NTNU
Norwegian University of
Science and Technology

Cognitive Radio and TV White Space Communications

TV White Space Geo-location Database System

Naroa Zurutuza

Master of Science in Communication Technology

Submission date:	June 2011
Supervisor:	Tor A Fjeldly, IET
Co-supervisor:	Pål Grønsund, Telenor R&I
	Paal Engelstad, Telenor

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

On September 23rd 2010, the US regulator FCC opened up for cognitive radio operation in unused spectrum of the TV bands, the so called TV white spaces, by allowing wireless users to access a geo-location database to download information about what spectrum is available. A similar approach seems to be taken by OFCOM in the UK.

Big actors from the IT-industry such as Google, Microsoft and Dell as well as actors from the communication-industry such as Motorola are also researching and pushing this technology.

The aim of this Thesis is to research in the use of those TV white spaces for wireless communications, focusing on the database system. For that, a geo-location database system is going to be implemented for TV white space access, wireless communication is going to be studied researching issues related to TV white space access.

The first part of the work is to study background material and to survey the state of the art on cognitive radio and TV white space communications. Both academic and commercial approaches, as well as standardization and regulations, should be studied to better understand the requirements for the TV white space database system. Then, the actual TV white space database system will be designed and optimized considering all requirements for such a system. This includes detailed design of both the actual SQL database and the design of the TV white space database system. The student will find the appropriate programming language, tools and framework. If time, the student will attempt on an implementation of the designed system. If successful, the student could also provide a simple demonstration of the system by implementing a web interface for finding TV white spaces in a given area.

Assignment given: 17. January 2011
Supervisor: Tor A. Fjeldly, IET
Co-supervisors: Pål Grønsund and Paal Engelstad, Telenor

Preface

This Master Thesis for the Norwegian University of Science and Technology (NTNU) was developed in collaboration with Telenor and carried out at University Graduate Center at Kjeller (UNIK).

Special thanks to my supervisors at Telenor Pål Grønsund and Paal Engelstad for all their guidance, attention and collaboration. Also special thanks to Tor A. Fjeldly for his help.

Abstract

The aim of this thesis is to research in the use of emerging TV white space communications, implementing a geo-location database system. For that, some research and theoretical studies related to cognitive radio and TV white space communications will be done first, focusing on current activities, standarization processes, commercial approaches and related projects. Once the background and the present TV white space communications status is analyzed, a geo-location database system will be designed and developed to prove the potential of this technology. The operation of the database system will be demonstrated through a web interface. In this way, an open and publicly accessible geo-location database system implementation and structure will be created (note that even if several database system creation initiatives are taking place, most of them are private). However, due to the lack of official regulatories, established standards, and actual transmission data (data from TV broadcasters, wireless microphones etc.), only an initial TV white space database system demo will be implemented to model the operation of the same. It will be possible to access and query this database system through a simple web interface for the Oslo area. After analyzing the results of the implementation and looking to other TV white space initiatives, some considerations for future work will be concluded.

Contents

Notation

Abreviations

A/D	Analog to Digital
BAS	Broadcast Auxiliary Service
BER	Bit Error Rate
BS	Base Station
CCC	Common Control Channel
CDB	Central DataBase
CDBS	Commision's Consolidated DataBase System
CEPT	European Conference of Postal and Telecommu nications Administrations
CMRS	Commercial Mobile Radio Service
CPE	Customer Premises Equipment
CR	Cognitive Radio
CTS	Clear-To-Send
DAO	Data Access Object
DPC	Dirty Paper Coding
DSA	Dynamic Spectrum Access
DTT	Digital Terrestrial Television
DTV	Ditigal TV
DVB	Digital Video Broadcasting
EM	ElectroMagnetic
ENG	Electronic News Gathering
ERP	Effective Radiated Power
FCC	Federal Communication Commission
FTP	File Transfer Protocol
HAAT	Height Above Average Terrain
IEEE	Institute of Electrical and Electronics Engineers
IoC	Inversion of Control
ISM	Industial, Scientific, Medical
IT	Information Technology
ITU	International Telecommunication Union
JDBC	Java DataBase Connectivity
JSP	JavaServer Pages
LDB	Local DataBase
LPTB	Low Power TV

LTE Long Term Evolution
MVC Model View Controller
NTIA National Telecom and Info Administration
OET Office of Engineering and Technology
ORM Object Relational Mapping
ORS Offshore Radiotelephone Service
OSA Opportunistic Spectrum Access
PKI Public Key Infrastructure
PLMRS Private Land Mobile Radio Service
PMSE Programme Making and Special Events
QoS Quality of Service
RAC Real Application Cluster
RF Radio Frequency
ROC Receiver Operating Characteristic
RRS Reconfigurable Radio System
RTS Request-To-Send
SBI Spectrum Bridge, Inc.
SCC Standarization Coordinating Committee
SDR Software Digital Radio
Second MO&O Second Memorandum Opinion and Order
SI Statutory Instrument
SNR Signal-to-Noise Ratio
SOAP Simple Object Access Protocol
SSL Secure Sockets Layer
TC Technical Committee
TLS Transport Layer Security
TVWS TV White Space
UHF Ultra-High Frequency
ULS Universal Lycensing System
UML Universal Modelling Language
UWB Ultra-WideBand
WG Working Group
WRAN Wireless Regional Area Network
WSD White Space Device
xG NeXt Generation

