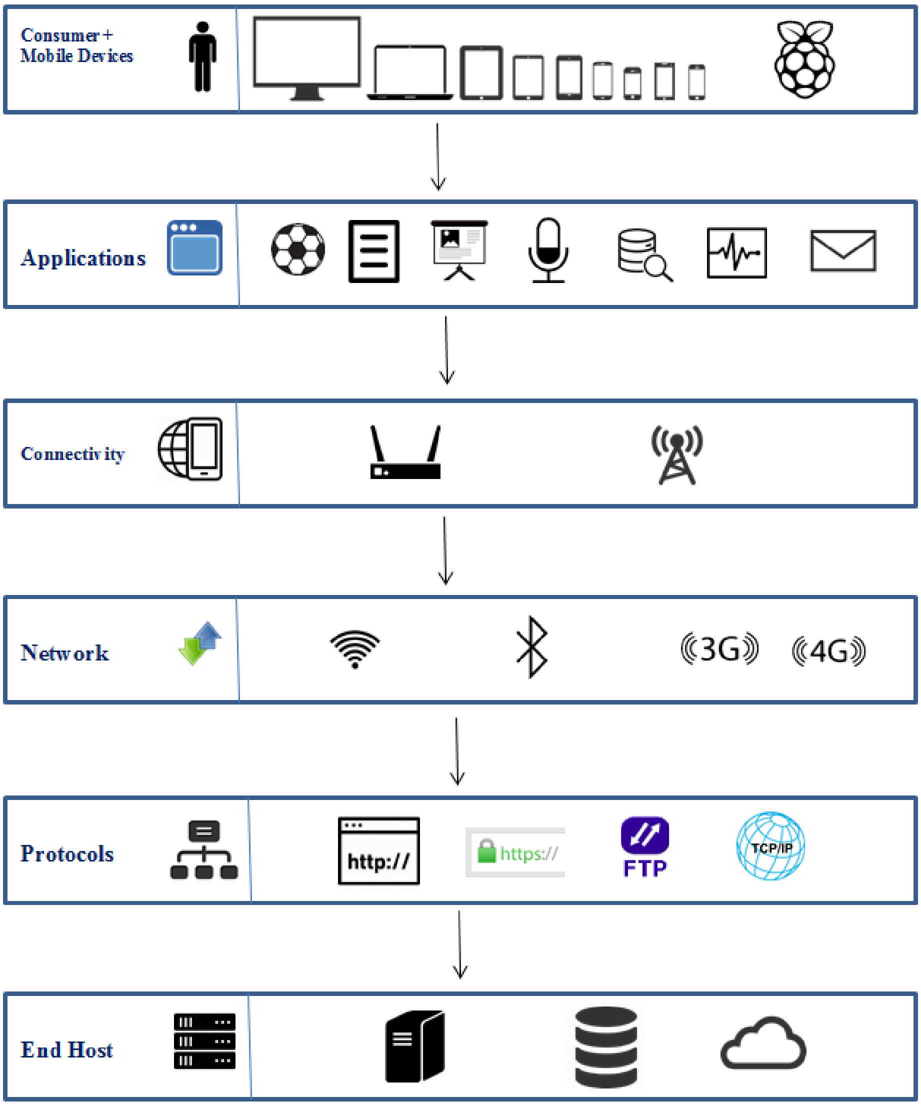


Figure 5.1: Elements of a Mobile System

The layers of Figure 5.1 are related with one another as follows:

- Several consumers use different types of mobile devices;
- Portable gadgets execute different types of applications;
- Mobile applications connect to a router or cellular network for Internet access;
- Internet access is provided through wireless technologies such as `WiFi`, `3G` or `4G` Networks;
- During network access, the applications utilize protocols for data transmission;
- Data is transmitted to the cloud or a server.

The first layer, *Consumer + Mobile Devices*, presents a variety of mobile devices such as televisions, tablets, mobile phones, music players, laptops or a Raspberry Pi. Currently, portable devices have a higher usage than desktop computers [73].

The second layer, *Applications*, presents the various types of applications developed for mobile devices, such as monitoring systems, audio players, database managers, email clients or text processors. The advancements in mobile technologies have enabled a wide range of applications that can be utilized in any location [38]. A mobile device can run different applications, providing services that take advantage of recent technologies.

The third layer, *Connectivity*, displays two ways in which a mobile device can connect to the Internet: a router or cellular network. A router is a device that sends data within a network of computers. A cellular network is utilized to provide mobile services. A portable device obtains Internet access through these two elements.

The fourth layer, *Network*, presents the networks in which a mobile device, through a router or cellular network, can transmit data, for example: `WiFi`; `Bluetooth`; `3G` or `4G` Networks. These technologies are constantly being updated with considerable benefits. Thus, Mobile Computing is characterized for evolving in terms of functionality, services and technologies.

The fifth layer, *Protocols*, displays the different types of protocols for data transmission. Currently, various types of protocols are emerging, providing modern features such as security, file transfers and access to websites.

The sixth layer, *End Host*, presents the target-locations to which the data will be transferred, such as a server, database or the cloud. This location depends on user configuration.

Mobile systems should seamlessly adapt to dynamic environments. In addition, researchers have stated that all future desktop and portable applications need to be capable of adapting to modern requirements [55].

Mobile Computing is an area in which there is no consensus to address the limitations of portable devices [55]. An important challenge is how to maintain or extend the lifespan of mobile systems. This element can be defined as the time in which the efficiency and availability are at their maximum capacity. The lifespan of a mobile system decreases over time and, consequently, affects its services.

As stated before, these acknowledgments and statements were considered when analyzing `myBee`. Thus, based on the said system and to approach such challenges, an architecture will be proposed that provides flexibility and scalability. Since the goal of Mobile Computing is to provide portability, adaptability and flexibility for users [6, 7], this infrastructure needs to possess such benefits to develop any type of mobile system. In addition, performance metrics will also be proposed in order to evaluate mobile systems.

5.2 Layered Architecture

Systems that do not possess a supporting architecture are difficult to modify due to not having a clear direction and purpose [97]. A well-developed infrastructure is vital for mobile systems, due to permitting flexibility, better management, and ease in the development and maintenance process, which can benefit and support portable systems considerably.

Developers can opt in applying two types of architectures: *modular* and *integrated*. The components are connected together in a modular architecture, each with its own responsibility. In an integrated architecture, there is no division between the modules.

Modular architectures have the following benefits:

- permits a clear and organized structure of the mobile system.
- each component has a designated responsibility, simplifying the development process.
- allows replacing or modifying components without affecting the rest of the system.

As shown in Figure 5.2, the proposed architecture for mobile systems is composed of three layers. The Service Layer manages the operations of the system. The Network Layer handles the communication protocols. Finally, the Hardware Layer is composed of the physical components of the device. The *middleware* was designed for flexibility and scalability, as well as to provide a simpler access to each layer. Therefore, the said component is based on interoperability. The *Management* module permits the user to configure or enable the operations, protocols, storage and physical components of the system.

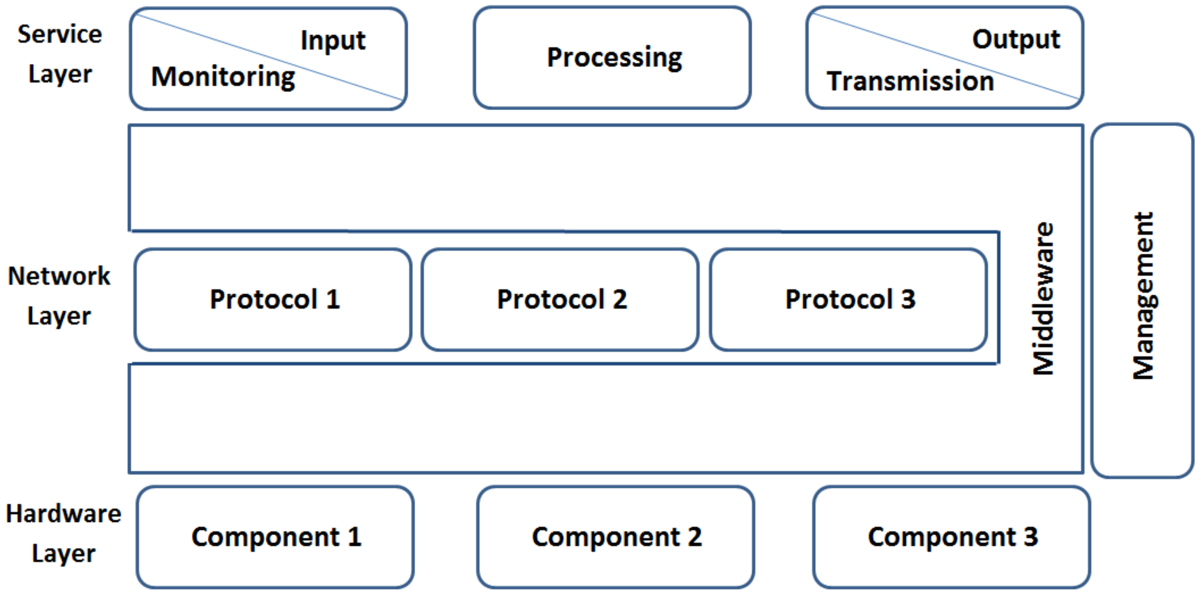


Figure 5.2: Generic Architecture for Mobile Systems

5.2.1 Service Layer

The Service Layer focuses on the three basic operations of an IS:

- **Monitoring/Input**: consists in *collecting* the data and then sending it to the *Processing* module.
- **Processing**: consists in manipulating the collected data. In addition, this module determines the type of output, whether it will be stored, sent to the base station/-server, or both.
- **Transmission/Output**: consists in transmitting the information, using the protocol in the Network Layer, to the server.

