

MACABA NOBRE BENEDITO PEDRO

PERSONALIZED SENSING SYSTEM

Orientador: Prof. Dr. Paulo Mendes

Universidade Lusófona de Humanidades e Tecnologias

Escola de Comunicação, Arquitetura, Artes e Tecnologias de Informação

Lisboa

2014

MACABA NOBRE BENEDITO PEDRO

UNIVERSIDADE LUSÓFONA
| Escola de Comunicação,
Arquitetura, Artes
e Tecnologias da Informação



PERSONALIZED SENSING SYSTEM

Dissertação apresentada para obtenção do Grau de Mestre em Engenharia Informática, no Curso de Engenharia Informática e Sistemas de Informação, conferido pela Universidade Lusófona de Humanidades e Tecnologias.

Orientador: Prof. Dr. Paulo Mendes

Universidade Lusófona de Humanidades e Tecnologias

Escola de Comunicação, Arquitetura, Artes e Tecnologias de Informação-ECATI

Laboratório de Investigação em Sistemas e Tecnologias Informáticas - SITILabs

Lisboa

2014

EPIGRAPH

*Information is produced once, and then copied
many times.*

DEDICATION

To Jair and Jemima

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support of many people.

First and foremost, I would like to thank God for health, strength and patience to carry out this work.

Many thanks to my advisor, Prof. Dr. Paulo Mendes who patiently guide me through this process, helping on knowledge process giving me a lot of challenges, reading my numerous revisions and helping making some sense and made me believe that time must be fulfilled, ideas implemented and innovation applied.

Special thanks to Prof. Dr^a Rute Sofia, for allowing me get in SITI Research Laboratory and for being the first person that shows me that this project can be materialized.

My sincere gratitude to Lusófona University specially to *Departamento de Engenharia Informática e Sistemas de Informação* (DEISI), for the opportunity to be part of their learning process. All teachers, in special Dr. José Rogado, Dr. Nuno Garcia and Eng.º Sérgio Guerreiro for the great persistent investigation projects we have made. To my classmates Márcio António, Mauricio Ribeiro, Pedro Nicolau, Áurea Lopes, Nelson Morais and all the class for being part of my journey and special thanks to Ricardo Barbosa for his great friendship.

Through my university, I had the opportunity to fellowship, providing me with the financial means to complete my studies.

Last but not the least, I would like to thank my wife for all her encouragement and support, for choosing to be part in my study process, to my lovely parents, brothers and sisters, my numerous friends in special Caetano Quiar, Marina Manuel and Graça Profírio who endured this long process with me, always offering support.

RESUMO

Com o surgimento dos ‘*smartphones*’, entre os anos de 2007 e 2010, a Indústria Informática, associada principalmente com as comunicações móveis, redobrou esforços em relação aos componentes tecnológicos para os dispositivos móveis.

No entanto, as principais empresas de computação como Apple, Microsoft e Google, apresentaram-se divididas quanto a disponibilização para os desenvolvedores externos do seu pacote de desenvolvimento de software, abrindo caminho a que ferramentas externas fossem surgindo.

Com a disponibilização do ‘*Mono for Android*’, ambiente integrado de desenvolvimento para dispositivos Android, pequenos desenvolvedores têm a possibilidade de apresentar suas próprias aplicações, muitas delas permitindo ao utilizador efetuar partilha de conteúdos.

Do estudo feito sobre diversos conceitos tecnológicos, desde as redes tolerantes a atrasos, sensores embutidos em dispositivos móveis, abstração de dados, aprendizagem máquina, redes centradas em conteúdo, nasce o Personalized Sensing System (PersonalSense) com o objetivo de demonstrar a possibilidade de partilha de conteúdos multimédia, incluindo texto, imagens e vídeos, tendo por base a análise de dados recolhidos, referentes ao comportamento do utilizador sua localização e atividade no momento da partilha.

O presente trabalho pretende demonstrar a possibilidade da implementação de mecanismos de partilha de conteúdos em sistemas ubíquos móveis, com base na classificação de dados recolhidos de middleware de gestão de sensores e na capacidade de comunicação direta e de baixo custo (sem infraestrutura) dos dispositivos moveis.

Palavras-Chave: Sensores, ID3, árvores de decisão, rede, conhecimento

ABSTRACT

With the emerging of Smartphone's, especially in the years between 2007 and 2010, the computing industry, mainly associated with mobile communications, increased efforts towards technological components for the devices mobile were made.

However, the major companies such as Apple, Microsoft and Google, presented themselves divided on the provision for external developer package their software development, paving the way for external tools that were emerging.

With the release of '*Mono for Android*', integrated development environment for Android devices, smaller developers have the opportunity to submit their own applications, many of them allowing the user to make content sharing.

The study on various technological concepts, since the delay tolerant networks, sensors embedded in mobile devices, data abstraction, machine learning, content-centric networks, is born Personalized Sensing System (PersonalSense) in order to demonstrate the possibility of sharing multimedia content, including text, images and videos, based on the analysis of data collected for the behavior of the user's location and activity at the time of sharing.

This work aims to show and demonstrate the possibility of completion of sharing mechanisms for content on mobile ubiquitous systems, based upon data classification collected from managing middleware sensors and the ability to have direct communication and low cost without infrastructure of mobile devices.

Keywords: Sensors, ID3, Decision Trees, Network, knowledge

ABBREVIATIONS

API – Application Programming Interface

BP – Bundle Protocol

CCN – Content Centric Networking

CF – Compact Framework

CLI – Common Language Infrastructure

CLR – Common Language Runtime

CLS – Common Language Specification

CPU – Central Processing Unit

CSV- Comma – Separated Values

CTS – Common Type System

DTN – Delay Tolerant Networking

GPS – Global Position System

GUI – Graphical User Interface

ICN – Information Centric Networking

IDE – Integrated Development Environment

IPN – Interplanetary Internet

ISP – Internet Service Provider

JIT – Just In Time

MANET – Mobile Ad-hoc Networking

MIMO – Multiple-Input and Multiple-Output

MSIL – Microsoft Intermediate Language

OSI – Open Systems Interconnection

SDK – Software Development Kit

SITILabs – Information Systems and Technologies Research Laboratory

USB – Universal Serial Bus

USP – Ubiquitous Sensing Platforms

VANET – Vehicular Ad-hoc Networking

VES – Virtual Execution System

WiMax – Worldwide Interoperability for Microwave

Wi-Fi – Wireless Fidelity

WLAN – Wireless Local Area Network

XML – eXtensible Markup Language

INDEX

1. INTRODUCTION.....	13
1.1. MOTIVATION	16
1.2. APPLICABILITY	17
1.3. OBJECTIVES	18
1.4. STRUCTURE OF THE DISSERTATION	19
2. NETWORKING CONCEPTS	20
2.1. INFORMATION CENTRIC NETWORKING.....	21
2.2. OPPORTUNISTIC NETWORKS.....	22
2.3. SUMMARY	24
3. SENSING AND INFERENCE	26
3.1. SENSING ABSTRACTION.....	26
3.2. INFERENCE.....	27
3.2.1. REQUIREMENTS	32
3.2.2. ALGORITHMS.....	35
3.2.3. FUNCTIONAL COMPARISON	36
3.2.4. ALGORITHM IMPLEMENTATION	40
3.3. SUMMARY	42
4. PERSONALSENSE FRAMEWORK.....	43
4.1. TAGGING MODULE.....	46
4.2. SENSING MODULE.....	47
4.3. INFERENCE MODULE.....	49
4.4. NETWORKING MODULE.....	52
4.5. GRAPHICAL INTERFACE.....	52
4.5.1. CONFIGURATION INTERFACE	52
4.5.2. USER INTERFACE.....	53
4.6. SUMMARY	54
5. CONCLUSION.....	57
6. FUTURE WORK.....	59
REFERENCES.....	60

TABLE INDEX

Table 1: Communications Scenarios networks	23
Table 2: Classification table with attributes	29
Tabela 3: Machine learning algorithms	30
Tabela 4: PersonalSense description classes	45
Table 6: PersonalSense Main Features	46
Table 7: Tags for inference process.....	47

ILLUSTRATION INDEX

Figure 1: Flowchart of Personalized Sensing System Middleware	14
Figure 2: Pervasive Sensing Architecture of CitySense Project.....	15
Figure 3: DTN Scenario	16
Figure 4:Information Centric Networking.....	21
Figure 5: Interface System.....	28
Figure 6: Machine learning algorithm process	29
Figure 7: Data classification technique examples	31
Figure 8: Decision Tree example.....	33
Figure 9: implementation code for decision tree	34
Figure 10: Tree node of sensor file.....	34
Figure 11: Code to illustrate the information gain	38
Figure 12: method code to infer the best attribute.....	40
Figure 13: Accord.NET Machine learning package.....	41
Figure 14: PersonalSense Diagram classes	43
Figure 15: PersonalSense state diagram	44
Figure 16: Format of Maestro file	48
Figure 17: Decision tree	51
Figure 18: PersonalSense Settings details	52
Figure 19: PersonalSense User Interface.....	53

1. INTRODUCTION

With the development of wireless communications technologies, from Bluetooth, Wi-Fi to the new one that is fourth generation mobile communication system (4G) in association with the improved Internet connections, mobile devices have received increased attention in recent times by professionals in areas such as telecommunications, market experts, passing by researchers in computing sciences and culminating to the end users.

The continuous research and development of a prototype component related to content sharing, devices interaction in a network, be it local or public and the ability to use the many resources available on mobile devices such as embedded sensors, will be the focus of the study who intends to follow this dissertation.

The data collection to be shared will pass from a specific process of classification where some algorithm techniques will be applied for a better learning process over the inferring data.

This investigation is about the development of behavior aware data sharing in pervasive systems, in which nodes tries to get data based on the current behavior of the user, detected by means of sensorial Information provided by available sensing middleware. Data sharing in pervasive environments is envisioned to be based on direct communication between devices following a data centric approach in delay tolerant networks.

With its main architecture represented in figure 1, the prototype baptized by *PersonalSense* is an acronym of *Personalized Sensing System* presented itself as a middleware available on a network for sharing contents based on states of device or human being, taking advantage of the mobiles sensory capabilities.

The main operational blocks of *PersonalSense* are:

- **Sensing Application** for managing sensorial capabilities in interaction. This dissertation will describe how sensor data could be used;
- **Data Tagging** is the block were tags configuration will be made;
- **Networking Application**, responsible to enabling the kind of contents to be shared through a shared system interface.