List of Figures

1. INTRODUCTION

In general, traditional wireless communication systems have fixed transmission parameters. In other words, their transmission frequency is fixed and the same in every location and instant of time, determined by regulatory policies. However, the recent popularity of telecommunications and wireless communications, has increased the usage of radio spectrum exponentially. In order to supply all the demand and improve communication parameters and Quality of Service (QoS), new technologies need to be developed. One of the attempts to solve the problem of spectrum lack is Cognitive Radio (CR) and TV white space (TVWS) communications in particular.

With the digital switchover, the so called digital dividend or white spaces appeared in the TV bands. These white spaces are unused frequency bands within the TV transmission spectrum. TVWS communications tries to reuse these unused channels by adapating its transmission parameters to the environment and to avoid causing interference to the primary users of the TV bands. In this way new frequency spectrum for unlicensed users or devices is abilitated.

TVWSs are of special interest because of two main reasons: first of all, their propagation characteristics are specially good for wireless communications, reducing propagation losses and hence, increasing coverage (in fact, TVWS communications are also known as "SuperWiFi"). Second of all, very little and relatively cheap infrastructure is required for their implementation, making them specially suitable for rural and undeveloped areas or countries. In consequence, there are two principal use cases foreseen in the TVWSs. First use case is local communications (so called "SuperWiFi") typically with laptops and handhelds, where a wireless access point will offer access to the Internet using TVWS frequencies. Second use case is wide area broadband in rural areas, with fixed antennas mounted on rooftops. There are also a range of other innovative usages of TVWS that might appear due to the favourable propagation characteristics.

Nevertheless, the main problem of TVWS communications is the possible interference with primary users. Primary users' protection has to be ensured and currently, there are two possible techniques to do it: sensing and geo-location database systems.

The sensing tries to detect radio signals and estimate the relative location of primary users in order not to interfere with them. However, it is not easy to determine the exact location of primary users provoking unwanted interference or white space missdetection. In consequence, alternative options are being researched.

The other technique is to use a geo-location database system to determine the available frequencies in a given location and time. This technology is preferred by the US regulator Federal Communication Commision (FCC) and it is currently under development by commercial companies. In order to know which channel to use in the TVWSs, White Space Devices (WSDs) access and query a geo-location database system indicating their location and the database system responds with a list of available frequencies and allowed transmit powers for that location. The WSD selects one of the available channels and starts its transmission.

The possibility to exploit a new business market and develop a new communications technology makes really interesting the research and development of TVWS communications, with special focus on the geo-location database system. However, lots of work and research is still needed before their complete commercialization.

The main contribution of this thesis to enhance the use of TVWS communications is the design and implementation of a geo-location TVWS database system. Especially, this includes the detailed design and implementation of three main parts of the TVWS database system; (i) the underlying database system, (ii) the program that communicates with the database to determine the available TVWSs and (iii) a web interface that can be used to present the TVWSs for a requested location. A second contribution is a throughout overview of related work and theoretical studies on TVWS research development. It also provides one of the first publicly accessible geo-location database systems so that interested entities can use it as a basis or reference for future developments and improvements.

1.1 SCOPE OF THE THESIS

As commented above, this thesis intends to research in TVWS communications by implementing a geo-location database system. The principal aim of the thesis is not to provide a complete commercial geo-location database system solution but to develop and design an initial database system publicly available for future implementations that, at the same time, proves the potential of TVWSs. This initial database system will try to cover all the possible aspects and problems of TVWS communications and it is intended to be as generic as possible in order to adapt to different regulatory entities and to enable the implementation of additional features, applications and functionalities easily. For that, several steps have been followed during the developement of the thesis.

1.1 OUTLINE OF THE THESIS

In Figure 1.1 the outline of the thesis is presented. After exposing and defining the aim of the thesis and the interest and motivation for its development in Section 1, some background research work in the subject has been done in Section 2. It starts from the most general concepts of CR (Section 2.1), passes through global terms of white spaces (Section 2.2) and limits and tradeoffs of CR and sensing technique (2.3) and ends up with the geo-location database system case (Section 2.4). Finally, the standarization process of CR and TVWS communications (Section 2.5) and some practical or commercial approaches of TVWS database systems (Section 2.6) are explained.

In order to get deeper knowledge in the geo-location database system area of interest, a more detailed theoretical study has been performed in Section 3. In fact, the proposals from several database adminstrators in the US have been analyzed in Section 3.1 to be aware of the main considerations to take into account when implementing the database system.

The background theory, commercial approaches, standards and theoretical studies from previous sections are used to develop the geo-location database system in Section 4. First of all, the initial geo-location database system design has to be defined and the tools, frameworks and programming languages for its implementation need to be selected (Section 4.1). Then, the database metastructure that supports the previously defined database system is determined (Section 4.2). The next step, exposed in Section 4.3, is to create and populate the actual physical database. Next, the initial code skeleton is defined (Section 4.4) and implemented (Section 4.5). The last step is to create a user interface to interact with the database system implemented (Section 4.6).