5.2.2 Network Layer

The Network Layer offers functionalities such as protocol configuration, and enabling and disabling data transmission. As stated before, the objective of the *middleware* is to offer flexibility, scalability and compatibility in order to add or configure protocols to the platform. The user selects which protocols will be utilized in the *Management* module.

5.2.3 Hardware Layer

The Hardware Layer offers functionality to add, configure, enable and disable the physical components, which are also controlled by the *Management* module. As in the Network Layer, the objective of the *middleware* is to provide flexibility to add or configure physical components. For example, components can be disabled until they are required by the system.

5.2.4 Class Structure

A class structure was designed and specialized to add specific functionalities (components). Thus, it is possible to modify the architecture to provide new functionality, such as using a different communication protocol, or even supporting a different device. Basically, the architecture has the following classes:

- Monitoring: provides the functionality to monitor a certain condition of the environment.
- Processing: provides the functionality to process the monitored data.
- Transmission: provides the functionality to receive and transmit the data.
- Protocol: provides the functionality for communication between devices.
- Hardware component: provides the functionality to manage a hardware component, whether it is a microcontroller, sensor or transmitter.
- Middleware: provides the functionality for communication between the layers.
- Management: provides the necessary functionality to manage the entire architecture.

The aforementioned classes are abstract, meaning they must be specialized to provide concrete functionality, with the exception of the *middleware* and *management* classes.

5.3 Performance Metrics

It is well known that infrastructures are *key* factors in affecting the performance of mobile applications [8]. According to Ching Kin Keong *et al.* (2015) [48], software architectural problems can be avoided by using high level evaluations.

Several architectures have been developed in the last few years, however, there are few metrics or guidelines that reassure developers that their implementations are top quality in terms of hardware, software [71] and lifespan. Additionally, due to rapid development cycles [64], increasing program size and mobile data traffic, there is a higher difficulty in determining whether computational infrastructures and systems are first class. In fact, companies have difficulties in successfully implementing mobile systems [64]. As technology evolves, hardware and software become more complex to evaluate.

Metrics and guidelines are indispensable as they allow for consistency in the development and evaluation process, as well as permitting a comparison between different implementations. Most related works focus in one category in particular, for example hardware, but not all four in conjunction (hardware, software, global system and lifespan). Since mobile applications are the future of computing, evaluations and performance analysis should be highly considered and well executed.

Several elements need to be considered when implementing mobile systems, such as:

- Limited hardware resources;
- Less lifespan than stationary devices;
- Used for a wide variety of purposes (even more so than desktops); and
- Constant change in working environments, as such they need to provide availability, flexibility and reliability.

Therefore, mobile systems need to be evaluated to reaffirm that their contributions and services are top quality. A metric can be defined as a measurement of a system characteristic. This measurement is important due to providing valuable information to guide, evaluate and compare implementations.

Performance analysis is becoming a difficult task due to emerging technologies. Since the development of a mobile system can be challenging, a performance analysis is useful because it offers predictions or feedback[22]. As illustrated in Figure 5.3, a performance analysis should be based on four key areas:

1. **Hardware analysis**: concerns the components of the system and how efficient they are in the process.

2. **Software analysis**: relates to how the mobile system manages tasks to offer quality services.

3. **Global system analysis**: refers to viewing an overall perspective of the mobile system.

4. **Lifespan analysis**: can be defined as the phase when efficiency and availability are at their maximum capacity.

Figure 5.3: Classification Scheme of Performance Analysis for Mobile Systems and their Architectures.

Thus, a performance analysis is proposed based on four categories in order to provide a general or specific feedback of the mobile system and its architecture. The following sections delve into each category more specifically, detailing theoretical metrics that will serve as guidelines to evaluate mobile systems and its infrastructures. Each metric is suggested taking into consideration the ISO/IEC 25010 for systems and software quality requirements and evaluation [43], which is a revision of the widely utilized ISO 9126 for measuring systems [1].

It is important to clarify that the metrics are not specific, and are proposed from a general perspective. Developers can opt in modifying them and adapting them to their own needs and afterwards evaluating their mobile systems based on the proposed classifications. The goal is to propose metrics, in general terms, for developers to easily adapt them to their portable systems and architectures. In addition, a line graph based on 4 dimensions will also be proposed for the developer to obtain an overview of the cost of system performance.

5.3.1 Hardware Analysis

As stated before, *Hardware* can be defined as the physical components of a computing device. The said evaluation is valuable due to providing data of its efficiency in various areas including energy consumption, which is highly interesting in research communities and one of the most limited resources in portable computing [8].

Communication is also considered to be an important element in mobile systems and wireless sensor networks [35], utilizing `WiFi` or cellular technologies such as `3G` for data transfers, however, the latter has proven to be more energy consuming than the former [8] and, thus, developers should take that highly into consideration.

The use of resources is a key factor in the success or failure of portable applications [8]. In addition, fault tolerance is a critical requirement for Wireless and Sensor and Actuator Networks [106].

Taking all of this into consideration, Figure 5.4 displays the proposed metrics in a line graph in order to evaluate a mobile system in terms of its *hardware*. This figure contains all metrics with their respective classification, in order to provide indicators for each situation.

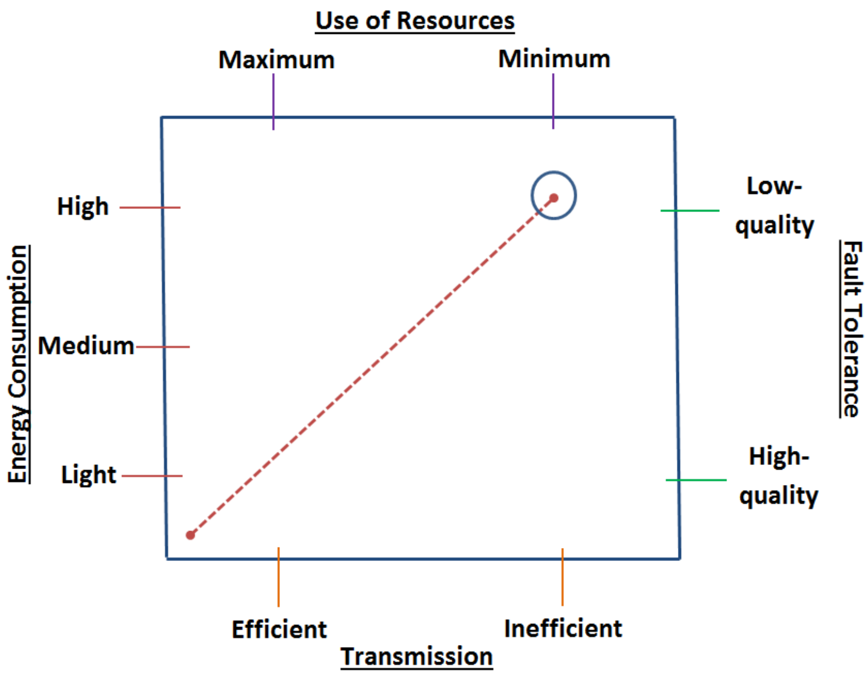


Figure 5.4: Evaluating a Mobile System in terms of Hardware.

The proposed metrics are:

Energy Consumption can be defined as energy utilized during operations. Energy efficiency is a major concern when developing portable applications [16, 48, 87, 112], garnering significant attention in the last few years [8]. Since mobile systems have limited quantity, architectures need to balance energy consumption during activity or inactivity in order to use as little as possible. This metric is classified in three categories: *light*, *medium* or *high*, measuring the *energy* utilized by hardware components during operations. Such metric is indispensable due to providing data in order to improve energy efficiency for mobile applications. It is important to highlight that many existing works have been implemented to improve energy efficiency in mobile networks but not portable applications [12]. The equation for Energy Consumption (EC) is:

$$EC = T + UR + SC \tag{5.1}$$

which signifies that it is equal to the cost of package transmission (T) plus the cost of use of resources (UR) plus system cost (SC), which the latter can be calculated by adding up the cost of all the operations being executed: $\sum_{i=1}^{SC_s} SC_i$.