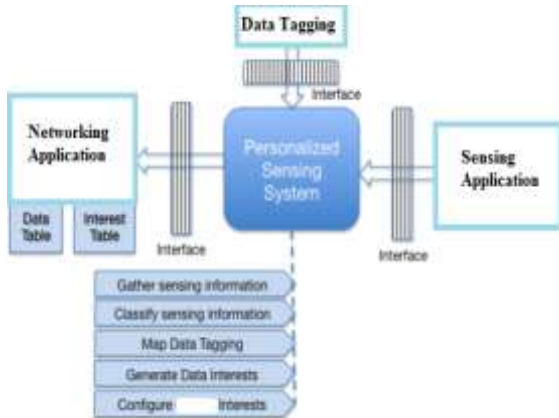


Figure 1: Flowchart of Personalized Sensing System Middleware

PersonalSense continuous investigation will be part of CitySense Project from Laboratory R&D in Informatics Systems and Technologies (SITILabs) with theme: **CitySense, Large-scale Opportunistic Sensing in Urban Scenarios**. Its main building blocks and challenges presented in figure 2 are described below:

- **Sensing Abstraction** - Respect privacy of the owner of the device; support sensing on all sensors;
- **Continuous Sensing** - Real time classifiers; Sensing duty cycles;
- **Cooperative Inference** - Exploit phones spatial distribution and context diversity; Sync for cooperative Inference; Cooperative computing for large inferences;
- **Social structures inferences** - Recognize many of the common structures in the user's routine; Learning of social structures; Learning of rhythms and network dynamics;
- **Subjective Sensing** - Understanding user's interests: correlation of sensing and applications; Service personalization: support applications tailored to individuals;
- **Sensing Visualization** - Affective computing applications, interacting with users through their social and emotional state; Social Networking Sensing, for sharing sensing experiences; Mude Sensing based on passive user interfacing.
- **Data Exchange** - Information-centric architecture; In-networking caching synchronization; Social-aware forwarding.

Within the CitySense project, *PersonalSense* will be integrated with Maestro and ICON based on the architecture illustrated in Fig. 14, where *PersonalSense* will be responsible for the subjective sensing property of the CitySense architecture.

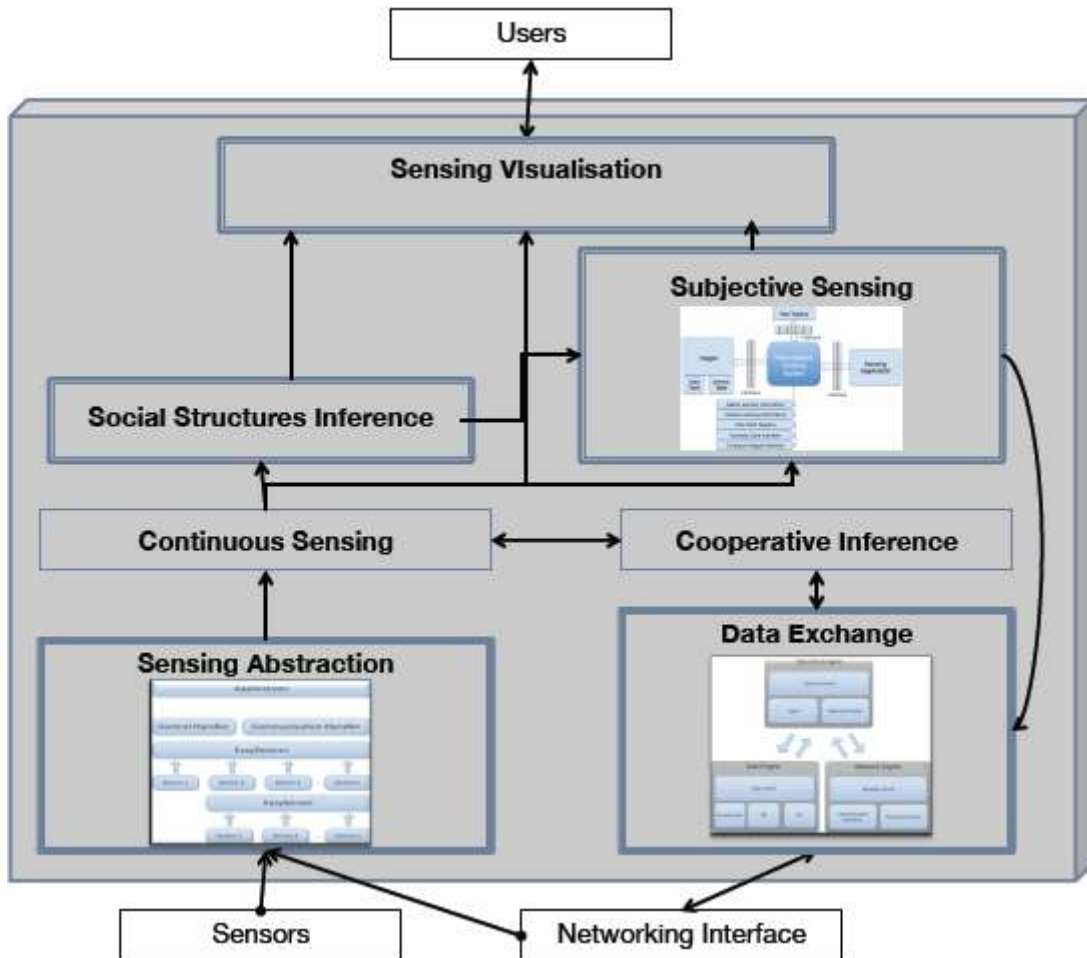


Figure 2: Pervasive Sensing Architecture of CitySense Project

Some aspects are relevant for continuous study of *PersonalSense* as a Project:

- Test all kinds of mobile sensors;
- Testing interaction with Maestro and ICON in a real mobile phone;
- Application of other classifiers algorithms;

1.1. MOTIVATION

All starts with the idea of how to connect someone that lives in a place with Internet and mobile connections with others who lives in distant places like remote villages without those conditions like a scenario represented in figure 2. Some concerns arise like which technologies would be applied, the infrastructure, even the communication protocols for sharing the data between these realities.

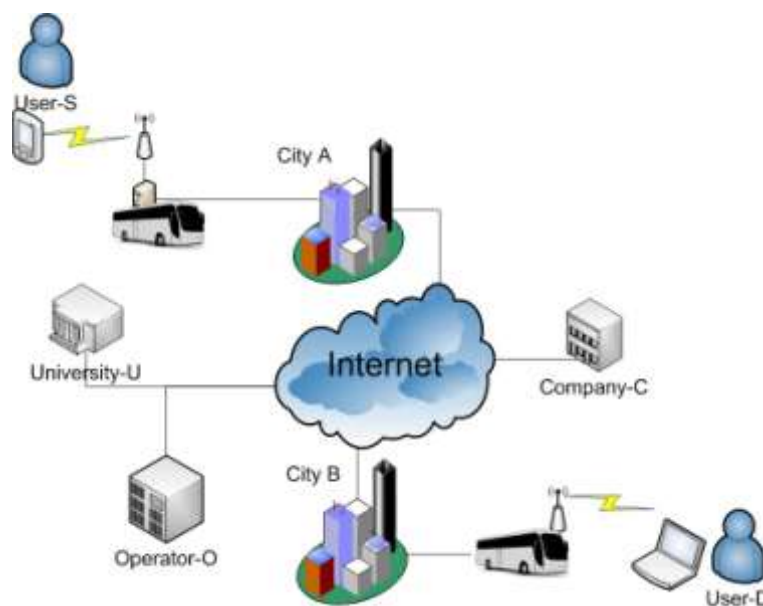


Figure 3: DTN Scenario

Source: (DTN Project, 2013)

In continuous investigations, arose the idea to build a middleware to run in mobile devices that uses the state and users behavior for sending the data only when the opportunity occurs, using mobile devices connection capabilities. Even in DTNs or Internet, the necessity for acquiring information based in mobile user activity through taking such data from some data repository automatically without user action, based only in their behavior or mobile status.

In this dissertation several of these aspects arouse the interest researches and because of its coverage were summarized as follows:

- Analysis of computational models;
- Mapping mobile devices events based on motion sensors, location and voice, with association to users' interests and tagging data;
- Support for wireless connectivity, Internet, as well as intermittent or weak connectivity;
- The development of applications to mobile devices based on sensory behavior;
- Sharing of contents considering the state of the device;
- Data classification;
- Content Oriented Content.

1.2. APPLICABILITY

Started in my¹ research in “*Services Oriented in Opportunistic Networking*” (Pedro, 2012, p1) (Annex 1), focused in the possibility to study and develop architectures to support integrated connectivity's with mobile devices that have embedded one or more kinds of sensors providing users to experiment exchange resources directly across environments with or without infrastructure and in highly dynamic mobile environments.

Considering mobile users' behaviors, (e.g. being sited, running or laying), and taking advantage of the automatic information acquired by *PersonalSense* through mobile devices sensory capacities, this middleware could achieve a wide application fields such as:

- **Internet** - *PersonalSense* would inform an Internet user that is able to receive music, video or news in his Smartphone depending the tags he specifies;

¹ Pedro, M (2012), Lisbon. An abstract from this dissertation author, previously accepted for publication and presentation at African Conference on Software and Applied Computing Symposium, Botswana 2012.

- **Healthy** - a device with *PersonalSense* would inform the network that it is able to receive automatically pertinent information about the healthy state of a patient to his familiars or information of some kind of disease in a specific region;
- **Transportation** – information alerts about the schedule of transports;
- **Weather** – information about the weather condition or environment status;
- **Military Forces** – information about a strange population concentration or some kind of public disorder;

Finally the perception of the user state through configuration of tags concerning in (sitting/not sitting, standing/running or conversation/silence), and consequently pre-configure user mobile profiles facilitating its interaction with others devices that care on the application.

1.3. OBJECTIVES

The main goal this dissertation intends to achieve is to develop a middleware able to analyze device sensory data through a classification process creating a profile of the device to send or receive some kind of content behind a proposed information architecture based on users' behaviors and mobile state.

In a more specific detail, this dissertation proposes to:

- Present a research and use of tools for classifying data as well as platforms able of generating interests based on sensory information forms to support personalized services.
- Evaluate inferring data putting available from mobile devices based in sensors through application classification algorithms.
- Made an approach on some technologies and frameworks that help to build and understand a multiplatform development of middleware for mobile devices;
- Finally, present interfaces with the possibility to access the functionalities of sensors based on tagging data and information from the user.

1.4. STRUCTURE OF THE DISSERTATION

The sequence of this dissertation is composed of five chapters started in chapter one with Introduction highlighting the motivation, applicability and main and specific objectives.

Chapter 2 presents thematic concerning with the Network Concepts. Instead of network for architectures only based in data, *PersonalSense* would be applied also in dynamics networks as Opportunistic Networks where devices and its users are in constant movement reaching different communication environments.

In chapter 3 will be made an approach on learning machine thematic, so important also because of its coverage becoming from data mining and here with a focus on data classification for abstracting users' behaviors presenting some possible mechanisms, algorithms and architectures, used to extract learning.

Chapter 4 is the presentation of middleware *PersonalSense*, its specifications, functionalities and configurations aspects, user's interfaces and development modules.

In the end of this dissertation, chapter 5 presents the conclusions aspects and topics for future work.