In Section 5 the operation of the implemented TVWS geo-location database system is demonstrated and the results are analyzed.

From the results got in Section 5 some conclusions (Section 6) and considerations for future work (Section 7) are concluded.

Finally, part of the code is included in the appendixes in order to support and facilitate future developments. Appendix A contains the code for the MySQL database creation, Appendix B contains code from the interfaces used for the database system development (implementation code is not included due to its large extention) and Appendix C contains code from the user interface.

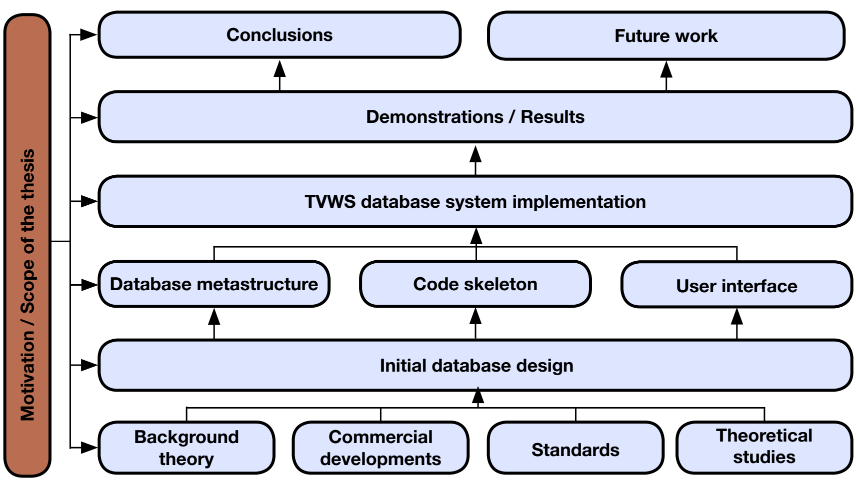

Figure 1.1 Structure of the scope of the thesis

2. RELATED WORK

Nowadays traditional wireless communications is based on static or fixed frequency allocation. Spectrum is regulated so that most of the bands are allocated exclusively to a single system licensed to use that band in any given location, this allocation is fixed and independent from location. Unfortunately, the reserved frequencies are not in use most of the time, deriving in spectrum inefficiency as shown in Figure 2.1. According to FCC, the utilization of fixed spectrum assignment is approximately 15-85% depending on space and time [1].

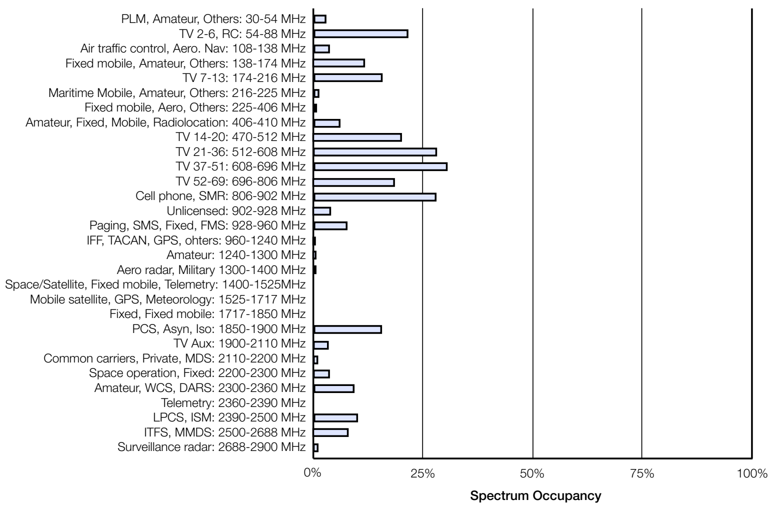

Figure 2.1 Measured spectrum occupancy averaged over seven locations [1]

Since the number of wireless networks and users is increasing exponentially, the available frequency spectrum is running out. To optimize its use, new opportunistic spectrum allocation (OSA) techniques such as Dynamic Spectrum Access (DSA) and CR, are being developed.

CR is the technology that makes possible the DAS, providing the capability to share the wireless channel with licensed users in an opportunistic manner [2]. CR networks will offer high bandwidth to mobile users, making it possible to first detect the available spectrum portions and other licensed users (spectrum sensing), then select the best available channel (spectrum decision), coordinate its access with other users (spectrum sharing) and liberate the channel in case that a licensed user is detected (channel mobility) [3]- concep shown in Figure 2.2 and Figure 2.3-. Different DAS techniques allow CR to operate in the best available channel.