Use of Resources signifies hardware and network usage in operations. This metric measures how much *resources* are being utilized. Since mobile applications have limited resources, architectures should take advantage of the said element to offer quality services, while at the same time being efficient in the process. The use of resources is classified as: *minimum* or *maximum*. Use of resources can be calculated by adding up the cost of all hardware resources (HR), ($\sum_{i=1}^{HR_s} HR_i$), and the cost of network technologies (NT), ($\sum_{i=1}^{NT_s} NT_i$), being utilized, therefore resulting in the following equation:

$$UR = HR + NT \tag{5.2}$$

Fault Tolerance can be defined as how the system (hardware) reacts to errors. This metric measures the *system performance* in response to errors from the hardware perspective and, if the system detects any, it should continue offering a quality-driven service that it is expected from it. This characteristic is categorized in: *low-quality* or *high-quality*. Fault Tolerance (FT) is equal to the cost of all system errors (SyE), ($\sum_{i=1}^{SyE_s} SyE_i$), plus the cost of the current system status (SS), therefore resulting in the following equation:

$$FT = SyE + SS \tag{5.3}$$

Transmission refers to the process of transferring data to an electronic device. As such, this metric evaluates how *efficient* the operation is. Two elements need to be considered in regards to this metric: transmission capability and energy consumption. Thus, architectures should maximize transmission capacity while at the same time utilize less energy. This metric is classified in two categories: *efficient* or *inefficient*. Transmission (T) is equal to the sum of all transmission Cost (TC) ($\sum_{i=1}^{TC_s} TC_i$) plus the cost of use of resources (UR), therefore resulting in the following equation:

$$T = TC + UR \tag{5.4}$$

5.3.2 Software Analysis

As stated before, software in Mobile Computing can be referred to the applications being executed in a portable device [76]. Its evaluation is important due to permitting awareness in system performance and reliability.

Software performance is a relevant and crucial attribute in the development process [74]. Additionally, it is considered a defining factor in purchasing mobile devices [87]. Since portable gadgets have limited resources, work-load is highly probable in its architectures and thus, should be considered in evaluations [8]. Latency is also integral, due to being related to service quality and performance. All the said elements are associated with one another, allowing an overall scope of the software being executed.

Figure 5.5 exhibits a proposed evaluation scheme based on *software* metrics.

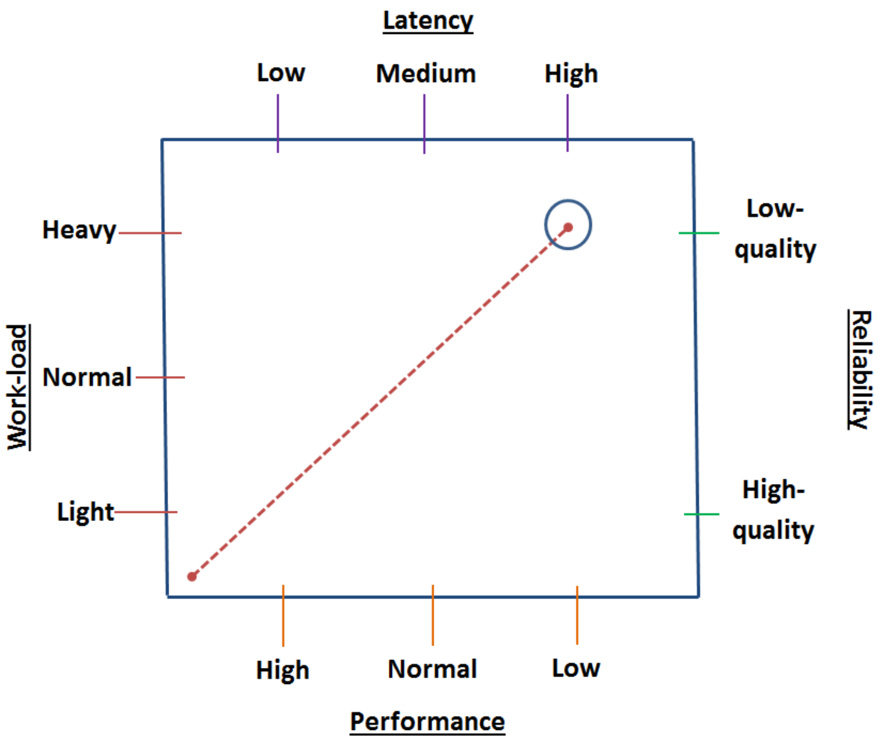


Figure 5.5: Evaluating a Mobile System in terms of Software.

The proposed metrics are:

Work-load refers to the amount of tasks to do in a certain amount of time. As such, this metric measures the amount of processing handled by the mobile system or architecture. This metric is classified as: *light*, *normal* or *heavy*. Work-load (WL) is equal to the sum of the system cost (SC), ($\sum_{i=1}^{SC_s} SC_i$), and the sum of all tasks being executed (NT), ($\sum_{i=1}^{NT_s} NT_i$), therefore resulting in the following equation:

$$WL = SC + NT \tag{5.5}$$

Latency refers to the delay the system had in order to execute a task. As such, this metric measures *time* in a certain unit (example: seconds, minutes, hours, clock cycles, others). Latency is the time spent to complete a task, which begins when the task is initialized and finishes when it is completed. Thus, Software Latency (L) is equal to the cost of all Tasks' Latency (TL), ($\sum_{i=1}^{TL_s} TL_i$), therefore resulting in the following equation:

$$L = TL \tag{5.6}$$

This metric is classified as: *low*, *medium* or *high*.

Reliability can be defined as how the services (software) reacts to errors. This metric measures the system performance in response to errors from the software perspective and, if the system detects any, it should continue offering a reliable service. This characteristic is categorized in: *low-quality* or *high-quality*. Reliability (RE) is equal to the cost of all service errors (SeE), ($\sum_{i=1}^{SE_s} SE_i$), plus the cost of the current system status (SS), therefore resulting in the following equation:

$$RE = SeE + SS. \tag{5.7}$$

Performance can be defined as the amount of work or tasks executed in a given time. It is important to note that it is not always the case that a higher work-load causes a performance decrease. This metric is classified as: *low*, *medium*, or *high*. The Performance (P) cost is equal to adding the cost of Work-load (WL) with the cost of use of resources (UR), resulting in:

$$P = WL + UR \tag{5.8}$$

5.3.3 Global System Analysis

Global evaluations are important due to providing an overall view of the system. While previous figures represented hardware and software evaluations individually, Figure 5.6 reveals an evaluation scheme based on a *global perspective* of the mobile system. All metrics chosen belong to previous evaluations (hardware and software), specifically two from each category. The metrics for hardware and software were selected based on their importance in mobile computing, therefore, offering a clear overview of the mobile system and its architecture.

The metrics proposed are the same as stated before, except that Fault Tolerance is equal to the cost of fault tolerance, from the hardware perspective (FT_H), plus the cost of reliability, from the software perspective (RE), therefore resulting in the following equation:

$$FT_g = FT_H + RE \tag{5.9}$$

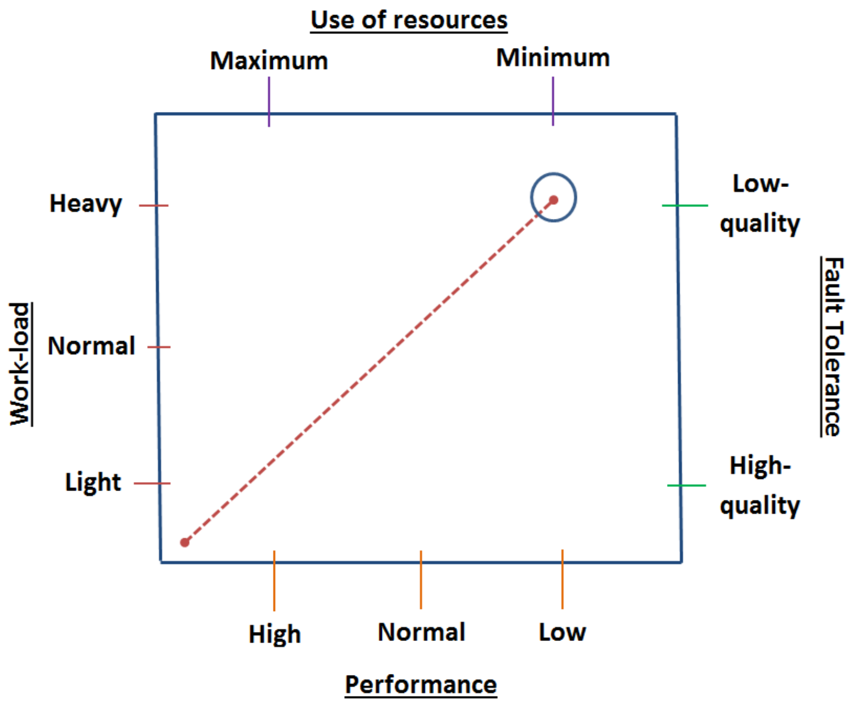


Figure 5.6: Global System Analysis.

5.3.4 Lifespan Analysis

Considering that one of the defining and most important aspects of a mobile system is its lifespan, Figure 5.7 represents the proposed line graph in order to evaluate the said element.

The proposed metrics for this area are:

Efficiency refers to the system capability to provide quality-driven services using the provided hardware and software resources. It can be calculated by dividing the cost of use of resources (UR) plus the cost of work-load (WL) with total cost of resources (TR), and multiplying by 100 to express in a percentage, resulting in the following equation:

$$E = ((UR + WL)/TR)X100 \tag{5.10}$$

As such, this metric measures the *efficiency* of the system service provided, and is classified as: *minimum* or *maximum*.