2. NETWORKING CONCEPTS

Communication networks have evolved to a more dynamic and mobile environment, where people are more concerned about having access to digital contents and services at anytime and anywhere. However, mobile networks implement connections between machines, making it relevant to analyze aspects such as the ineffectiveness of applications and efficient use of bandwidth.

This investigation starts with exploitation behind the main concepts of Information Centric Networking (ICN), making an approach to Opportunistic Networks to its challenges and their interaction with PersonalSense. Based in studies through different platforms presenting as differentiating factors in this current paradigms of communications, such as:

- NetInf (NetInf, 2014) (Network of Information), an approach to connect different technology and administrative domains into a single information-centric network based on a hybrid name-based routing and name resolution scheme and support many different types of networks and deployments, including Internet access/core network configuration, data centers and infrastructure-less networks;
- Hagggle (Hagggle - A content-centric network architecture for opportunistic communication, 2013), an architecture that allows mobile devices to exchange content based on the direct interests between themselves when they happen to come in close range contact;
- CCNx (CCNx, 2013), an open source project in networking, based on one fundamental architectural change: replacing named hosts with named content as the primary abstraction;
- ICON (Morais, 2013), a middleware for information and context based networking, able to allow content sharing based on interests and users' context.

This chapter finalizes with an overview *PersonalSense* use for enable its contents to another middleware present in a network.

2.1. INFORMATION CENTRIC NETWORKING

Information Centric Networking seeks to adapt the network architecture to the current network usage patterns supporting the fact that the vast majority of current Internet usage consists of data being disseminated from a source to a numbers of users and agents as devices represented in figure 4. ICN has a founding principle that a communication network should allow a user to focus on the data rather than having to reference a specific, physical location where that data is to be retrieved from.

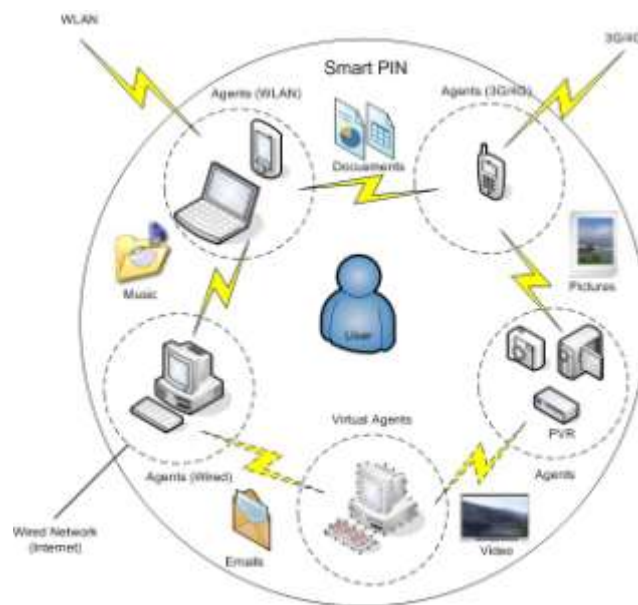


Figure 4: Information Centric Networking

(Smart Pin, 2009)

The wide range of benefits it presents, some are displayed as below:

- Use a data storage cache at each level of the network, decreasing dramatically the transmission traffic, and also increase the speed of response;
- Allows a simpler configuration of network devices;
- Security into the network at the data level;
- The name of content sufficiently describes the information;

Researchers are considering some main aspects as building content routers that support content-centric networking at high speed and using an application layer design for deploying a content-centric interface. With that some benefits such as an easier deployment, backwards compatibility and a more flexible delivery support.

Another approach for data centric networking, with focus on the context of the user is a study presented by ICON, an approach developed at (SITI Labs, 2013)² (Informatics Systems and Technologies Research Laboratory).

2.2. OPPORTUNISTIC NETWORKS

With increasing deployment of wireless networks, opportunistic networks or OppNets also known as *Delay Tolerant Networks* (DTN) are becoming commonplace associated to the high costs and limitations to the access points with Internet provided by ISPs (Internet Service Providers).

Commonly in a standard network, nodes are usually connected statically with the communication path pre-determined through network layer of the Open Systems Interconnection (OSI) Model³.

OppNets appear as islands formed by nodes between devices of different characteristics but in certain opportunistically space of time and in a given geographical environment. Nodes also can be presented in a fixed way being intermittent communications due to variations in the wireless conditions, becoming parts of mobile structures without a pre-determined path of communication. This process occurs dynamically between devices in where each other will act as a route to carry on a message.

With them the natural people mobility can be used to transport stored data relying on Bluetooth technology, even wireless LANs as Wi-Fi/IEEE 802.11.

In the Table 1 are represented some research scenarios in where OppNets could be implemented.

² SITI Labs - Informatics Systems and Technologies Research Laboratory, created in 2009 at Lusofona University of Lisbon.

³ A conceptual model that characterizes and standardizes the internal functions of a communications system applying abstraction layers.

Table 1: Communications Scenarios networks

RESEARCH SCENARIOS OF IMPLEMENTATIONS OF OPPORTUNISTIC NETWORKS	
WIMAX	Modern Mobile Network with alternative radio technologies.
Wireless Local Area Networks	802.11n IEEE communications standard technologies, MIMO (Multiple-Input and Multiple-Output) and hybrid mobile;
Ad-Hoc Mesh Networks	Using different radio technologies, coordinating of radio spectrum, self-organization;
Vehicular Networks	Inter-Vehicular real-time ad-hoc communications and using satellites;
Cognitive Radio Network	Methods to detect disorders, networks with multiple physical radio layers;
Sensor Networks	Efficient protocols, hierarchical topologies, data aggregation and information flows;
Pervasive Systems	Heterogeneous radio technologies, sensor integration with Wireless Local Area Network (WLAN) mobile.

In its evolutions OppNets can integrate some different paradigms of network technologies and communications as shown in table 1, and their wide implementation can be exemplified as follows:

- **Interplanetary Internet**

Interplanetary Internet (IPN) a kind of networks where communications would be with errors and greatly delayed by the great interplanetary distances. They acts as a store and forward network of internets that are more often disconnected using the Bundle Protocol (BP) able to seek errors and disconnections.

- **Ad-Hoc Networks Military**

These systems operate more in hostile scenarios where mobile nodes, environmental factors or intentional interference of electronic devices and systems can cause disconnections. The traffic in these networks may have to compete with other services for priority levels, example of a traffic data that may have to wait several seconds or more while a high priority of voice traffic is loaded on the same line of traffic. Such systems have characteristics with a very high level in its security structure.

- **Vehicular Ad-Hoc Networking**

An example of Vehicular Ad-Hoc Networking (VANET) is the technology applied in cars in movement as nodes in a network to create a mobile network. In this kind of technology every participating car is a wireless router or node with a possibility to create a wide range of network.

- **Mobile Ad-Hoc Networking**

Mobile Ad-Hoc Networking (MANET) presents as a self-configuring network infrastructure of mobile devices connected by wireless. In MANETs, devices dynamically change their links to other devices, also when received some traffic to another destination acting like a router. Their main characteristic is that they can operate themselves without an Internet infrastructure. MANETs can be used for communicating among vehicles and between vehicles and roadside equipment.

- **Sensory Networks**

These networks are often characterized by its extreme limitation from battery, memory and CPU (Central Processing Unit) capacity. When implemented, such networks can achieve higher scales with the possibility of reaching hundreds to thousands of network nodes. Communication within them is usually schematic for battery conservation and sets of nodes are often named (or addressed) only in aggregated form. Interactions with other networks are realized through nodes called *proxy's* with specifics protocols. For this kind of networks with the technologies involved Internet will be needed, for its heterogeneity, elasticity and mobility support.

2.3. SUMMARY

In a widely way could DTNs be considered as a part in developing societies and are poise to play a key part in future space networks. Walking together with DTNs but more focused to the current communications technologies ICN more concerned with how the data putting in the network should be read or understood by others devices is also the mote in this study.

Considering the fact that data must be shared with other devices, it's important to use a middleware that provides the information data to the network. This dissertation choose ICON an ICN middleware developed in the DTN-Amazon Project of SITILabs, which main characteristics are its interface concept of a delivery-centric application allowing others applications to stipulate diverse delivery requirements that place certain constraints on how the content should be provided.

Theses constraints can deal with such things as performance, resilience, security and anonymity. Through such an interface, applications can shape how the underlying delivery is performed without needing to handle such concerns themselves.

Middleware *PersonalSense* can be used at any network even Internet or OppNets under data-centric concept. This dissertation follows a review about the utilization of sensitive devices (eg. Smartphone) in direct connection environments. In disruptive environments, encounters between devices will occur dynamically, and with device communication capabilities the opportunities of achieving information's based on tags will occur automatically.

3. SENSING AND INFERENCE

Inference is the act or process of deriving logical conclusions from premises known or assumed to be true (The Free Dictionary, 2009), and in this same source, sensing can be understood as the learning of some concept.

Becoming from data classification, abstraction process to machine learning, this chapter will be a description of some thematic that help to understand the concepts presented above in aggregation with mechanism available for this kind of research.

3.1. SENSING ABSTRACTION

“Each significant piece of functionality in a program should be implemented in just one place in the source code. Where similar functions are carried out by distinct pieces of code, it is generally beneficial to combine them into one by abstracting out the varying parts.” (Pierce, 2002).

As said by Schmidt *“The phrases of any semantically meaningful syntactic class may be named.”* (Schmidt, 1994). The concept of abstraction presented above can be translated as a path by which process derived from the usage and classification of concrete concepts, principles or methods. With this idea, abstraction could be seen as a way to reduce the information content of a concept or an observable phenomenon, typically to retain only information which is relevant for a particular purpose. About a specific product abstraction retains its particular characteristics on general attributes and behaviors.

In computer language abstraction is used to separate categories and concepts from instances of implementation details. The idea is to reduce details so that the programmer can focus on a few concepts at a time presenting the system in abstraction layers. The use of abstraction also avoids code duplication in a program.

An example of abstraction implementation is illustrated in a middleware project developed within CitySense Project of SITILabs, Maestro, which was a part of sensing abstraction, which main properties are the support sensing on all sensors and respect privacy of the owner of the device. Maestro is an embedded type middleware, which runs locally on

the device and has layers of abstraction, recalling, sensors, storage and communications and is intended to work for Ubiquitous Sensing Platforms (USP).

Maestro was developed in C# managed language differs for the other middleware's in sensing thematic because it could be distributed to all devices that can run the .NET Framework on their Operating System's, these devices include Windows phones, android phones, iPhones, Android Tablets, Windows workstations and Linux workstations (Barbosa, 2012, P. 45).

Maestro has the ability to create device profiles, letting users to select which types of sensors and network interfaces to expose to the network and also sensor virtualization that enables the borrowing and incorporation of external sensors onto a physical device, as well as a highly dynamic and loosely coupled architecture, built on top of Dependency Injection principles that make the solution very extensible and highly scalable.