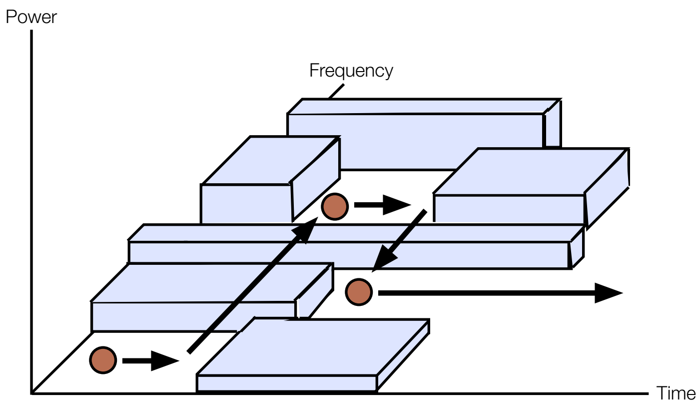

Figure 2.2 Dynamic spectrum access concept

However, since the existing wireless networks were designed to work with static frequency allocation, some coexistence problems between conventional and CR networks have to be solved. Due to the fluctuating nature of the available spectrum and the diverse QoS requirements of different applications, CR networks need to handle several challenges.

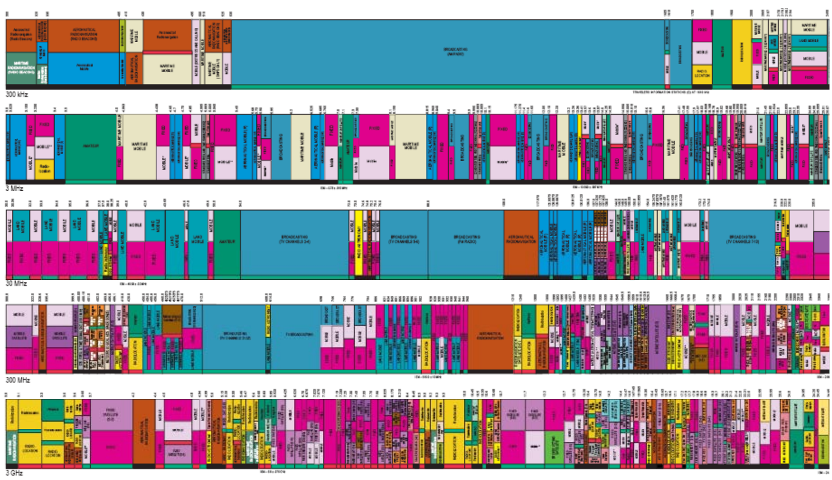

Figure 2.3 Cognitive radio concept [2]

2.1 COGNITIVE RADIO

A "Cognitive Radio" is a radio that can change its transmitter parameters based on interaction with the environment in which it operates. (FCC). In other words, a type of radio that can sense and autonomously reason about its environment and adapt accordingly. This radio could employ knowledge representation, automated reasoning, and machine learning mechanisms in establishing conducting or terminating communication or networking functions with other radios. Cognitive radio s can be trained to dynamically and autonomously adjust its operating parameters. (IEEE 1900.1 Group)

2.1.1 DYNAMIC SPECTRUM ACCESS

As mentioned above, DAS improves the inefficiency of static spectrum access by providing opportunistic access to the available spectrum. There are three main different dynamic spectrum access strategies [1] as shown in Figure 2.4.

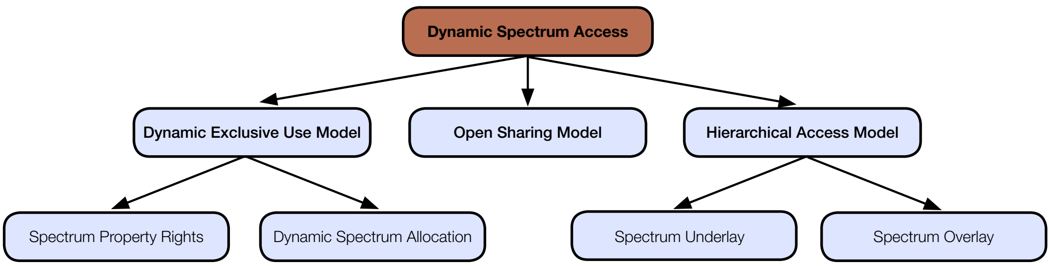

Figure 2.4 Dynamic spectrum access classification

Dynamic exclusive use model

This model tries to maintain the basic structure of the current static spectrum management policy. While maintaining the exclusive access to licensed users, it gives more flexibility to the allocation and use of the spectrum.

We can distinguish two groups under this model, Spectrum property rights, and Dynamic spectrum allocation. The Spectrum property rights allows the licensed users to sell spectrum and choose technology. In this way, the market is supposed to select the most economically profitable distribution. The Dynamic spectrum allocation assigns spectrum dynamically to different services in order to optimize spatial and temporal statistics.

Open sharing model

This model tries to open the sharing model among peer users or spectrum commons by developing unlicensed Industrial, Scientific and Medical (ISM) bands. Centralized or distributed sharing models can be applied.

Hierarchical access model

This model provides a hierarchical access to primary and secondary users. Secondary users could use the spectrum while keeping interference with primary users limited. Two hierarchical models have been developed: Spectrum underlay and Spectrum overlay.