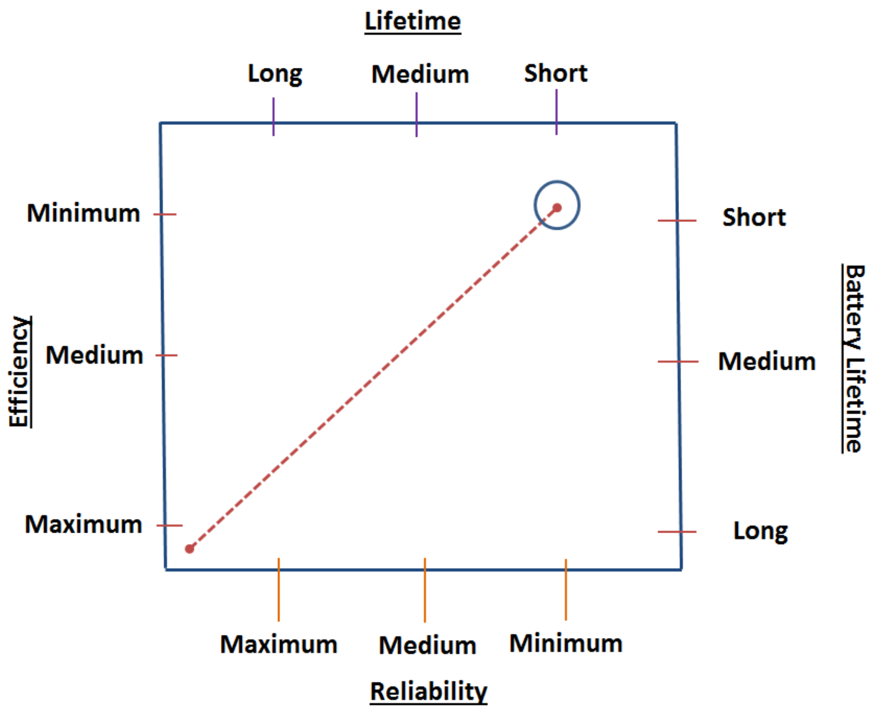


Figure 5.7: Determining the Lifespan of a Mobile System.

Lifetime measures the *time* the system is offering a quality service to users. Thus, it can be calculated by adding the cost of efficiency (E) with its cost of reliability (RE), resulting in:

$$LT = E + RE \tag{5.11}$$

This metric is classified as: *short*, *medium* or *long*.

Battery Lifetime measures the *time* the battery is maintaining the system active. As such, this metric can be calculated considering the cost of energy consumption (EC). This metric is categorized as: *short*, *medium* or *long*.

Reliability measures the *reliability* of the system and is classified as *minimum* or *maximum*. As stated before, the equation for this metric is:

$$RE = SeE + SS \tag{5.12}$$

with the result given in a percentage.

5.3.5 Summary of Proposed Metrics

Table 5.1 displays all of the metrics in their respective categories. As such, these *16* metrics in *4* categories (hardware, software, global analysis, lifespan) are worth considering in the Mobile Computing industry.

Table 5.1: Summary of Metrics

Number	Type	Metric
1	Hardware	Energy Consumption
2	Hardware	Use of Resources
3	Hardware	Fault Tolerance
4	Hardware	Transmission
5	Software	Work-load
6	Software	Latency
7	Software	Reliability
8	Software	Performance
9	Global	Use of Resources
10	Global	Fault Tolerance
11	Global	Work-load
12	Global	Performance
13	Lifespan	Efficiency
14	Lifespan	Lifetime
15	Lifespan	Battery Lifetime
16	Lifespan	Reliability

5.4 Guidelines

It is important to detail how the proposed metrics can be utilized in the development and evaluation process. As stated before, architectures are vital for the success of a mobile system. In addition, they are a *key* element in Mobile Computing. Thus, guidelines are offered in order to implement a well-developed infrastructure.

5.4.1 Architecture

A modular architecture can be implemented with three modules:

Software Module refers to the application and services offered.

Network Module refers to the communication and protocols utilized.

Hardware Module refers to the hardware components utilized.

For each module, metrics can be applied in order to validate such components. For example, in the *software module*, the proposed *software* metrics such as *work-load*, *performance*, *latency* and *reliability* can be utilized. In the *hardware module*, the following metrics can be applied: *energy consumption*, *efficiency*, *lifetime* and *battery lifetime*. Finally in the network module, *use of resources*, *transmission* and *fault tolerance* can be utilized in order to evaluate the network technologies and protocols. Developers can opt in utilizing only one type of metric, such as *software*, *hardware* or *lifespan*, to validate their mobile systems.

5.4.2 Evaluation

In addition to the development process, evaluating a mobile system is just as important. As such, there are 3 evaluation methods aimed at developers who want to verify their mobile systems are offering quality services.

Location This method is based on the environment in which the mobile system will be evaluated. Location is categorized in *internal* or *external*.

- **Internal**: refers to mobile systems evaluated in laboratories or internally in order to verify correct functioning.
- **External**: signifies evaluating mobile systems outdoors or externally to validate their quality services in handling certain obstacles that may interfere with its operations.

Data This method is based on the data type which will be used during the evaluation. Data type is categorized in *synthetic* or *real*.

- **Synthetic**: consists in evaluating the mobile system with synthetic information generated by the user.
- **Real**: refers to evaluating the mobile system with real and concrete data in order to assure the system functions correctly.

Modeling This method is based on how the mobile system will be evaluated. Modeling is categorized in *simulation* or *real*.

- **Simulation**: consists in imitating real-life environments or situations in order to validate mobile systems.
- **Real**: refers to validating implementations in real-life situations, environments, and devices.

It is important to state that a well-defined methodology incorporates several methods.

6

Related Works

This chapter is divided into two sections: Traditional and Mobile systems; and Metrics.

6.1 Mobile and Traditional Systems

The following works are based on architectures for traditional and mobile systems. The projects were divided into three subsections: Architectures for Distributed Systems; Addressing challenges of Mobile Computing; and Simplifying the development process of a specific application.

6.1.1 Proposed Architectures for Distributed Systems

Ehab Al-Shaer *et al.* (1999) [3] proposed and implemented a monitoring architecture called `HiFi` to manage large-scale distributed systems. The objective of this project was to implement a scalable, dynamic, flexible and transparent infrastructure that is capable of correcting distributed applications.

Schahram Dustdar *et al.* (2003) [24] detailed a three-layered architecture for distributed and collaborative mobile systems. The researchers emphasized that an infrastructure for portable applications must be flexible and adaptable to new requirements. Thus, the objective was to propose an architecture, for distributed mobile systems, that permits a collaboration between users.

Sebastien Leriche *et al.* (2007) [66] proposed an architecture that provides flexibility and simplicity in developing, updating and maintaining distributed large-scale applications.

6.1.2 Addressing Challenges of Mobile Computing

Thomas Phan *et al.* (2001) [91] developed a *middleware* to manage a high number of users in a mobile-internet application. The objective was to improve the infrastructure by providing higher scalability.

Gilda Pour *et al.* (2006) [92] implemented an architecture, based on mobile agents, to resolve challenges such as low bandwidth and network failures. Other aspects taken into consideration during the project include flexibility, reliability, extensibility, and maintainability of mobile systems.

Yuri Natchetoi *et al.* (2008) [82] proposed a *Service Oriented Architecture* (`SOA`) for mobile applications to minimize data transmission and storage.

Ashiq Khan *et al.* (2011) [49] proposed a configurable and flexible architecture for mobile networks. The said model aims to reduce costs and be a basis for resolving future challenges.

Hong-Han Shuai *et al.* (2011) [108] developed an architecture called `MobiUP` for mobile devices, which aims to solve the challenge of limited-bandwidth for video streaming. The platform was developed to be flexible when implementing codecs. The experiments proved that it improves video quality and reduces data transmission.

Anas Showk *et al.* (2012) [107] proposed a parallel architecture to minimize energy consumption when utilizing *Long Term Evolution* (`LTE`) technologies. The infrastructure balances between energy consumption and task-load. The results show a 39% improvement when utilizing one, two, three, and even four cores.

Rafaa Tahar *et al.* 2012 [111] developed a flexible architecture for mobile nodes utilizing multiple interfaces. The infrastructure enables protocol implementations in wireless communication areas. Furthermore, the platform minimizes energy consumption during data transmission. The experiments proved that the architecture is flexible for improvements and efficient in energy consumption.

Bimal Aklesh Kumar (2014) [53] proposed a flexible architecture to address web-development challenges in mobile devices. The researcher stressed that portable gadgets have processing and storage limitations, resulting in the necessity of flexible and reusable infrastructures. The experiments demonstrated that the architecture reduces resources consumed on a portable device.

Leibo Liu *et al.* (2014) [69] proposed an architecture called `Ver-Comp` to address the limitations on non-smart devices, such as cameras or printers. The infrastructure has the purpose of addressing challenges related to processing power and wireless communications, giving more independence to these devices. A variety of techniques were proposed for the software, hardware, and *middleware*. Furthermore, the researchers emphasized that the platform will allow non-smart devices to communicate with their counterparts and, thus, provide innovative applications and services.