The PersonalSense interest in Maestro is precisely to associate in its Sensing Module the capability to read the sensing data provided by it. For that, will be necessary for both applications to interact through an interface of sensing. *PersonalSense* will need to travel to the Maestro directory systems files to get the XML (eXtensible Markup Language) files for make the classification. This aspect will be shown in session 5.4 about inference module. The format of the file and the path of directory file must be previously configured.

3.2. INFERENCE

Defined by (American Heritage Dictionary, 2009), Inference is “*act or process of deriving logical conclusions from premises known or assumed to be true*”. Known in the Artificial Intelligence as Inference Engine a tool composed with *expert systems* consisted also of a user interface and a knowledge base, that store facts about some environment applying logical rules to the knowledge base and deduce new knowledge.

As illustrated by the picture in figure 5, in an expert system there is an interaction conducted by the user interface between user and the system. The user provides information about the problem to be solved and the system then attempts to provide insights inferred from the knowledge base. (Griffin & Lewis).

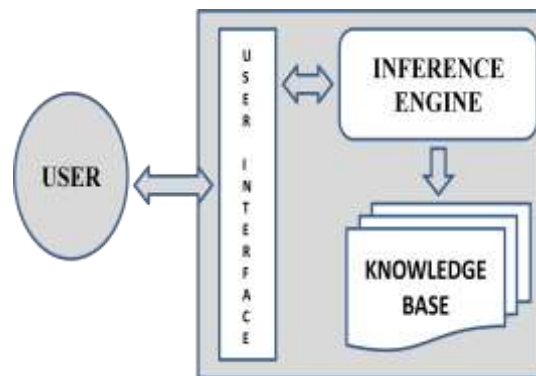


Figure 5: Interface System

Knowledge base consists of some encoding of the domain of expertise for the system, and can be represented in form of semantic net, procedural representation, procedural rules or frames. This work will consider only procedural rules for knowledge base. These rules occur in sequences and expressed in form of *conditions* and *actions*,

if <conditions>

then <actions>

where if the *conditions* are true the *actions* are executed. When rules are examined by the inference engine, actions are executed if the information supplied by the user satisfies the conditions in the rules.

Inference process would iterate as each new fact in the knowledge base could trigger additional rules in the inference engine. Inference engines work primarily in one of two modes: forward chaining and backward chaining. Forward chaining starts with the known facts and asserts new facts from satisfied conditions in rules. Backward chaining takes actions and queries the user about information which may satisfy the conditions contained in the rules. It is a verification process rather than an exploration process. This dissertation will be presented an inference engine which operates by the method of forward chaining.

As referenced by Gama “*The problem of machine learning can be formulated as a search problem in a space of possible solutions.*” (Gama, 2012). Machine learning studies how to automatically learn to make accurate predictions based on observations and, classification task can be understand as a supervised learning function that maps a set of input

data into a finite number of categories, obtaining accurate prediction of the data, where in it, each example belongs to a class among a defined set of classes. The training data presented in the beginning of the learning process (fig. 6) consist of a set of training examples, and the examples consist of a set of attributes and an attribute in a discrete data.

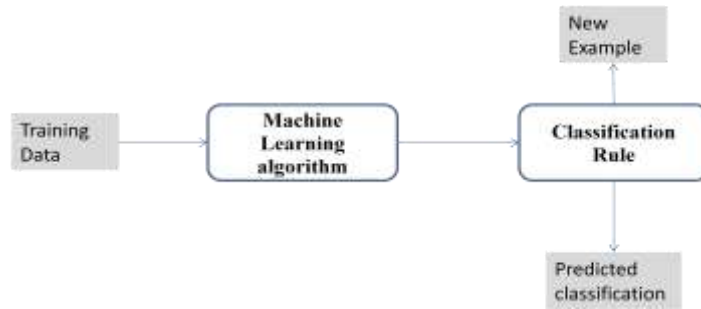


Figure 6: Machine learning algorithm process

The purpose of a classification algorithm is to find relationship between attributes and a class so that the classification process can use this relationship to predict the class of a specific new example or unknown example as shown in table 2.

Table 2: Classification table with attributes

CLASSIFICATION PROCESS WITH ATTRIBUTES		
Sensor (Predictive Attribute)	Values (Predictive Attribute)	Result (Class attribute)
Accelerometer	0,05474096	True
Accelerometer	-0,7390029	False
Accelerometer	-0,6373411	False

With this, means that the classification consists in obtaining a model based on a set of examples that describe a function non-known. This model is then used to predict the value of the attribute-target of new examples.

Data analysis can be performed through the process of knowledge discovery in databases, which gathers several steps as tasks, methods and algorithms. The classification models have two types of attributes: predictive and objectives. Generally an attribute objective corresponds to a categorical variable that represents classes previously defined. The predictive attribute are used to infer from what class a new object belongs.

There are different techniques with features for classification, segmentation, association, visualization and forecasting data, such as Neural Networks, Rule Induction, Genetic Algorithms, Bayesian Methods, Decision Trees, and Nearest - Neighborhood. Described in the table below In machine learning algorithms can be organized into a taxonomy based on the desired outcome of the algorithm or the type of input available during training the machine.

Tabela 3: Machine learning algorithms

MACHINE LEARNING ALGORITHMS		
Supervised learning	Analyzes the training data and produces an inferred function, which can be used for mapping new examples.	<ul style="list-style-type: none"> • Support Vector Machines • Neural Network • Naive Bayes • Decision trees • Linear regression • Logistic regression • Linear discriminant analysis • K-nearest neighbor algorithm
Unsupervised learning (Ghahramani, 2004)	Algorithms operate on unlabelled examples. The objective is to discover structure in data not to generalize a mapping from inputs to outputs.	<ul style="list-style-type: none"> • Artificial neural network • Data clustering • Apriori algorithm • K-means algorithm
Semi-supervised learning (Zhu, 2008)	A supervised learning generalization; combines both labeled and unlabelled examples to generate an appropriate function or classifier. The desired output values are provided only for a subset of the training data. The remaining data is unlabelled.	
Reinforcement learning (Kaelbling, Michael, & Moore, 1996)	Reinforcement learning is the problem faced by an agent that learns behavior through trial-and-error interactions with a dynamic environment. Concerned with how intelligent agents ought to act in an environment to maximize some notion of reward. The agent executes actions which cause the observable state of the environment to change. Through a sequence of actions, the agent attempts to gather knowledge about how the environment responds to its actions, and attempts to synthesize a sequence of actions that maximizes a cumulative reward.	<ul style="list-style-type: none"> • Temporal difference learning • Q-learning • Learning automata • Monte Carlo method • SARSA

After the process of description the relations between objects, a machine learning algorithm learns from a set of training data, returning one result in a set of available results. Each kind of known algorithm uses its own way to represent the result to be returned. Decision trees use a tree structure in which internal node is represented by a question about the value of a specific attribute and each external node is associated with a class.

The preference to represent the set of data classified for the algorithm depends from the chosen technique as illustrated in figure 7.

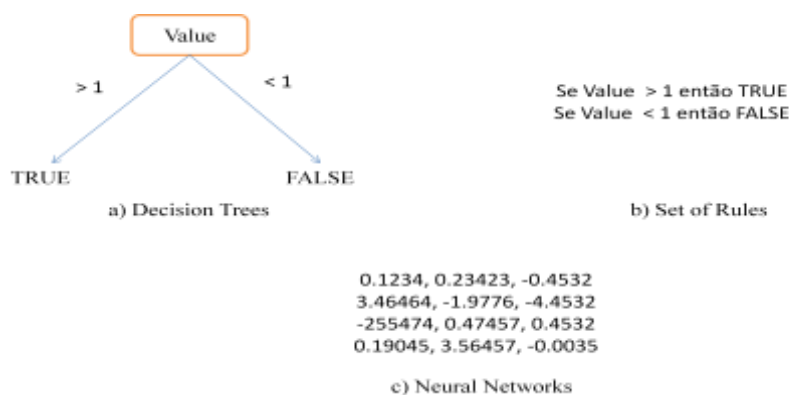


Figure 7: Data classification technique examples

Described in section 3.2.3, ID3 algorithm chosen in the dissertation used in the induction of decision trees returning a little node in it search preference. Decision trees in learning process represent a predictive model which maps observations about an item to conclusions about the item's target value. Classification trees or regression trees represent themselves the descriptive names for such tree models.

Although decision tree have some limitations as the concepts difficult to express by them or when its learners create over-complex trees that do not generalize well from the training data. Some concerns become available in the chose the study of Decision Trees in this dissertation such as:

- Its visually and explicitly form to represent decisions and decisions making;
- The simple form to interact and understand;

- Requires little data preparation, without the necessity to require data normalization as occur with other techniques;
- Besides of other techniques that are specialized in analyzing datasets that have only one type of data, decision trees are also able to handle with numerical (Neural Networks) and categorical data;
- Their easily form to explain by Boolean logic, observations in analysis;
- Using statistical tests, it is possible to validate that makes it possible to account for the reliability to understand;
- Its robustness because of its capability to performs well even if its assumptions are somewhat violated by the true model from which the data were generated.
- It performs well also with large amount of data.

3.2.1. REQUIREMENTS

Although the variety of existing tools does not match efficiently with this study, as the discovery of patterns and relationships in the data, however, was developed a module corresponding to the classification task from data provided by the sensing middleware Maestro whose grading method employed was decision trees, applying the algorithm ID3 (Quinlan, 1979) to its induction.

In order of that, classification task to be performed requires a particular method that must be adopted according to the knowledge to extract from the data file. In development of the classification sensing module, becomes important the choice of training algorithm that fits the desired classification according the performance, tasks and implementations of sensory data.

As illustrated before, PersonalSense choice goes to *Decision Trees* method with *ID3 algorithm* due the fact that they present themselves as a suitable technique for finding and describing structural patterns in data that aid in its perceptions as well as make future predictions and essentially obtain gain knowledge.

The decision trees method consists of ways to represent knowledge in the form of tree. This is a sorting method which assists in the classification and prediction of unknown samples based on records of a training set.

This method allows the user of a particular application to define the output object. As such, from a group of data it is possible to identify the important factor correlated to this object. However, a tree after being mounted can be classified unknown samples without necessarily being tested values of its attributes. It is only necessary to know the classes of each record in the training set, and the definition of elements of the tree mounted by the algorithm. As illustrated in figure 8, a decision tree is represented by a set of nodes connected by a branch subdivided as follows:

- Root node - the tree starts;
- Common node - share certain attribute and generate ramifications, representing conjunctions of features that lead to the class labels leaves represent;
- Leaf nodes - represent the class labels, contains information about the classification algorithm.