The Spectrum underlay imposes constraints on the transmitted power of the secondary users in order not to exceed the permitted interference with primary users. One way of keeping the transmit power low and reaching high data rates is to spread the transmitted signal over a ultra-wide frequency band (UWB).

The Spectrum overlay imposes constraints on when and where to transmit, exploiting local and instantaneous spectrum availability (OSA).

To improve efficiency, both the underlay and overlay models can be combined.

2.1.2 COGNITIVE RADIO TECHNOLOGY

CR, built on a software radio platform, is a context-aware intelligent radio potentially capable of autonomous reconfiguration by learning from and adapting to the communication environment. Since a CR is a radio that can change its transmitter parameters based on the interaction with its environment, CR should fulfil two main requirements: cognitive capability and reconfigurability [3].

The cognitive capability identifies the spectrum portions that are available in a specific moment and place through the interaction with the environment. These available spectrum portions are the so called spectrum holes or white spaces.

In a CR network, it should also be possible to transmit and receive through different frequency values using different access technologies. That is why the parameters of a CR can be modified to adapt to the environment and use the best frequency band. This ability is called reconfigurability.

Architecture for cognitive radio networks

CR networks can be divided in two groups, the primary network and the cognitive network. The primary network is a licensed network that has exclusive right to access to a specific frequency band. Cognitive networks do not have a license to operate in the desired band, and is often referred to as the secondary network.

The fundamental components and architecture of a CR network, as defined by Akyildiz [3] and much used in the litterature, are represented in Figure 2.5 and are the following:

· Primary User: A primary user has a license to operate in a certain spectrum band. Its access can be only controlled by the base-station and should not be affected by other unauthorized users.

· Primary Base-Station: Primary base-station is a fixed infrastructure network component with a spectrum license. Sometimes, primary base-station may require both licensed and CR protocols for the primary network access of CR users.

· Cognitive Radio User: Is an unlicensed user, so the spectrum access is allowed only opportunistically. The CR user should have the capabilities of spectrum sensing, spectrum decision and spectrum mobility. It has to be able to communicate with other CR users apart from the base-station.

· Cognitive Radio Base-Station: CR base-station is a fixed infrastructure component with CR capabilities. It provides single hop connection to CR users without spectrum access license.

As mentioned, CR users can communicate both with the base-station and other CR users. In consequence, there are three possible access types in a CR network [3]:

· Cognitive Radio Network Access: This access occurs when a CR user accesses its own CR base-station. All the operations take place inside the CR network so their medium access scheme is independent of the primary network's.

· Cognitive Radio Ad Hoc Access: This access occurs when a CR user communicates with other CR users through ad hoc connection [4, 5]. CR users can have their own medium access technology.

· Primary Network Access: This access occurs when a CR user accesses the primary base-station through the licensed band. In this case, CR users should support the medium access technology of primary network and the primary base-station should support CR capabilities.

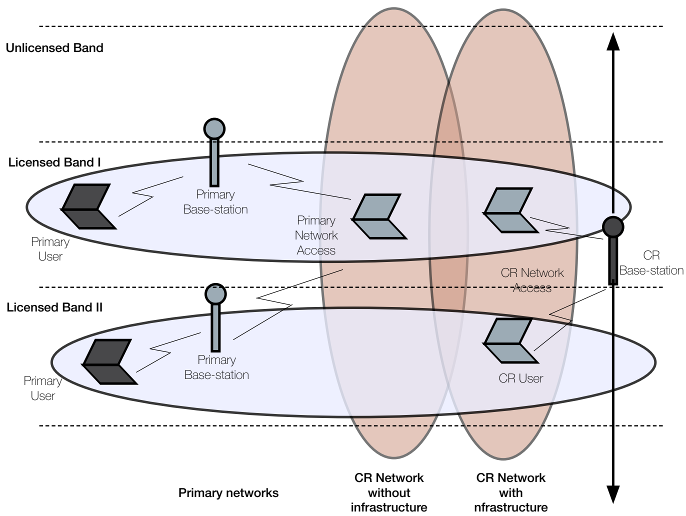

Figure 2.5 Architecture for cognitive radio networks

Spectrum management framework for cognitive radio networks

CR has to coexist with other licensed and non-licensed users being the main challenges of a CR network [3]:

· Interference Avoidance with primary networks.

· QoS awareness in order to decide an appropriate spectrum band, considering dynamic and heterogeneous spectrum environment.

· Seamless Communication, regardless of the appearance of the primary users.

To address these challenges, four main functionalities take place in the CR spectrum management: spectrum sensing, spectrum decision, spectrum sharing and spectrum mobility.

Spectrum sensing

A CR needs to monitor and sense the available spectrum bands in order to detect the spectrum holes or free portions. There are different sensing methods such as primary transmitter detection, primary receiver detection or interference temperature management.

However, when implementing real sensing techniques with energy detection, CR users cannot distinguish between primary signals and other CR user signals, becoming even more difficult the hole detection.

On the other hand, sensing and transmission cannot be performed at the same time. In consequence, during the sensing time, all the CR users have to stop transmitting decreasing efficiency. Furthermore, the interaction between primary and CR networks is poor and CR users cannot determine the exact position of primary users and interference measures are not precise.