Jingchu Liu *et al.* (2014) [68] presented an infrastructure, named `CONCERT`, which addresses challenges related to 5G networks. According to the authors, the number of network connections is increasing considerably and, thus, a flexible and scalable infrastructure is necessary to minimize latency, reduce costs, and be efficient.

Jianming Zhang *et al.* (2015) [132] proposed an architecture for `5G` networks, which was designed to be flexible, scalable, improve energy consumption and reduce costs. The researchers emphasized that the number of mobile devices is increasing daily and, thus, a platform that adapts itself to modern technologies and environments is required.

6.1.3 Simplifying the Development Process of a Specific Application

Iara Augustin *et al.* (2002) [6] and Thomas Kunz *et al.* (1999) [55] implemented an architecture to simplify the development of adaptable mobile systems. The researchers concluded that infrastructures focusing on adaptability, mobility and flexibility are necessary.

R. Mukkamala *et al.* (2004) [78] proposed an architectural model to reuse system components. The infrastructure consists in a layer responsible for selecting, reconfiguring and integrating reused elements. Thus, the architecture provides a faster and simpler system development. The platform was applied in medical fields and aerospace-structural modeling.

Mohand Tahar Chebbine *et al.* (2005) [13] developed an architecture to adapt web content for any mobile device. The infrastructure was implemented in a system called *Content Adaptation System for Heterogeneous Environments* (`CASHE`) , which provides adapted Internet content for several mobile devices. The objective was to prevent programmers from developing multiple versions of the same code for portable devices, thus, reducing time and increasing efficiency.

Zong Wang *et al.* (2009) [119] proposed an architecture for *Software Defined Radios* (`SDR`). The infrastructure provides flexibility, ease in the development process and low energy consumption.

Enric Pastor *et al.* (2010) [88] proposed a flexible and reusable architecture to simplify the development process for remote-sensing applications. The system, on which the architecture was validated, is called `Red Eye` and consists in detecting, analyzing and controlling forest fires in the Mediterranean Sea.

Chris Lu *et al.* (2011) [70] presented an architecture to develop and update an educational mobile game. The platform was implemented to be lightweight, flexible, scalable and capable of managing limited resources in portable devices.

Stephen S. Nestinger *et al.* (2011) [84] developed an architecture, called `Mobile-R`, to implement Multi-robot systems. The infrastructure was developed to be flexible and reconfigurable for maintaining control mechanisms. A *middleware* was developed to provide an increased fault-tolerance, simplicity and adaptability. The platform was validated utilizing the following virtual robots: `Khepera III` and `Pioneer2DX`. Both proved to be successful in the experiments.

Xiping Hu *et al.* (2013) [40] implemented an architecture to simplify the development of applications related to natural disasters, such as earthquakes, tsunamis or hurricanes. The infrastructure consisted in two layers: Services and Applications. The Service Layer is based on a `SOA` and the Application Layer is oriented towards users and social networks.

Roberto S. Silva Filho *et al.* (2015) [28] proposed an architecture to ease mobile system development in field engineering. The developers concluded that these applications are not geared towards portability. Thus, the proposed infrastructure provides contributions such as: *Application Program Interfaces* (`APIs`) to develop mobile systems, data synchronization, and a *middleware* that permits integrating different applications in order to improve the service quality.

6.2 Metrics

In recent years there have been works related to defining metrics [27, 71, 46], in terms of hardware and software, to address such issues.

Hanny Fauzia *et al.* (2014) [27] proposed metrics, based on ISO/IEC 25010, for mobile thick client architectures that uses web `API`, focusing specifically on performance efficiency and reliability. The authors concluded that mobile applications are being used considerably and, as such, quality models are required in order to offer better services.

Daniel Lübke (2015) [71] proposed static code metrics in order for developers to obtain feedback and readjust their architectures when necessary. The authors concluded that few software metrics have been proposed for guiding architectural decisions.

Minho Ju *et al.* (2016) [46] proposed two metrics in particular, performance and power consumption of a mobile system, to evaluate a simulation framework. The authors stated that both categories are indispensable for portable applications, due to allowing an efficient design and optimization.

Carmen B. Rodriguez-Estrello *et al.* (2007) [99] proposed performance metrics in Mobile Wireless Communication Networks taking into consideration both Resource Insufficiency and Link Unreliability. The authors concluded that these metrics will serve as guidelines and models for future developments of mobile networks.

Yao Xuanzheng *et al.* (2016) [122] proposed an enhanced model for performance analysis of estimation algorithms. The authors stated that performance analysis for that specific field was simple and incomplete. Thus, they proposed a model that permits the user to apply suitable metrics to evaluate the algorithms.

Ramin Izadpanah *et al.* (2016) [45] presented a new approach for performance analysis of non-blocking algorithms. The developers proposed metrics to simplify the investigation of the said algorithms and applications. These metrics were utilized in conjunction with hardware metrics in order to provide vital information. The authors concluded that these evaluation schemes can be utilized by developers in order to analyze their algorithms, as well as to compare different implementations.

Deepak Kumar *et al.* (2015) [54] presented a performance analysis in order to evaluate different multiprocessor architectures for radio signal processing. The authors proposed metrics such as Latency and Efficiency in order to analyze several infrastructures. Because of the proposed metrics, the authors were able to compare implementations and conclude which is better-suited for different situations.

Claudia Melania Chituc (2015) [14] proposed a methodology to monitor Service Level Agreements in a cloud environment. The author stated that performance analysis metrics are required in order to perform such an action, as well as an architecture, which will be tested to analyze its scalability.

Table 6.1 shows that the majority of the aforementioned related works were based on hardware and software. However, the developers did not focus on *lifespan*, a *key* characteristic of mobile systems. In addition, the developers proposed metrics based on a specific type of mobile system, but not in general terms.

Table 6.1: Comparison of Related Works with the Proposed Metrics

Related Work	**Number of Metrics**	**Metrics Proposed**	**Categories**	**Objective**
[27]	8	Functional Stability, Reliability, Performance efficiency, Operability, Security, Compatibility, Maintainability, Transferability	Hardware and Software	Mobile thick client (Web API)
[71]	5	Activity type counts, Total basic activity count, Total structured activity count, Extension activity count, activity distribution	Software	Software Architecture
[46]	2	Performance and Energy Consumption	Hardware	Simulation Framework (Web browsing)
[99]	2	Resource insufficiency and Link unreliability	Network	Network
[45]	6	Announcement count, Helped announcement, RC_remove_descr, RC_offload, RC_watch_fail, HP_watch_fail	Hardware and Software	Non-blocking algorithms
[54]	6	Latency, Efficiency, Load Imbalance, Buffer requirement, Scalability, Flexibility	Hardware	Multiprocessor Architecture
Proposed Metrics	**16**	**Energy Consumption, Use of Resources, Fault Tolerance, Transmission, Work-load, Latency, Reliability, Performance, Use of Resources (Global), Fault Tolerance (Global), Work-load (Global), Performance (Global), Efficiency, Lifetime, Battery Lifetime, Reliability (Lifespan)**	**Hardware, Software, Global, Lifespan**	**Any type of mobile system**

7 Conclusion

Mobile systems are expanding both in users and technologies. Developers are implementing portable applications and architectures rapidly. The evaluation of `myBee` in a real-world environment proves the efficiency of architectures to monitor bee colonies. Based on `myBee`, it was acknowledged that mobile systems can be executed in dynamic environments, which results in a complex development process. Thus, this master's project had two objectives: propose a flexible architecture that simplifies the development and maximizes the lifespan of any mobile system; and propose metrics in order to validate portable systems.

The architecture is composed of 3 layers. The Service Layer manages the operations of the system. The Network Layer handles the protocols utilized to transmit the data. The Hardware Layer is composed of the physical components of the mobile devices. The *middleware* is considered to be one of the most important characteristics of the proposed infrastructures due to permitting flexibility and scalability, which simplifies and streamlines the development process. The *middleware* is also based on interoperability, allowing communication and easy access between each component. In addition, the *Management* module permits the user to add or configure protocols, operations or physical components of the mobile system.

Several infrastructures have been developed over the last few years. However, there are few guidelines or schemes to evaluate such implementations in order to reassure that their top quality. Therefore, a performance analysis is proposed based on four key areas: *hardware*, *software*, *global perspective*, and *lifespan*. All of the said categories utilize variables that developers need to take into consideration when dealing with mobile systems and architectures. It is recommended to apply the said contributions in order to validate such implementations and offer quality services and technologies.

The contributions of this master's project can be classified as follows:

- Technical: `myBee` was well-documented in order to better-comprehend the system for future modifications such as sensors to analyze sound and weight of the beehive.
- Scientific: two proposals - a layer-based architecture for mobile systems and metrics in order to compare and validate implementations.

Future works include the following:

- Applying the proposed theoretical metrics in a real-world environment.
- Applying the *generic architecture* in other systems in order to validate its efficiency and flexibility.
- Updating the Machine Learning module in order to detect beehive abandonment.

REFERENCES

[1] Software engineering - product quality, ISO/IEC 9126-1. Technical report, International Organization for Standardization, 2001.

[2] Jorge Aguiar, Filipe Portela, Manuel Filipe Santos, José Machado, António Abelha, Álvaro Silva, Fernando Rua, and Filipe Pinto. Pervasive information systems to intensive care medicine - technology acceptance model. In *Proceedings of the 15th International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 177–184, 2013.