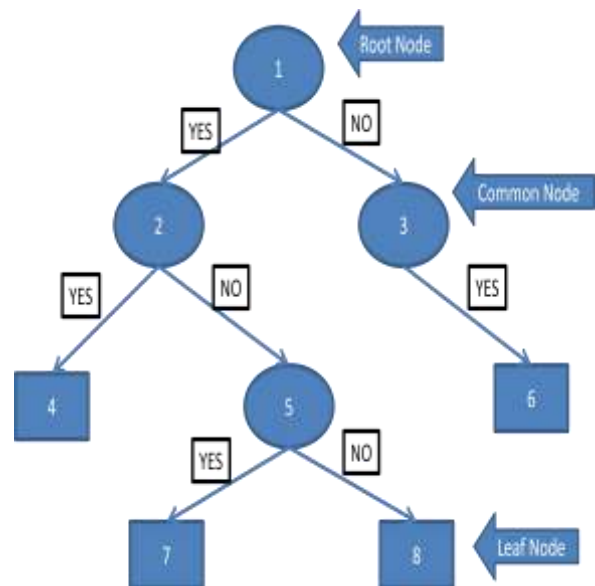


Figure 8: Decision Tree example

The idea of the algorithm is to build a tree where each node indicates the test of an attribute. The attributes chosen are called splitters. The choice is made from the highest information gain of the algorithm which is the information quality attribute. The attribute classifies the set of instance selected from some selected criteria's such as information gain or ratio of the gain. Examples of input and output or collected data could be used. The ramifications have all possible values of the attribute specified in the node to facilitate understanding and interpretation.

As described in inference at section 5.3, in this dissertation the requirements to inference process are related in a sequence and with specific main classes. In *PersonalSense* a sub-class *implementacaoArvoreDecisao.cs* has a function that prepare the construction of the decision tree (fig. 9) before read the data provided by a previous selected sensor file.

```
public class ImplementaçãoArvoreDecisao
{
    string _fonteFicheiro;

    public string GetTree(string fonteFicheiro)
    {
        _fonteFicheiro = fonteFicheiro;
        RawDataSource samples = new RawDataSource(_fonteFicheiro);

        ArvoreAtributoColecao atributos = samples.GetValidAttributeCollection();

        arvoresDecisao id3 = new arvoresDecisao();
        ArvoreNode root = id3.mountTree(samples, "resultado", atributos);

        return PrintNode(root, "");
    }

    public string PrintNode(ArvoreNode root, string tabs)
    {
        string returnString = String.Empty;
        string prefix = "Melhor atributo: ";

        if (tabs != String.Empty)
            prefix = " -> Resultado Provável: ";
        returnString += (tabs + prefix + root.Atributo) +
            Environment.NewLine;
    }
}
```

Figure 9: implementation code for decision tree

After some integrated classes such as *RawDataSource.cs* with the function to return the selected file, the *ArvoreAtributoColecao.cs* responsible to manage the attributes obtaining the collection of valid attributes, also the implementation of the own ID3 algorithm and finally the class to generate de tree *ArvoreNode.cs*.

In the same *implementaçãoArvoreDecisao.cs* class, *PrintNode* method is responsible to return a string with a tree node (fig. 10), by considering aspects as the attributes to be filled in the tree by analyzing probable values and best attributes culminating with the probable result.

```
Melhor atributo: values
Entrada: 0.05474096
-> Resultado Provável: True

Entrada: -0.6373411
-> Resultado Provável: False

Entrada: -0.7390029
-> Resultado Provável: False
```

Figure 10: Tree node of sensor file

This inferred process with apply of decision trees return the idea of the probable result, so PersonalSense defines for Accelerometer analysis inference process that when the result becomes (True, True, False) it must infer as demonstrated in table 6, that the *User State* could be *Walking* or *Stopped* the *Event* process *Movement* and inform also that the *Picture* or *Music* could be the kind of that that previous device can receive. Important also is the *Common Interest* of the user, in this specific case will be *Technological* contents.

3.2.2. ALGORITHMS

For classification process in decision trees as described in section 3.2.1 for supervised learning, some algorithms where be considerate in this work. The table 4 illustrates the analyzed algorithms with its own advantages and limitations. Parameters for the chosen process were:

- Available programming language code, with preference in C#;
- Algorithm that generate a decision tree model;
- Classifies a known set of training data;
- Classifies any kind of training data (numeric or nominal);
- Generate only one best attribute in evaluation;

In this dissertation ID3 and C4.5 were the algorithms analyzed with their implementation of C# code. The algorithm ID3, identified by Iterative Dichotomizer, developed by J. Ross Quinlan⁴ is an algorithm that uses logic and mathematics to process, organize and simplify a large amount of data. It also has the ability to operate non-numeric data, which is one, difference between ID3 and statically methods, because while the ID3 assumes nominal attributes, statistical methods uses numerical attributes.

Algorithm C4.5. is an improved method of ID3, with the main proposes of addressing the shortcomings of the ID3. Overcome the obstacle on the adjustment, through the use of tree pruning and overcome real-world common practical problem, example of attributes with numeric values, missing values and noisy data, by increasing the efficiency

⁴ Is a computer science researcher in data mining and decision theory. He has contributed extensively to the development of decision tree algorithms, including inventing the canonical C4.5 and ID3 algorithms.

level in processing time and memory used, providing the possibility to perform a cross-validation, thereby increasing the quality to estimate errors by the classifier.

This algorithm gives preference to less complex hypotheses that are consistent with the observed reality and has the ability to generate decision rules and compare them independently of the trees constructed.

PersonalSense does not choose C4.5 even the new ones because of their capacity to works with unavailable values. PersonalSense makes the inference by classifying available values analyzing them by specific attributes obtaining some kind of learning.

In PersonalSense middleware, the classification module was developed by applying the algorithm for the induction of ID3 decision trees, which allows the user to interact better regarding to the classification task by means of decision trees, giving them an alternative to aggregate knowledge regarding to the process of data mining.

The number of records to sort may vary from the storage parameters specified by the user. For *PersonalSense* current version, just one parameter will be chosen.

3.2.3. FUNCTIONAL COMPARISON

Based on decision tree method and the chosen algorithm for inferring the set of training data, this block will be concerned about the comparison process between ID3 and C4.5 tested algorithms in this dissertation. Some steps must be put in consideration for this analyzing process:

- A classifier is constructed to summarize a set of predetermined classes, by learning from a set of training data;
- The classifier is used to determine the classes of newly arrived data, as determined by one of the attributes;
- The step known as supervised learning will occur since the class label of each sample is provided.
- The learning model will be represented in form of decision trees, classification rules, or mathematical formula.

The challenge lies in constructing the decision tree is how to find the best attribute to split the sample data. The following describes two common criteria.

- Information Gain
 - Concerning to the simplest criterion, uses the entropy measure, calculated as

$$Entropy(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

where S is the dataset, c is the number of classes and P_i is the proportion of each class. The information gain is then calculated as

$$Gain(S, A) = Entropy(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where $V(A)$ is the set of all possible values for attribute A , and S_v is the subset of S for which attribute A has value v .

- Gain Ratio
 - There exists a natural preference in information gain, as it favors attributes with many values. For example in PersonalSense the attribute *value* may have the highest information gain, but if it lead to a very broad decision tree of depth one and is inapplicable to any future data.
 - Gain ratio is an advanced criteria which penalizes attributes by incorporating split information.

$$Split\ Information(S, A) = - \sum_{i=1}^c \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

```

/// <summary>
///   Static class for common information measures.
/// </summary>
///
public static class Measures
{
    /// <summary>
    ///   Computes the split information measure.
    /// </summary>
    ///
    /// <param name="samples">The total number of samples.</param>
    /// <param name="partitions">The partitioning.</param>
    /// <returns>The split information for the given partitions.</returns>
    ///
    public static double SplitInformation(int samples, int[][] partitions)
    {
        double info = 0;

        for (int i = 0; i < partitions.Length; i++)
        {
            double p = (double)partitions[i].Length / samples;
            if (p != 0) info -= p * Math.Log(p, 2);
        }

        return info;
    }
}

```

Figure 11: Code to illustrate the information gain

This information is sensitive to how broadly an uniformly the attribute splits the data. The gain ratio is calculated as

$$\text{Gain Ratio}(S, A) = \frac{\text{Information Gain}(S, A)}{\text{Split Information}(S, A)}$$

The information gain is calculated for all attributes and the method will select the best one. This ratio will not be defined when the split information is zero, and the ratio may tend to favor attributes for which the split information is very small.

Acting recursively constructing a tree in a top-down structure, ID3 algorithm uses information gain as the measure to determine the best attribute, and then creates a node for each possible attribute value, and partitions the training data into descendents nodes. There are three conditions to stop the recursion:

- All samples at a given node belongs to the same class;
- No attribute remains for further partitioning;
- There is no sample at the node.

The second algorithm in analyze in this dissertation is C4.5, an extension of ID3 algorithm performs similarly to ID3 except using gain ratio to determine the best attribute. It also makes some improvements to ID3, including that it can handle numerical attributes by creating a threshold and splitting the data into those whose attribute value is above the threshold and those that are less than or equal to it. C4.5 can also prune the decision tree after creation, which reduces the size of the tree.

Analyzing both algorithms, some characteristics could be returned such as:

- ID3
 - Ability to operate non-numeric data;
 - Work only with available values;
 - Generate one classifiers form the same set of training;
 - The attribute with the highest information gain is the one that best classifies the set of training samples.
- C4.5
 - Apply the tree pruning with the ability to work with missing values and noisy data, increasing the efficiency level in processing time and memory used;
 - Increasing the quality to estimate errors by the classifier;

The ID3 algorithm uses measures of information gain to reduce the uncertainty about the value of the output object. The gain information consists of a statistical measure used to construct decision trees in order to choose the test attribute between all concerned with the node in question.

```

///<summary>
///Retorna o melhor atributo
///</summary>
///<param name="atributos"> Um vetor com os atributos</param>
///<returns>Retorna o que tiver maior ganho</returns>
///
private ArvoreAtributo getBestAttribute(DataTable samples, ArvoreAtributoColecao atributos)
{
    double maxGanho = 0.0;
    ArvoreAtributo resultado = null;

    foreach (ArvoreAtributo atributo in atributos)
    {
        double aux = ganho(samples, atributo);
        if (aux > maxGanho)
        {
            maxGanho = aux;
            resultado = atributo;
        }
    }
    return resultado;
}

```

Figure 12: method code to infer the best attribute

The attribute that has the highest information gain is the one that best classifies the set of training samples. Therefore, the use of the concept of information gain allows minimizing the final depth of the decision tree.

3.2.4. ALGORITHM IMPLEMENTATION

The code of chosen algorithm ID3 belong from a framework called *Accord.NET* obtained in <http://accord-framework.net/>. Presenting itself as an open source framework which its development kit could be imported to a development environment like Visual Studio .Net, provides itself specific libraries packages in C#.Net Programming Language able to implement such as static analysis, processing function even for machine learning.

The package *Accord.MachineLearning* provides a folder (fig. 13) with Decision Trees classes (*DecisonNode.cs*; *DecisionTree.cs*; *DecisionVariable.cs*; *TreeExpression.cs*), with also a *Learning* package with the code of algorithms C4.5, ID3 and, measure of information gain classes (*C45Learning.cs*; *ID3Learning.cs*; *Measures.cs*).