Efficient spectrum sensing is essential in CR so it is important to find the optimal sensing technique with the optimal sensing period and observation time that maximizes the efficiency maintaining a given interference level.

Spectrum decision

The spectrum sensing in a CR network will provide several unused bands spread over a wide frequency range with different channel characteristics. The CR user should be able to select the most suitable unused spectrum band for each specific application. This is called spectrum decision.

There are different criteria such as, bit error rate (BER), capacity, path loss, interference, link layer delay, primary user appearance probability etc. to decide the best available spectrum band. Once the available channels are characterized, the most suitable one according to the QoS requirements of the application is selected.

Spectrum sharing

Since there are many CR users that may access the same frequency bands at the same time and location, a CR user needs to coordinate its access with other users. The spectrum availability changes over time and space and in consequence, a dynamic inter-cell spectrum sharing is needed in a CR network. The spectrum sharing technique should maximize the cell capacity, minimize interference to neighbour users and protect primary users. There are different types of spectrum sharing techniques.

The spectrum sharing techniques can be centralized (the spectrum allocation and access are coordinated by a central user) or distributed (spectrum allocation and access are coordinated distributed through different nodes), cooperative (measures in different nodes are considered) or non-cooperative (measures in a single-node are taken into account), intranetwork (allocation within the components of the same CR network) or internetwork (allocation within components of different CR networks).

A common control channel (CCC) would simplify the sharing process. Nevertheless, since whenever a primary users accesses a frequency band CR users should vacate this band, it becomes quite difficult to implement the CCC.

Spectrum mobility

The licensed users have priority over non-licensed users when accessing a certain frequency band. In fact, CR users must switch to a new spectrum band as soon as they detect the presence of a licensed user. This process is called spectrum mobility.

The spectrum mobility implies a spectrum handoff where the parameters of the different layers' protocols need to be updated. The spectrum mobility management should offer the fastest handoff possible with the minimum performance degradation.

2.2 SPECTRUM HOLE OR WHITE SPACE

A band of spectrum can be considered underused if it can accept secondary transmissions without harming the operation of primary transmissions. The region of space–time–frequency in which secondary use is possible is called a spectrum hole [6].

Even if a frequency is free in a given time, the use of this frequency by secondary users can cause interference to adjacent frequency users, so the objective is to provide sufficient benefit to secondary users while protecting primary users from interference. That is why a spectrum hole is defined as a frequency band in which a secondary can transmit without interfering with any primary receivers. In Figure 2.6, secondary user A wants to transmit to secondary user B but for that it has to ensure that it will not interfere with primary transmitters and receivers [1].

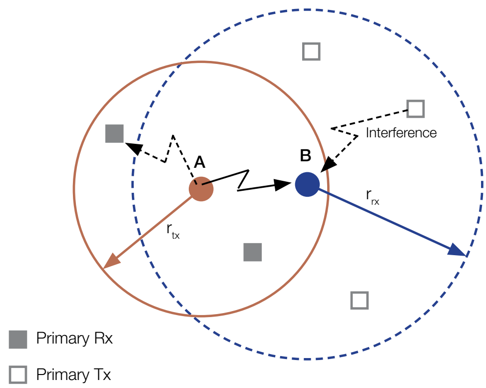

Figure 2.6 Spectrum hole transmission and interference areas

These holes appear because the regulatory policies assign frequencies over several years and long distances although real communications usually take place over few seconds and meters. However, it is not so easy to change the policies because the life of technology is determined by the economical aspect of the same and some technology has to be active enough time to achieve a return on the infrastructure and equipment investments. Dynamic access techniques are used to take advantage of the spectrum holes without causing interference to adjacent users.

2.2.1 TV WHITE SPACE

The spectrum holes localized within the TV spectrum range are known as TVWSs. TV broadcast bands have special interest since new approaches for TV-band spectrum holes for enabling wide-area Internet services are being considered and their transmission characteristics are really good for radio communications. In fact, FCC has approved the dynamic access of unlicensed users in TVWSs and the Institute of Electrical and Electronics Engineers (IEEE) is developing the 802.22 and 802.11af standards.

When analogue TV was replaced by digital TV, many channels and TV frequency bands happened to be available. The fact that TV frequency bands offer better characteristics and conditions to wireless communications than the ones that are currently in use and the need of more frequency bands to supply the increasing demand, provoked the development of new technologies that intend to take advantage of the available TVWSs [7].

Until now, most of the research has been done in the US regulated by the FCC. To visualize the amount of underused frequency bands, the TV signals are analyzed in Figure 2.7, where the number of available TVWSs in each area of the United States is represented. It is clear that this number is quiet high, so if a way to reuse them is developed, the spectral efficiency would increase considerably.