[3] E. Al-Shaer, H. Abdel-Wahab, and K. Maly. Hifi: a new monitoring architecture for distributed systems management. In *Proceedings of the 19th IEEE International Conference on Distributed Computing Systems, 1999.*, pages 171–178, 1999.

[4] Miguel A Altieri. The ecological role of biodiversity in agroecosystems. *Agriculture, Ecosystems & Environment*, 74(1):19–31, 1999.

[5] Millennium Ecosystem Assessment et al. *Ecosystems and human well-being*, volume 5. Island Press Washington DC, 2005.

[6] I. Augustin, A.C. Yarnin, J.L.V. Barbosa, and C.F.R. Geyer. Isam, a software architecture for adaptive and distributed mobile applications. In *Proceedings of the International Symposium on Computers and Communications.*, pages 333–338, 2002.

[7] Iara Augustin, Adenauer Yamin, Cláudio Fernando Resin Geyer, and Jorge LV Barbosa. Requisitos para o projeto de aplicações móveis distribuídas. In *VII Congreso Argentino de Ciencias de la Computación*, 2001.

[8] Javier Berrocal, Jose Garcia-Alonso, Cristina Vicente-Chicote, Juan HernÃ¡ndez, Tommi Mikkonen, Carlos Canal, and Juan M. Murillo. Analysing the resource consumption patterns of mobile applications in early development phases. *Pervasive and Mobile Computing*, pages –, 2016.

[9] R.E. Bonney. *Beekeeping: A Practical Guide.* Storey Communications, 1993.

[10] N. Bradbear, Food, and Agriculture Organization of the United Nations. *Bees and their role in forest livelihoods: a guide to the services provided by bees and the sustainable harvesting, processing and marketing of their products.* Non-wood forest products. Food and Agriculture Organization of the United Nations, 2009.

[11] R. C. R. Camargo, F. M. Pereira, and M. T. Lopes. *Sistema de Produçao de Mel.* EMBRAPA Meio Norte, 2002.

[12] C. A. Chan, W. Li, S. Bian, C. L. I, A. F. Gygax, C. Leckie, M. Yan, and K. Hinton. Assessing network energy consumption of mobile applications. *IEEE Communications Magazine*, 53(11):182–191, November 2015.

[13] M.T. Chebbine, A. Obaid, S. Chebbine, and R. Johnston. Internet content adaptation system for mobile and heterogeneous environments. In *IFIP International Conference on Wireless and Optical Communications Networks.*, pages 346–350, March 2005.

[14] C. M. Chituc. Towards a methodology for trade-off analysis in a multi-cloud environment considering monitored qos metrics and economic performance assessment results. In *Proceedings of the International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 479–482, November 2015.

[15] A.S. Chowdhury and M. Gregory. Performance evaluation of heterogeneous network for next generation mobile. In *International Conference on Computers and Information Technology.*, pages 100–104, December 2009.

[16] L. Codrescu, W. Anderson, S. Venkumanhanti, M. Zeng, E. Plondke, C. Koob, A. Ingle, C. Tabony, and R. Maule. Hexagon dsp: An architecture optimized for mobile multimedia and communications. *IEEE Micro*, 34(2):34–43, March 2014.

[17] Sandun Dasanayake, Jouni Markkula, Sanja Aaramaa, and Markku Oivo. Software architecture decision-making practices and challenges: An industrial case study. In *Proceedings of the Australasian Software Engineering Conference*, pages 88–97, September 2015.

[18] José Eduardo Mendes de Figueiredo, Vera Maria Benjamim Werneck, and Rosa Maria EM da Costa. Simulando a dinâmica populacional de uma colmeia para o ensino de biologia. *Revista Brasileira de Informática na Educação*, 21(01):106–120, 2013.

[19] Pompílio Vieria de Souza. Dicionário de apicultura: ABC do apicultor. In *XIII Congresso Brasileiro de Apicultura*, Florianópolis - SC - Brasil, November 2000.

[20] G Deepak and BS Pradeep. Challenging issues and limitations of mobile computing. *International Journal of Computer Technology and Applications*, 3(1):pages 177–181, 2012.

[21] G. Delac, M. Silic, and J. Krolo. Emerging security threats for mobile platforms. In *Proceedings of the International Convention MIPRO*, pages 1468–1473, May 2011.

[22] Antinisca Di Marco and Cecilia Mascolo. Performance analysis and prediction of physically mobile systems. In *Proceedings of the 6th international workshop on Software and performance*, pages 129–132. ACM, 2007.

[23] Winston Edson Dunham. Hive temperatures for each hour of a day. *Ohio Journal of Science*, 31(3), 1926.

[24] S. Dustdar and H. Gall. Architectural concerns in distributed and mobile collaborative systems. In *Proceedings of the Eleventh Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2003.*, pages 475–483, February 2003.

[25] R. C. Dutra, H. F. Moraes, and C. L. Amorim. Interest-centric mobile ad hoc networks. In *NCA'2012, Proceedings of the International Symposium onNetwork Computing and Applications (NCA)*, pages 130–138. IEEE, August 2012.

[26] J. Edmondson, W. Anderson, J. Gray, J. P. Loyall, K. Schmid, and J. White. Next-generation mobile computing. *IEEE Software*, 31(2):44–47, March 2014.

[27] H. Fauzia, I. H. Laksmiwati, and B. Hendradjaya. A quality model for mobile thick client that utilizes web api. In *Proceedings of the International Data and Software Engineering Conference*, pages 1–6, November 2014.

[28] R. S. S. Filho and A. Tewari. Distributed architecture for mobile contextual integrated field work applications. In *Proceedings of the IEEE International Conference on Mobile Services*, pages 81–88, June 2015.

[29] D.W. Fitzgerald, F.E. Murphy, W.M.D. Wright, P.M. Whelan, and E.M. Popovici. Design and development of a smart weighing scale for beehive monitoring. In *Irish Signals and Systems Conference.*, pages 1–6, June 2015.

[30] M. Giammarini, E. Concettoni, C. C. Zazzarini, N. Orlandini, M. Albanesi, and C. Cristalli. Beehive lab project - sensorized hive for bee colonies life study. In *Proceedings of the International Workshop on Intelligent Solutions in Embedded Systems*, pages 121–126, October 2015.

[31] C. Gibb, D. Ainsworth, N. Pratt, R. Sessa, Food and Agriculture Organization of the United Nations, and Youth and United Nations Global Alliance. *The Youth Guide to Biodiversity*. Food and Agriculture Organization of the United Nations, 2013.

[32] D. Egea Gondolfo and A. Faustino da Silva. Network model addressed by interests in kernel level. *IEEE Latin America Transactions*, 14(2):998–1004, February 2016.

[33] D. Goswami. Mobile computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(9):pages 846–855, September 2013.

[34] W. Goudalo and C. Kolski. Towards advanced enterprise information systems engineering - solving resilience, security and usability issues within the paradigms of socio-technical systems. In *Proceedings of the 18th International Conference on Enterprise Information Systems - Volume 2: ICEIS,*, pages 400–411, 2016.

[35] V. Goudar and M. Potkonjak. Fault-tolerant and low-power sampling schedules for localized basns. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(1):86–95, March 2013.

[36] P. Guinness and B. Walpole. *Environmental Systems and Societies for the IB Diploma*. Cambridge University Press, 2012.

[37] T. Hague, J. A. Marchant, and N. D. Tillett. Autonomous robot navigation for precision horticulture. In *Proceedings of the IEEE International Conference on Robotics and Automation.*, volume 3, pages 1880–1885, April 1997.

[38] Rachel Harrison, Derek Flood, and David Duce. Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science*, 1(1):1–16, 2013.

[39] M. Hilmi, N. Bradbear, D.J. Mejía-Loríо, Food, Agriculture Organization of the United Nations. Rural Infrastructure, and Agro-Industries Division. *Beekeeping and Sustainable Livelihoods*. Diversification booklet. Rural Infrastructure and Agro-Industries Division, Food and Agriculture Organization of the United Nations, 2011.

[40] X. Hu, V. C. M. Leung, W. Du, B. C. Seet, and P. Nasiopoulos. A service-oriented mobile social networking platform for disaster situations. In *Proceedings of the Hawaiin International Conference on System Sciences*, pages 136–145, January 2013.

[41] IEEE. IEEE standard glossary of software engineering terminology. *IEEE Std 610.12-1990*, pages 1–84, December 1990.

[42] ISO. *ISO 9241-11: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs): Part 11: Guidance on Usability.* 1998.

[43] ISO/IEC. ISO/IEC 25010 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. Technical report, 2010.

[44] ISO/IEC/IEEE. ISO/IEC/IEEE systems and software engineering – architecture description. *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*, pages 1–46, December 2011.

[45] R. Izadpanah, S. Feldman, and D. Dechev. A methodology for performance analysis of non-blocking algorithms using hardware and software metrics. In *Proceedings of the International Symposium on Real-Time Distributed Computing (ISORC)*, pages 43–52, May 2016.

[46] Minho Ju, Hyeonggyu Kim, and S. Kim. Mofysim: A mobile full-system simulation framework for energy consumption and performance analysis. In *Proceedings of the International Symposium on Performance Analysis of Systems and Software*, pages 245–254, April 2016.