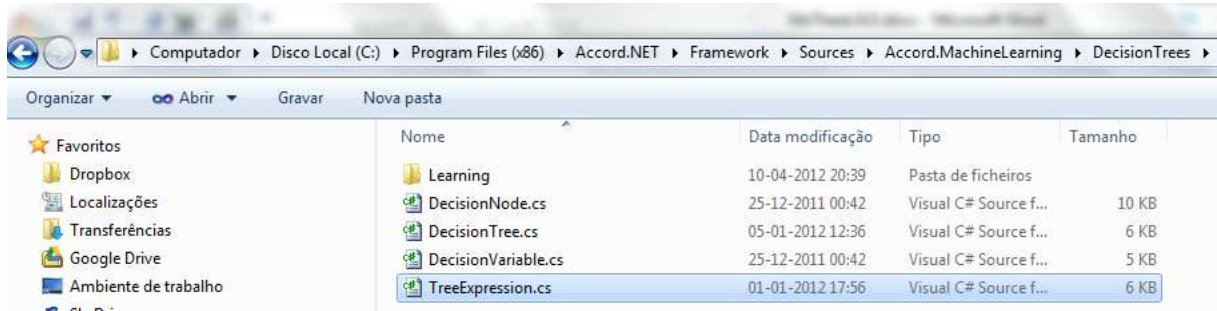


Figure 13: Accord.NET Machine learning package

For this dissertation some adaptation were made in the code of the classes to implement the idea of classification process through the inference in training data of sensors values. Bellow will be presented the main classes provided by Accord.Net Learning process and readapted in PersonalSense project:

- *arvoreDecisao.cs*: main class that represents a decision tree, with methods such as solving input parameters in a tree classification. In this class some characteristics are in analyze such as the return with the result of the counting of total positive samples in the data source; return of entropy value with the application of a mathematic formula that analyze the proportion of positive and negative values; a method to verify attributes with a positive or negative values; make the best attribute from information gain method; mount the tree take in consideration the samples in consideration.
- *ArvoreNode.cs*: may or may not contain child nodes contained in a collection of nodes represented by the class *ArvoreNodeColecao.cs*. When a node is not a leaf, it contains the specified class with a collection of child nodes. The branch specifies an attribute index, indicating which column from the data set (the attribute) should be compared against its children values. The type of the comparison is specified by each of the children. When a node is a leaf, it will contain the output value which should be decided for when the node is reached.
- *ArvoreAtributo.cs*: specifies the nature of each variable processed by the tree, such as, if the variable is continuous, discrete, it expectations or valid limits.

- *ArvoreNodeColecao.cs*: is the class that contains the collection of decision nodes, specifying the index of an attribute whose current value should be compared against its children nodes and which type of comparison is specified in each child node.

3.3. SUMMARY

The classification module of sensing arises in order to apply the concept of data mining and knowledge available regarding their tasks and results of the applied algorithm. The mining of classification models in databases is a process consisting of two phases: learning and testing. In the learning phase a classificatory algorithm is applied on a set of training data selected at random, resulting in obtaining the constructor of the classifier itself, with the application of several techniques like decision tree methods.

PersonalSense follows this focus considering the values of sensors from mobile devices as attributes giving them relevance in the study. The selection of ID3 algorithm instead of others is because of its relevance aspect in analyzing data even nominal or numeric, and its stability with the development platform and programming language used in the project. PersonalSense choose to use the free ID3 algorithm code from Accord.NET extension for its implementation in the learning mechanism.

The architectures platforms presented in this section were the product of this investigation process to show the evolution process of classification data in research Labs and Institutions with their more approach with program languages platforms integrated to the nowadays technologies.

4. PERSONALSENSE FRAMEWORK

This dissertation intends to present a middleware able to analyze sensory data received from previously configured mobile devices behind a proposed information architecture based on users’ behaviors and mobile state. As illustrated previously in figure 1, PersonalSense main building blocks are sensing, tagging and communication interfaces with the function to make interaction with other mobile applications.

The *PersonalSense* product is an inference engine presented as a prototype consisting of a set of modules for classifying process taking in consideration learning algorithms in which were included the inference engine implemented in C# programming language also a menu driven interface.

As the representation of classes diagram represented in figure 14 *PersonalSense* was created based in software engineering which main propose was to introduce a connection between sensing process, data tagging and consequently the generation of interests to an information centric network. This dissertation gets itself a focus by the sensing opportunistic process and a connection with data classification build automatically interests in an ICN.

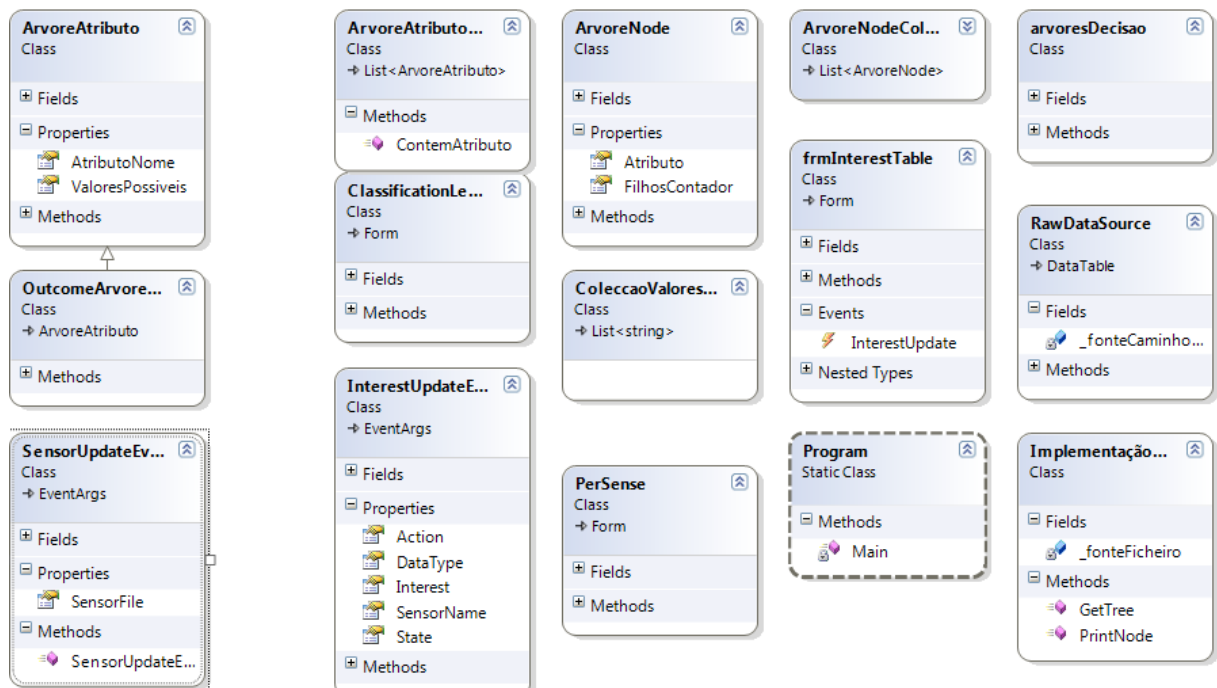


Figure 14: PersonalSense Diagram classes

The knowledge representation language is a high-level block structured in C# which allows the middleware to construct a knowledge base. It is based upon production rules of the form of

if <conditions> then <action list>

where *conditions* are expressions involving *attributes* and logical connective. As variables in programming languages, attributes have types which must be numerical {-0, 0.9345234, 0.23452344} or string {true, false}.

The core of PersonalSense process is represented in the diagram of image below which states description are represented in table 4 by the main classes of PersonalSense Project.

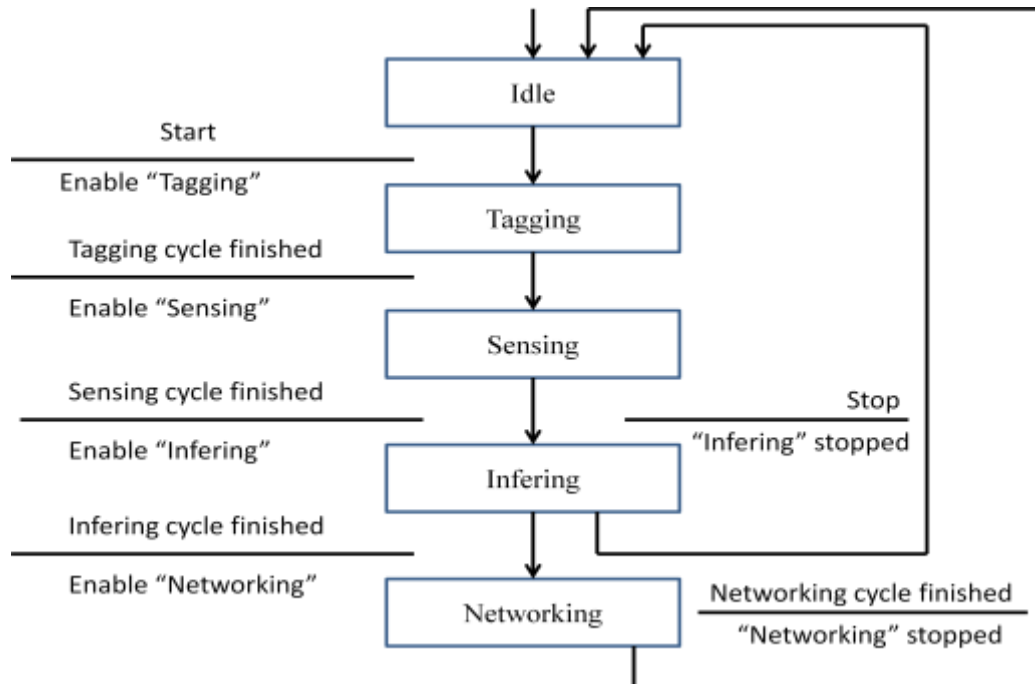


Figure 15: PersonalSense state diagram

Tabela 4: PersonalSense description classes

PERSONALSENSE MODULES		
State	Class	Description
Idle	PersonalSense.cs	Main project class
Tagging	frmInterestTable.cs InterestUpdateEventArgs sensorUpdateEventArgs.cs	Interest Interface Interest file created
Sensing	Classificationlearning.cs RawDataSource.cs arvoreDecisao.cs	Source data reading Data Table creation Implementation of Decision Tree Application of ID3 algorithm Entropy calculation Gain attribute calculation Attributes obtained (class and predictive) Inferring Data (learning process)
Inferring	rawDataSource.cs	Data file interpretation Learning process obtained
Networking	PersonalSense.cs	Info sent to the network

PersonalSense must have the ability to view and interact with other application previously configured for that. Applications behind SITILabs like Maestro and ICON Systems can be seen as partners being installed in the same device for receiving and or sending data. For that propose, PersonalSense interfaces are:

- *IDLE* → Application enabled by the user but without activity.
- *TAGGING* → Tagging process starts when the application gives a chance to the user to make a configuration of device profile.