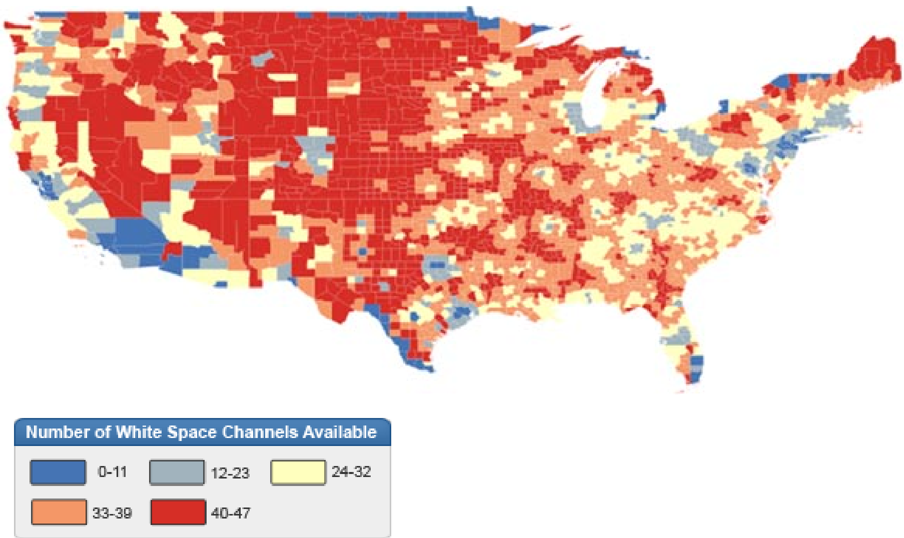

Figure 2.7 TV white space channels in USA December 2009 [7]

In Figure 2.8, the map from Figure 2.7is plotted as a graphic with the available channels from Berkeley (California) to Washington. The upper blue curve shows the actually available number of channels based on the International Telecommunications Union (ITU) models for wireless signal propagation. The lower tan curve represents the opportunity based in the actual IEEE 802.22 standard, where a CR user can access a channel only if it is sufficiently empty. The 802.22 standard requires a sensitivity of -116 dBm, what avoids any interference but also makes that many channels that are safe to use are discarded because they are above -116 dBm. Indeed, sampling the United States uniformly by area, 56% of the total 67 TV channels are available on average and only 22% can be recovered using the -116 dBm rule [7].

Figure 2.8 Number of free channels from Berkeley to Washington [7]

2.2.2 HOW TO DETECT A SPECTRUM HOLE

First, lets consider an ideal detector that informs if it is possible for a secondary user to access a certain frequency in a given space-time value. If the detector mistakes a hole as occupied, a spectrum opportunity is lost because of a false alarm and when the detector mistakes a occupied channel as hole, there is a missdetection and a collision may occur. As is shown in Figure 2.9, a smaller false alarm probability ξ implies a larger missdetection probability δ and vice versa. A compromise between the two criteria should be adopted [1].

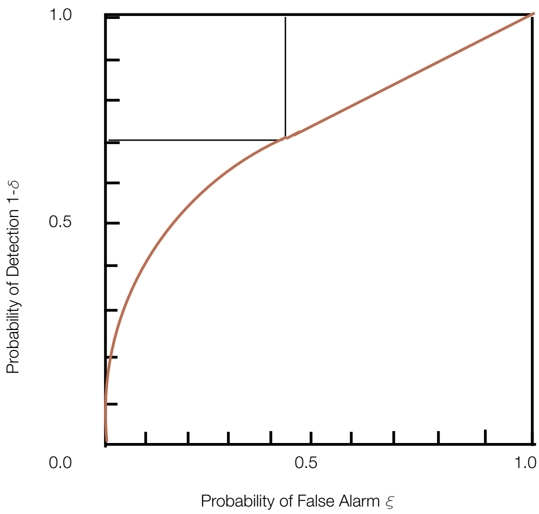

Figure 2.9 Probability of hole detection

In Figure 2.10 a primary transmitter and receiver are visualized [6]. Without interference, a receiver within the red circle with radius r_{dec} can decode the signal coming from the transmitter. A receiver outside the red circle cannot decode the signal. In order to allow secondary users, some interference should be tolerated. The blue circle represents the protected zone within primary receivers are guaranteed to decode the transmitted signal. The space between both circles is called "sacrificial zone" and a primary receiver has no guarantees to decode the transmitted signal when a secondary user is working there. To protect the primary receivers, a no-talk region is established around them (white circle) inside which secondary users cannot transmit. In Figure 2.10a, the transmit power of the secondary user is low so the no-talk region is relatively small. On the other hand, when the transmit power of the secondary user is high, the no-talk region is bigger (Figure 2.10b).

It can be concluded that since the global no-talk area is the union of the no-talk zones of all the primary receivers, a hole is the supplement of the same. To make possible for secondary users to detect these holes, they should have information about the location of the primary receivers. To get this information, primary participation is necessary and the secondary system should be able to determine the location of primary receivers by sensing them.

Another complication appears when secondary users cannot decode primary transmitter's signal, considering it as noise or interference and may deciding not to transmit within the spectrum hole. Some techniques like dirty paper coding (DPC) or partial knowledge of the transmitter codebook have been proposed to solve the problem. However, they are not robust enough.