[47] H. Kandjani, A. E. Andargoli, A. Mohtarami, and R. Shokoohmand. A conceptual framework to classify strategic information systems planning methodologies. In *Proceedings of the 15th International Conference on Enterprise Information Systems - Volume 2: ICEIS,*, pages 190–196, 2013.

[48] Ching Kin Keong, Koh Tieng Wei, Abdul Azim Abd. Ghani, and Khaironi Yatim Sharif. Toward using software metrics as indicator to measure power consumption of mobile application: A case study. In *Proceedings of the Malaysian Software Engineering Conference*, pages 172–177, December 2015.

[49] A. Khan, D. Jurca, K. Kozu, W. Kellerer, and M. Yabusaki. The reconfigurable mobile network. In *Proceedings of the IEEE International Conference on Communications Workshops.*, pages 1–5, June 2011.

[50] Douglas Santiago Kridi, Carlos Giovanni Nunes de Carvalho, and Danielo Gonçalves Gomes. A predictive algorithm for mitigate swarming bees through proactive monitoring via wireless sensor networks. In *Proceedings of the 11th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks*, pages 41–47, New York, NY, USA, 2014. ACM.

[51] John Krogstie, Kalle Lyytinen, Andreas Lothe Opdahl, Barbara Pernici, Keng Siau, and Kari Smolander. Research areas and challenges for mobile information systems. *International Journal of Mobile Communications*, 2(3):220–234, 2004.

[52] A. Kumar. Third generation personal communication systems. In *IEEE International Conference on Personal Wireless Communications.*, pages 313–318, February 1996.

[53] B. A. Kumar. Layered architecture for mobile web based applications. In *Proceedings of the Asian-Pacific World Congress on Computer Science and Engineering*, pages 1–6, November 2014.

[54] D. Kumar, D. Deb, and R. Mamgain. Analysis of different multiprocessor architectures for radar signal processing with performance metrics. In *Proceedings of the International Radar Conference*, pages 289–294, October 2015.

[55] Thomas Kunz and James P Black. An architecture for adaptive mobile applications. In *Proceedings of the International Conference on Wireless Communications.*, 1999.

[56] A. Kviesis and A. Zacepins. System architectures for real-time bee colony temperature monitoring. *Procedia Computer Science*, 43:86–94, 2015.

[57] A. Kviesis and A. Zacepins. Application of neural networks for honey bee colony state identification. In *ICCC'2016, Proceedings of the 17th International Carpathian Control Conference*, pages 413–417. IEEE, May 2016.

[58] Armands Kviesis and Aleksejs Zacepins. System architectures for real-time bee colony temperature monitoring. *Procedia Computer Science*, 43:86 – 94, 2015.

[59] Armands Kviesis, Aleksejs Zacepins, Mahmut Durgun, and Saban Tekin. Application of wireless sensor networks in precision apiculture. In *Proceedings of the Engineering for Rural Development, 14*, pages 440–445, 2015.

[60] Armands Kviesis, Aleksejs Zacepins, Gatis Riders, et al. Honey bee colony monitoring with implemented decision support system. In *Proceedings of the International Scientific Conference*. Latvia University of Agriculture, 2015.

[61] Emilio A Laca. Precision livestock production: tools and concepts. *Revista Brasileira de Zootecnia*, 38:123–132, 2009.

[62] Patrick Lavelle and Asmeret Asefaw Berhe. Nutrient cycling. *Ecosystems and Human Well-Being: Current State and Trends: Findings of the Condition and Trends Working Group*, 1:331–353, 2005.

[63] E. Lawrence, B. Pernici, and J. Krogstie. *Mobile Information Systems: IFIP TC 8 Working Conference on Mobile Information Systems (MOBIS) 15-17 September 2004, Oslo, Norway*. IFIP Advances in Information and Communication Technology. Springer US, 2006.

[64] Christine Legner, Nils Urbach, and Christoph Nolte. Mobile business application for service and maintenance processes: Using ex post evaluation by end-users as input for iterative design. *Information and Management*, pages 1–15, 2016.

[65] R. Lent. Improving the location of mobile agents on self-aware networks. In *Proceedings of the Tenth International Conference on Enterprise Information Systems - Volume 5: ICEIS,*, pages 239–242, 2008.

[66] S. Leriche and J. Arcangeli. Flexible architectures and agents for adaptive autonomic systems. In *International Workshop on Engineering of Autonomic and Autonomous Systems, 2007.*, pages 99–106, March 2007.

[67] Qing Li and G. Clark. Mobile security: A look ahead. *IEEE Security and Privacy.*, 11(1):78–81, January 2013.

[68] J. Liu, T. Zhao, S. Zhou, Y. Cheng, and Z. Niu. Concert: a cloud-based architecture for next-generation cellular systems. *IEEE Wireless Communications*, 21(6):14–22, December 2014.

[69] L. Liu, W. Zhu, S. Yin, E. Y. Tang, and P. Peng. An uneven-dual-core processor based mobile platform for facilitating the collaboration among various embedded

electronic devices. *IEEE Transactions on Consumer Electronics*, 60(1):137–145, February 2014.

[70] C. Lu, M. Chang, Kinshuk, E. Huang, and C. W. Chen. Architecture and collaborations among agents in mobile educational game. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 556–560, March 2011.

[71] Daniel Lübke. Using metric time-lines for identifying architecture shortcomings in process execution architectures. In *Proceedings of the Second International Workshop on Software Architecture and Metrics*, pages 55–58, Piscataway, NJ, USA, 2015. IEEE Press.

[72] X. Ma, S. Yang, W. Jin, Y. Zhao, H. Ma, and W. Luan. Construction of a recombinant major royal jelly protein 1 (MRJP1) expression vector for mammary gland-specific expression. In *International Conference on Biomedical Engineering and Biotechnology (ICBEB)*, pages 988–991, May 2012.

[73] Qusay H. Mahmoud. Integrating mobile devices into the computer science curriculum. In *Frontiers in Education Conference.*, pages S3E–17–S3E–22, October 2008.

[74] D. Maplesden, E. Tempero, J. Hosking, and J. C. Grundy. Performance analysis for object-oriented software: A systematic mapping. *IEEE Transactions on Software Engineering*, 41(7):691–710, July 2015.

[75] Mari Matinlassi, Eila Niemelä, and Liliana Dobrica. *Quality-driven architecture design and quality analysis method. A Revolutionary Initiation Approach to a Product Architecture.* VTT Technical Research Centre of Finland, 2002.

[76] I. J. Mojica, B. Adams, M. Nagappan, S. Dienst, T. Berger, and A. E. Hassan. A large-scale empirical study on software reuse in mobile apps. *IEEE Software*, 31(2):78–86, March 2014.

[77] C. Mucignat-Caretta. *Neurobiology of Chemical Communication.* Frontiers in Neuroscience. CRC Press, 2014.

[78] R. Mukkamala, R. Ramesh, N. Agarwal, M. Bandagadde, and S. Myadam. Reusability: a technique for the design of hums architectures. In *The Digital Avionics Systems Conference, 2004.*, volume 2, pages 7.E.2–71–10, October 2004.

[79] F. Edwards Murphy, E. Popovici, P. Whelan, and M. Magno. Development of an heterogeneous wireless sensor network for instrumentation and analysis of beehives. In *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference*, pages 346–351, May 2015.

[80] F.E. Murphy, M. Magno, L. O'Leary, K. Troy, P. Whelan, and E.M. Popovici. Big brother for bees - energy neutral platform for remote monitoring of beehive imagery and sound. In *IEEE International Workshop on Advances in Sensors and Interfaces.*, pages 106–111, June 2015.

[81] F.E. Murphy, B. Srbinovski, M. Magno, E.M. Popovici, and P.M. Whelan. An automatic, wireless audio recording node for analysis of beehives. In *Irish Signals and Systems Conference.*, pages 1–6, June 2015.

[82] Yuri Natchetoi, Viktor Kaufman, and Albina Shapiro. Service-oriented architecture for mobile applications. In *Proceedings of the International Workshop on Software Architectures and Mobility*, pages 27–32, New York, NY, USA, 2008. ACM.

[83] A. Neighbour. *The apiary; or, Beses, bee-hives and bee culture*. 1866.

[84] S. S. Nestinger and H. H. Cheng. Mobile-r: A reconfigurable cooperative control platform for rapid deployment of multi-robot systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 52–57, May 2011.

[85] O. Noran. Enhancing enterprise information systems synergy in disaster management. In *Proceedings of the 15th International Conference on Enterprise Information Systems - Volume 3: ICEIS,*, pages 275–283, 2013.

[86] Masoud Nosrati, Ronak Karimi, and Hojat Allah Hasanvand. Mobile computing: principles, devices and operating systems. *World Applied Programming*, 2(7):399–408, 2012.

[87] Sudeep Pasricha, Brad K. Donohoo, and Chris Ohlsen. A middleware framework for application-aware and user-specific energy optimization in smart mobile devices. *Pervasive and Mobile Computing*, 20:47 – 63, 2015.