- *SENSING* → Enabled to read device sensors and make data classification process.
- *INFERRING* → Learning process from data inferred.
- *NETWORKING* → Device state sent to the network.

Table 5: PersonalSense Main Features

PersonalSense Main Features			
PersonalSense Interfaces	Functionality using the interface	External used Applications	Observations
Sensing	Decision Tree Classification Learning	Maestroo	Emulator file used in this study
Tagging	Generations of interests	PersonalSense	Automatically by classifier or Manually by user
Networking	Sharing process	ICON	In this study PersonalSense just enable state device to network

The functional aspects of the application are facing to the reading, analysis and enable the data to be used for others applications presented as input and output data. The flowchart below considers all the process from reading sensing data file through its shared process.

4.1. TAGGING MODULE

It is a classification process for inferring in which users are able to choose manually the elements for inference. A list of items where the user is required to select a pre-entered tagging values are representative in this process. The *PersonalSense.cs* class for user interaction and *InterestTable.cs* that automatically create the device profile are the main classes in this process.

When tagging process occurs automatically users' will not interact with device. Tagging classes are also responsible to manage the elements already in the user's interface as described in table 7.

Table 6: Tags for inference process

TAGGING ELEMENTS FOR INFERENCE				
State	Event	Data Type	Interest	Sensor
Walking	Movement	Music	Music	Accelerometer
Stopped	Movement	Picture	Technology	Accelerometer
Inside	Location	Picture	Nature	GPS
Outside	Location	Video	Movies	GPS
Talking	Conversation	Text	News	Microphone
Non-talking	Conversation	Text	Books	Microphone

This module finalizes with a call to sensing process if the user has the necessity to read sensors to create a device profile. If not, the device is prepared to share its state to the network. At this moment *sensorUpdateEventArgs.cs* class in which is the main variables used to capture the interests to be shared in networking will be active.

4.2. SENSING MODULE

Sensing module is responsible for internal processing activity. Out of user's perception, will be in this step that will occur the data classification process and be obtained the behavior learning.

PersonalSense had tested in two ways. In the first approach, PersonalSense read the own sensor of device obtaining the inferred data internally managed by classifier.

The second represented in figure 10 was the test simulating a file sent by another partner application like Maestro. On it structure highlighted in red are represented the type of sensor and the chosen attribute. Based on them, the class result will be generated internally by the application of the algorithm that will generate the decision tree.

```
<?xml version="1.0" encoding="utf-8"?>
<Dump>
  <SensorReading Parameters="3"
    Timestamp="03-07-2012 19:42:19 +01:00"
    RawDataType="float"
    DeviceId="fd4ad302-ee5e-47a4-a844-2c3193b25423"
    SensorType="AccelerometerGeneric"
    SensorId="83786a6b-600a-4b2d-afd2-ee9bc0f03fa6">
    <Values>
      <Value>0,05474096</Value>
      <Value>-0,7390029</Value>
      <Value>-0,6373411</Value>
    </Values>
  </SensorReading>
</Dump>
```

Figure 16: Format of Maestro file

Internally this process can be described as follow:

→ After tagging process finished, in the source the *dump.Load* method will be used to load it to an XmlDocument.

→ Sensing cycle starts with *ClassificationLearning.cs* class that main functions are:

- Testing the file format;
- Verifying if the file is empty file;

→ An event occurs with the implementation of the tree by the call of *ImplementacaoArvoreDecisao.cs* class declared inside of *arvoreDecisao.cs* class. Its main function will be to obtain the tree and print the result of readings.

For this process, several classes are called with their specific functionalities:

- *RawDataSource.cs*

This class creates a data table responsible to achieve the info of the file to be verified by ID3 Algorithm process.

- *ArvoreAtributoColecao.cs*

Verify the attribute condition by a Boolean statement creating a collection list.

- *arvoreDecisao.cs*

This class implements a decision tree by implementation of ID3 algorithm. With the Entropy Calculation, this class calculates the entropy between the proportions on positive values through negatives. This cross to the data table will return the chosen values only. In the end, a decision tree will be built based in the chosen values.

- *ArvoreNode.cs*

This class creates a sequence of nodes in the tree by verification of the quantity of attributes.

- *ArvoreNodeColecao.cs*

Connected with *ArvoreNode.cs*, to build a list of nodes, this class will be responsible by the print of the *Best Attribute* in analyze and the *Result Most Probable*.

- *ArvoreAtributo.cs*
 - *ColecaoValoresPossiveis.cs*

Connected with *ArvoreAtributo.cs*, to build a list of most probable values, this class returns a collection of the most probable values by name and value using *get* function.

The process proceeds to inferring with the creation of node list using the function *GetElementsByTagName*.

4.3. INFERENCE MODULE

In order to execute a rule-based expert system using the method of *forward chaining* an action is executed whenever they appear on the action list of a rule whose conditions are satisfied. This process involves assigning values to attributes, evaluating conditions, and checking to see if all of the conditions in a rule are satisfied. A general algorithm for this might be:

While values for attributes remain to be input
Read value and assign to attribute
Evaluate conditions
Fire rules whose conditions are satisfied

In this approach, *PersonalSense* method fires first the rule which the system design defined first. Conditions are only evaluated at the time they might change and that rules are checked only when they might be ready to be fired, not before. This occurs as attributes are assigned values and shall only consider rules and conditions affected by the new attribute assignment.

With that, basic components of the inference engine for a rule-based system are:

Attributes: X, Y, Z

Conditions: C₁, C₂

Rules: R₁, R₂

Actions: True, False

An action will be executed when a rule containing it is fired. And the rule will be fired only when all of its conditions are satisfied. A *counter* to each rule was assign to detect it and use it to keep track of exactly how many of the conditions in the rule are currently satisfied. This check is used to see if a rule is ready to fire when all conditions have become true. In turn, a condition need be evaluated only when all of its attributes have been defined and one has changed. This is kept track of with a counter assigned to that condition. In addition, an attribute is flagged as *defined* or *undefined*.

Going the other way, we can determine which conditions need be checked and maybe evaluated with the aid of a *condition list* assign to each attribute. Then, the rules which need checking and possibly firing appear on a *rule list* allocated to each condition. And, each rule possesses an *action list* which enumerates the actions to be executed when the rule is fired.

Then the various lists are set up and the rules and the relationships between the attributes, conditions, rules, and actions may be presented as the graph in figure 8. The graph is also in some sense an illustration of the inference engine for a system containing the above two rules since the engine operates by doing a depth-first search of the graph, beginning at the attribute being changed and continuing down the graph whenever the counter assign to a condition or a rule indicates that all of the information required is present.

For assigning a value to an attribute and performing all other appropriate tasks that this assignment triggers some aspects must be put in consideration:

- All conditions which might change are checked;
- All rules which might fire are examined;
- Nothing is examined *unless there is a need to examine it*.

Inference process occurs after classification process. If the automatic option was chosen, classification process will return the data interpretation acquired from some specific data file or by reading device sensor. After this step, knowledge will be inferred and sharing process enabled to be done.

In this learning process as already represented in table 2 could be observed the class attribute presented by “Resultado” and predictive attributes by “Values” and “accelerometer”. In this inference process with the return of decision process object could be inferred for the accelerometer sensor attribute that most influenced the construction of the model of knowledge gained (*values*). The learning process infers based on the class attribute “Resultado” that the device is in some position according to its axis position. So *PersonalSense* inference assumes that the content to be shared in this situation for this specific sensor could be a “*picture file*”, taking in consideration table 7.

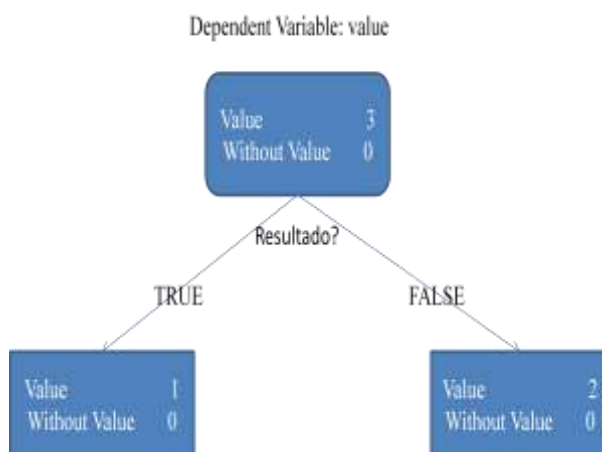


Figure 17: Decision tree

4.4. NETWORKING MODULE

Assuming that the inferring process is done *PersonalSense.cs* main class will be the structure responsible to send the inferred values to the network or share device state with another partner middleware. This work does not intended to exploitation the algorithm for interaction with ICON or other ICN application but only focused on the sensing classification process. As related in the (Section 7 – Future Work), will be *PersonalSense* intention to make a scalable process to mix its process with other applications.

4.5. GRAPHICAL INTERFACE

4.5.1. CONFIGURATION INTERFACE

PersonalSense first action is it installing process on the device. It was tested successfully in the Android Emulator⁵ and tablets (Acer Iconia, Memup SlidePad 704CE). The installation process occurs by installing the package directly from the development platform in this case, Xamarin Studio.

In settings area of Android device in section AppInfo, *PersonalSense* is described with some default configuration it belongs, such as the permission to have full access to the network illustrated in Figure 12, important to demonstrate its presence and be listened in the network by other devices.

For the data configuration in the network, *PersonalSense* provides a folder where the data will be stored which will act as a data server. Following the wizard will bring some options representing the info in the folder that becomes available to send the info to the network and also in this same wizard appear the selection of full access to read and write, with permission of users to change the files. In some devices is only necessary to go into the file manager, locate the share folder and use it like a shared database that will be available to any user who is working at the station.

⁵ (AVD, 2014)– Android Virtual Device Manager, provides a graphical user interface in which can be created and managed Android Virtual Devices, required by Andr Figure 18: *PersonalSense* Settings applications without using a physical device. details

4.5.2. USER INTERFACE

As a shared content middleware PersonalSense focus will stays on its user interfaces. To facilitate the user interaction with the tool, providing him with a simple and standardized interface with module and the data to be used some steps must be followed.

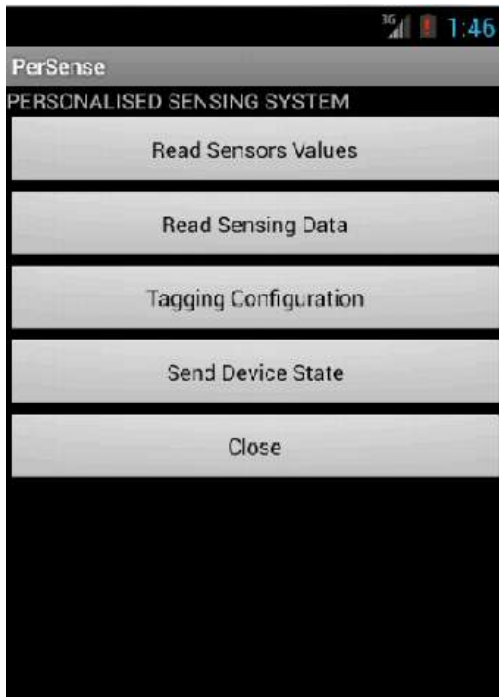


Figure 19: PersonalSense User Interface

Based on the way people think and work with mobile devices, PersonalSense UI intends to be pretty intuitive, highlighting the functionality of an application and enabling a positive emotional connection with users. PersonalSense assumes nature of the ease of use of the applications and their similarity in use, does not waste time or availability factor of reading, as well as the space available for the display of contents.