[88] E. Pastor, M. Sole, J. Lopez, P. Royo, and C. Barrado. Helicopter-based wildfire monitoring system software architecture. In *IEEE Aerospace Conference.*, pages 1–18, March 2010.

[89] Barbara Pernici, editor. *Mobile Information Systems: Infrastructure and Design for Adaptivity and Flexibility.* Springer-Verlag New York, Incorporation., Secaucus, NJ, USA, 2006.

[90] Twan Peters, Öykü Iş, Olgerta Tona, and Aleš Popovič. How system quality influences mobile bi use. *International Journal on Information Management*, 36(5):773–783, October 2016.

[91] Thomas Phan, Richard Guy, and Rajive Bagrodia. A scalable, distributed middleware service architecture to support mobile internet applications. In *Proceedings of the Workshop on Wireless Mobile Internet*, pages 27–33, New York, NY, USA, 2001. ACM.

[92] G. Pour and N. Laad. Enhancing the horizons of mobile computing with mobile agent components. In *IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse, and IEEE/ACIS International Conference on Computer and Information Science, 2006.*, pages 225–230, July 2006.

[93] M. Rabaey, E. Vandijck, K. Vandenborre, H. Tromp, and M. Timmerman. Business process embedded information systems - for flexibility and adaptability. In *Proceedings of the Eighth International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 144–150, 2006.

[94] Silvia C. Remolina and Kimberly A. Hughes. Evolution and mechanisms of long life and high fertility in queen honey bees. *AGE*, 30(2):177–185, 2008.

[95] O.A.M. Reyes, A.A.M. Avila, G. Sebastian Eslava, and G.B. Rozo. Beekeeping monitoring module. In *IEEE Colombian Workshop on Circuits and Systems.*, pages 1–6, November 2012.

[96] Hassan Reza and Narayana Mazumder. A secure software architecture for mobile computing. In *International Conference on Information Technology: New Generations (ITNG)*, pages 566–571. IEEE, 2012.

[97] Mark Richards. *Software Architecture Patterns.* O'Reilly, 1st edition, 2015.

[98] Nancy M. Rizk, Amr Ebada, and Eman. S. Nasr. Investigating mobile applications' requirements evolution through sentiment analysis of users' reviews. In *Proceedings of the International Computer Engineering Conference*, pages 123–130, December 2015.

[99] C. B. Rodriguez-Estrello, F. A. Cruz-Perez, and G. Hernandez-Valdez. System level performance metrics in mobile wireless communication networks considering both resource insufficiency and link unreliability. In *Procedings of the IEEE Global Telecommunications Conference*, pages 2612–2616, November 2007.

[100] G. Roussos, A.J. Marsh, and S. Maglavera. Enabling pervasive computing with smart phones. *IEEE Pervasive Computing.*, 4(2):20–27, January 2005.

[101] N. Rozanski and E. Woods. *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives.* Addison-Wesley, 2nd edition, 2012.

[102] A.K. Salkintzis. A survey of mobile data networks. *IEEE Communications Surveys.*, 2(3):pages 2–18, March 1999.

[103] R. Sammouda, A. Touir, Y.A. Reyad, N. Adgaba, A. Ai-Ghamdi, and S.S. Hegazy. Adapting artificial hopfield neural network for agriculture satellite image segmentation. In *International Conference on Computer Applications Technology.*, pages 1–7, January 2013.

[104] E.D. Schulze and H.A. Mooney. *Biodiversity and Ecosystem Function: With 22 Tables.* U.S. Government Printing Office, 1994.

[105] T. D. Seeley. *Ecologia da Abelha: Um Estudo de Adaptação na Vida Social.* Paixão Editores LTDA, 2006.

[106] H. Shen and Z. Li. A kautz-based wireless sensor and actuator network for real-time, fault-tolerant and energy-efficient transmission. *IEEE Transactions on Mobile Computing*, 15(1):1–16, January 2016.

[107] A. Showk, S. Traboulsi, and A. Bilgic. An energy efficient multi-core modem architecture for lte mobile terminals. In *International Conference on New Technologies, Mobility and Security.*, pages 1–6, May 2012.

[108] H. H. Shuai, D. N. Yang, W. H. Cheng, and M. S. Chen. Mobiup: An upsampling-based system architecture for high-quality video streaming on mobile devices. *IEEE Transactions on Multimedia*, 13(5):1077–1091, October 2011.

[109] D. Soares and L. Amaral. Reflections on the concept of interoperability in information systems. In *Proceedings of the 16th International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 331–339, 2014.

[110] Yao-Ting Sung, Kuo-En Chang, and Tzu-Chien Liu. The effects of integrating mobile devices with teaching and learning on students' learning performance: A meta-analysis and research synthesis. *Computers and Education*, 94:252 – 275, 2016.

[111] R. Tahar, A. Belghith, and R. Braham. A generic mobile node architecture for multi-interface heterogenous wireless link layer. In *Proceedings of the International Conference on the Network of the Future*, pages 1–5, November 2012.

[112] Z. Tang, S. Guo, P. Li, T. Miyazaki, H. Jin, and X. Liao. Energy-efficient transmission scheduling in mobile phones using machine learning and participatory sensing. *IEEE Transactions on Vehicular Technology*, 64(7):3167–3176, July 2015.

[113] M. A. M. Tardío, L. J. A. Rosado, and G. O. Bellot. Web-enabled decision support systems for precision viticulture. In *Iberian Conference on Information Systems and Technologies.*, pages 1–6, June 2012.

[114] Universidade Federal do Rio de Janeiro/Universidade Estadual de Maringá. Aplicações de sistemas móveis em controle ambiental. 2013.

[115] V.L. Uskov. Mobile software engineering in mobile computing curriculum. In *Interdisciplinary Engineering Design Education Conference.*, pages 93–99, March 2013.

[116] BJ Van Alphen and JJ Stoorvogel. A functional approach to soil characterization in support of precision agriculture. *Soil Science Society of America Journal*, 64(5):1706–1713, 2000.

[117] Upkar Varshney. Mobile and wireless information systems: applications, networks, and research problems. *Communications of the Association for Information Systems*, 12(1):11, 2003.

[118] D.H. Wall, R.D. Bardgett, V. Behan-Pelletier, J.E. Herrick, J. Six, T.H. Jones, D.R. Strong, and W.H. van der Putten. *Soil Ecology and Ecosystem Services.* Oxford University Press, 2013.

[119] Zong Wang and T. Arslan. A low power reconfigurable heterogeneous architecture for a mobile sdr system. In *IEEE International Symposium on Circuits and Systems.*, pages 2025–2028, May 2009.

[120] M.L. Winston. *The Biology of the Honey Bee.* Harvard University Press, 1991.

[121] Chengshan Xiao. Estimating velocity of mobiles in edge systems. In *IEEE International Conference on Communications.*, volume 5, pages 3240–3244, 2002.

[122] F. Yang, X. Yao, W. Zhang, and Y. Su. Research and analysis of performance evaluation metrics for estimation algorithms. In *Proceedings of the International Chinese Control Conference (CCC)*, pages 5195–5199, July 2016.

[123] Xu Yang, Xinyi Huang, and Joseph K. Liu. Efficient handover authentication with user anonymity and untraceability for mobile cloud computing. *Future Generation Computer Systems*, 62:190 – 195, 2016.

[124] J. Ye, B. Chen, Q. Liu, and Y. Fang. A precision agriculture management system based on internet of things and webgis. In *International Conference on Geoinformatics.*, pages 1–5, June 2013.

[125] A. Zacepins and T. Karasha. Web based system for the bee colony remote monitoring. In *International Conference on Application of Information and Communication Technologies*, pages 1–4, October 2012.

[126] A. Zacepins, A. Kviesis, P. Ahrendt, U. Richter, S. Tekin, and M. Durgun. Beekeeping in the future - smart apiary management. In *ICCC'2016, Proceedings of the 17th International Carpathian Control Conference*, pages 808–812. IEEE, May 2016.

[127] A. Zacepins and J. Meitalovs. Implementation of multi-node temperature measurement system for bee colonies online monitoring. In *International Carpathian Control Conference*, pages 695–698, May 2014.

[128] A. Zacepins, J. Meitalovs, V. Komasilovs, and E. Stalidzans. Temperature sensor network for prediction of possible start of brood rearing by indoor wintered honey bees. In *International Carpathian Control Conference.*, pages 465–468, May 2011.

[129] A Zacepins, E Stalidzans, and J Meitalovs. Application of information technologies in precision apiculture. In *International Conference on Precision Agriculture*, Indianapolis, Indiana, USA, July 2012.

[130] Aleksejs Zacepins, Valters Brusbardis, Jurijs Meitalovs, and Egils Stalidzans. Challenges in the development of precision beekeeping. *Biosystems Engineering*, 130:60 – 71, 2015.

[131] Aleksejs Zacepins and Toms Karasha. Application of temperature measurements for the bee colony monitoring: a review. In *Proceedings of the 12th International Scientific Conference of Engineering for Rural Development*, pages 126–131, 2013.

[132] J. Zhang, W. Xie, and F. Yang. An architecture for 5g mobile network based on sdn and nfv. In *Proceedings of the International Conference on Wireless, Mobile and Multi-Media*, pages 87–92, November 2015.