Other aspects considered were the uniform presentation of the terminology associated with textual icons used and the custom UI elements, maintaining meaning and coherence of actions throughout the application, providing to the user an immediate feedback of their actions.

In dialogue with the user through messages, the use of common terms such as "*Share Content*", "*Movement*" or "*Sensing*" will assist the application interaction with the user.

The interaction process is initialized and controlled by the user. The application merely suggests the following actions being the same route using the whole of the user's will. The aim is to make the user become familiar with the behaviors and PersonalSense controls.

As shown in Figure 19, "*Read Sensor Values*" will initialize the main system process by reading the classification file and returning the inference process presenting the perfil components suggested by PersonalSense, such as the state of device, the event occurring at that moment and the data type suggested that device to receive.

After the contents be available in the network devices with the same platform infrastructure will receive alerts only about contents preconfigured based on their specific interests.

In the case the user choose, "*Read Sensing Data*", for the process occur automatically, another graphical interface will be provided, in which will be applied the classification algorithm. The same file is then converted for the approved reading format of learning algorithm.

In sequence will be obtained the 'best' attribute value with results of classification process, the fields of the interest form will automatically be populated, with the references of sensor type and the state field will determine the type of data that will be available on the network as referenced in table 7.

4.6. SUMMARY

The middleware PersonalSense classifies in events sensory capabilities and communication supported by the device. Sensory capacity will be accessed through a partner middleware, called Maestro developed in the project Citysense of SITILabs or through an emulation local sensory data. The communication skills will be emulated through a system of data sharing, called ICON that was developed in the project DTN-Amazon of SITILabs.

The development line followed by PersonalSense suggested its interaction with Maestro middleware. The sensing data file provided by this middleware must be taken, in time to time and periodically in a pull-type approach in a specific system directory file (/Maestro/Sensors/XmlFiles) in the mobile device in a XML format identified by sensor name. For this Dissertation sensing file data collected from Maestro will correspond with accelerometer sensor which main features this sensor presents. After *PersonalSense* classification process tagging interface (section 4.3) will be able to consider the state of the end-device to upload the specific data as shown in table 7, the tagging table.

Sensing module, tagging and inferring modules created with C# language with a SDK of Android in Xamarin Studio were crucial to this work be done, given a main idea of the interest in study of the behaviors of devices even its users in such network environment.

Some concerns during this work could be described as:

- The inexistence of a standard platform able to mix a classification algorithm with a mobile development language;
- Testing the application in an emulator even in a real mobile device, taking long times of debugging the application;
- Development platforms for mobile devices are selling their SDK when developer wants to for more period of time.

Built with SDK of Mono for Android, that is made up of a set of assemblies, namespaces, and classes that are optimized for mobile platforms. It works across Visual Studio and MonoDevelop, as well as a plug-in for operating systems other than Windows.

The choose to build a native application is their support for device features such as the accelerometer, file system, camera, cross-domain web services, and other features that are not available in HTML and JavaScript. Also, they do not depend on the wireless network to be loaded, whereas a web application is dependent on the wireless network for nearly everything.

An application made in Android gives the developer a lot a freedom do share data, behaviors or even create something of a distributed application because of its core components that can work independently of each other. Another important issue of Android applications is that it starts only one thread per process and every application runs in its own

instance of the Dalvik virtual machine. This sandboxing method protects your application from being corrupted by other running applications. Every application is protected so that only the device user and the application can access the application's data or resources. In addition, they play a large role in protecting your application data from malicious attacks.

5. CONCLUSION

Elliot considers that “*a process consumes time, gives sometimes headache from the creators of products that work for him, but it is an essential element to its success.*” (Elliot, 2012, p. 58). In its similarity with others software products, *PersonalSense* is a middleware that would walk through a process of continuous development because of its characteristics from communication patterns, integration of services and the necessity to follow contributing in continuous investigation at sensor mobiles area, data centric networking approaches and inference paying great attention in the data to obtain information.

Smartphone’s market has attracted increasing interest of mobile user’s and companies, arousing the necessity to manage the information available in the network and the own structure of network without a pre-defined infrastructure known as OppNets. *PersonalSense* born with the idea to use the opportunity encounters between devices to share information based on user’s interests. The ability to interact and share information *PersonalSense* lead this study which consists in the use of sensory device capabilities by applying classification methods and automatically infer the process to the network.

Most current systems are constructed in the form that the user interacts with them, most often because of the necessity of constant interaction with the devices, user’s lose their need to share contents. *PersonalSense* take this advantage with proposition to build a middleware that automatically share the information, using device capabilities, using the ability to adapt the type of data based on the interests of the user's context.

The construction of a pervasive environment, with features such as installation, maintenance, and easy portability, requires a set of aggregations that go from the development environment to the used language, as by the perceived need to be installable on any operating system or higher part of mobile devices. Hence arises the choice for developing on Android, because of their existence in various brands of hardware. The main challenge in building the system was the lack of algorithms of the methods chosen for development in mobile devices, which is why most of the tests have been performed on physical device emulators’ mobile code. The use of Xamarin Studio platform overcomes the problem of portability of the code developed for mobile to different mobile frameworks. Android platform has the main feature that it could run in most of hardware mobile environments, presenting as a suitable feature from developers research. By this token, *PersonalSense* for its existence presents also mono

project line, precisely to present cross-platforms able to enable an application to be scalable, because of its, although this study was not the focus but just being in CitySense Project plan.

The idea of sharing contents using the behaviors of mobile users' based in inferring sensors data received from mobile devices was the main feature of *PersonalSense* existence as a middleware, that this dissertation was related.

In its first steps *PersonalSense* existence was important in research of the main features of development infrastructures and communication to sensing abstraction aspects passing from classification algorithms. According with machine learning techniques, decision trees shows up as a surprisingly versatile tool in the most inductive learning methodologies used today, both in terms of application, work and academic research and are also important when the question to be solved must be readily understood and interpreted by humans. However, decision trees, implemented by simple training algorithms with ID3 or C4.5 could not operate very effective depending on the problem proposed and the data source to be inferred.

6. FUTURE WORK

For future work, *PersonalSense* the project should be directed to the need to adapt a standard classifier for reading sensors in mobile devices.

REFERENCES

- Alvares, R. (2005). *Investigação do Processo de Stemming na Língua Portuguesa*. Dissertation presented to Programa de Pós-Graduação em Computação da Universidade Federal Fluminense to obtain Master Graduation, oriented by Ana Garcia, Niterói.
- American Heritage Dictionary. (2009). *The Free Dictionary*. (H. M. Company, Ed.) Obtido em 17 de February de 2014, de The Free Dictionary: <http://www.thefreedictionary.com/inference>
- AVD. (2014). Obtido em 2 de February de 2014, de Android Developers: <http://developer.android.com>
- Barbosa, R. (2012). *Sensing Middleware Collect and Share*. Lisboa: ECATI - ULHT.
- CCNx. (2013). Obtido em 11 de June de 2013, de CCNx: <http://www.ccnx.org/>
- DTN Project. (2013). Obtido em 10 de June de 2013, de Sprout: <http://sprout.ics.uci.edu/projects/dtn/>
- Gama, C. F. (2012). Extração de Conhecimento de Dados, Data Mining. In J. C. Gama, *Extração de Conhecimento de Dados, Data Mining* (p. 101). Lisboa: Silabo.
- Ghahramani, Z. (16 de September de 2004). Unsupervised Learning. Universiy College London, London, UK.
- Griffin, N. L., & Lewis, F. D. (s.d.). *University of Kentucky*. Obtido em 17 de February de 2014, de University of Kentucky: <http://www.cs.uky.edu/~lewis/papers/inf-engine.pdf>
- Haggle - A content-centric network architecture for opportunistic communication*. (2013). Obtido em 10 de June de 2013, de <https://code.google.com/p/haggle/>
- Kaelbling, L. P., Michael, L. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence* , 4.
- Mono*. (May de 2011). Obtido em 13 de June de 2012, de Mono: http://www.mono-project.com/Main_Page
- Morais, N. (2013). *ICON*. Lisboa: SITILabs.
- NetInf*. (4 de February de 2014). Obtido em 4 de February de 2013, de NetInf: <http://www.netinf.org/>
- Pierce, B. (2002). *The Abstraction Principles Reads*.
- Quinlan, R. (March de 2013). *Ross Quinlan*. Obtido em 2013 de June de 12, de Ross Quinlan: <http://www.rulequest.com/Personal/>

Schmidt, D. (1994). *The Structure of Typed Programming Languages*. MIT Press.

SITI Labs. (2013). Obtido em 2013 de June de 11, de SITI Labs: <http://siti.ulusofona.pt/>

Smart Pin. (17 de February de 2009). Obtido em 11 de June de 2013, de Seung-Bum Lee: <http://www.eeng.dcu.ie/~sblee/research.html>

The Free Dictionary. (2009). Obtido em 5 de February de 2014, de The Free Dictionary: <http://www.thefreedictionary.com/inference>

Zhu, X. (19 de July de 2008). *University of Wiscosin*. (U. o. Madison, Ed.) Obtido em 23 de 02 de 2014, de University of Wiscosin: http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf

Witten, H., Frank, E., & Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

ANEXES

ANNEX 1: SERVICES ORIENTED IN OPPORTUNISTIC NETWORKING

SERVICES ORIENTED IN OPPORTUNISTIC NETWORKING

nobre.pedro@gmail.com

January 23, 2011

Abstract

With the growth of social networks mobile devices are rapidly becoming the preferred mean of interactions' device of communication in people's lives.

Developing architectures with the capability to support an integrated connectivity with devices who have embedded sensors, such as accelerometer, digital compass, gyroscope, GPS, microphone, camera, and connectivity with IEEE 802.11, Bluetooth, Wi-Fi, GPRS, or other, to make the contents sharing more frequent and extended even to old devices, enabling them to choose the best available connectivity and protocol for situation and user-specific policies, to run in several platforms', applications could have the capability to exchange pictures, news, messages, songs or videos between mobile devices based on the users' interests. This kind of service architecture intended to reduce the client cost in the Internet connectivity and the believe that it will revolutionize many sectors like economy, business, healthcare, social networks, environmental monitoring, and transportation making the devices more interactive in sharing content by concern without necessity to have a permanent access to the World Wide Web.

This document gets a focus on the possibility to implement an networking model behind Services Oriented providing users to experiment exchange resources directly across environments with or without infrastructure and in a highly dynamic mobile environments.