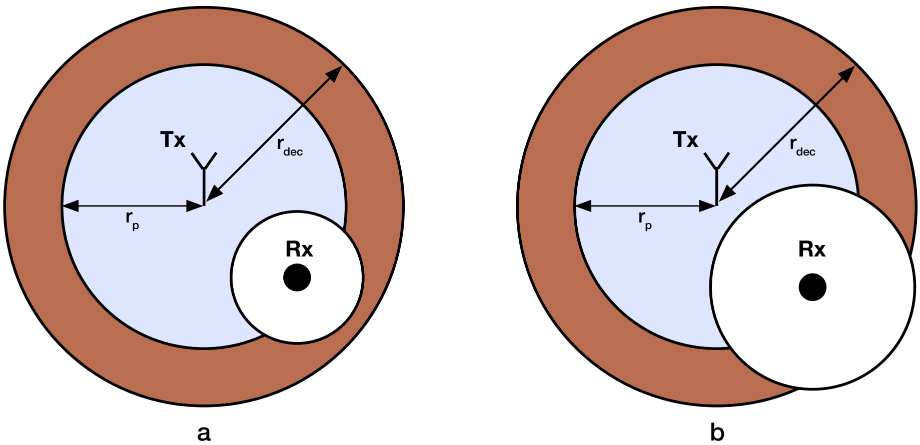

Figure 2.10 Primary transmitter's protection area and receivers decoding area for low and high transmit powers

To define the interference constraint at least two parameters are needed: the maximum interference level η perceived by a primary receiver and the maximum probability that the interference at a primary receiver may exceed η [1]. These two parameters will vary depending on the application, location, traffic etc. If the interference constraint is strict, many spectrum holes will not be detected while if the interference constraint is tolerant, problems with licensed bands could appear.

2.2.3 METRICS AND MODELS

A secondary user needs to determine its position relative to the primary transmitters and detect the primary receivers in order to make sure that it will not interfere with them [8].

However, it is not easy to determine the presence of primary transmitters and receivers. To detect primary transmitters, signal detection techniques such as energy detector, matched filter or cyclostationary detector can be used.

There are different methods to detect white spaces and different metrics are used to measure their performance and compare them [6].

SNR as Proxy for distance

The secondary user measures the power of the transmit signal and uses it to determine its distance from the primary transmitter. It is important to decide what is the maximum transmit power bellow which it is possible to the secondary user to transmit without causing interference. If the primary transmit power is p_t, the attenuation factor is α and the measured primary power at the secondary user is P, the secondary user transmit threshold can be defined as [6]:

$$P \underset{\text{use}}{\overset{\text{do not use}}{\gtrless}} p_t - 10\log_{10}\left(r_n^{\alpha}\right) \tag{2.1}$$

This condition assumes that it is possible for the secondary user to perfectly determine its relative position respect the primary users with just measuring the received signal strength. Unfortunately, the transmit signal suffers from multipath and shadowing, decreasing the received power. This can make that the secondary user assumes that it is outside the no-talk region (because of the low received power) when it is inside. To avoid the shadowing and multipath effects, a budget parameter is introduced getting the following condition [6]:

$$P \underset{\text{use}}{\overset{\text{do not use}}{\gtrless}} p_t - \left(10\log_{10} r_n^{\alpha} + \Delta\right) \tag{2.2}$$

Δ is a safety factor. If it is small, the multipath and shadowing effects will not be completely cancelled and in consequence, the secondary user may sometimes interfere the primary user but most of the holes will be detected. If Δ is big, non-interference will be ensured at the expense of loosing some holes.

Traditional sensing metrics

· Receiver operating characteristic (ROC): the ROC is a curve that plots the P_{MD} as function of P_{FA} for a fixed sensing time and signal-to-noise ratio (SNR).

· Sensitivity: the lowest value of the operating SNR for which the detector satisfies a given target P_{MD} and P_{FA}.

· Sample complexity: the overhead caused by the sensing time required to achieve a target P_{MD} and P_{FA} for a given SNR.

· Robustness to uncertainties: the detector should be robust to device-level uncertainties like the noise level and to system-level uncertainties like shadowing. Device-level uncertainties can be handled by modifying the traditional metrics like the sensitivity or the SNR wall. To deal with system-level uncertainties, they should be included in the specifications. In 802.22 for example, a -116 dBm sensitivity is required (a safety margin of 20dB). But as mentioned, this implies that many valid white spaces will be discarded.

New system-level metrics

· Safety: it calculates the probability of interference. The probability of potential interference is the probability that a secondary user located in the no-talk region declares it unused. For that we also define the probability of shadowing and multipath. This probability depends on the assumed model of shadowing and multipath. We get that [6]:

$$F_{HI} = \sup_{0 \le r \le r_n} \sup_{F_r \in \mathbb{F}_r} \mathcal{P}_{F_r}(D = 0 | r_{actual} = r). \quad (2.3)$$

Where the outer supremum shows the uncertainty in secondary user decisions and the inner supremum shows the uncertainty in the fading distribution.

· Performance: the ability of a secondary user to identify spectrum holes. With only one primary transmitter a opportunity should be defined as [6]:

$$P_{FH}(r) = \mathcal{P}_{F_r}(D = 0 | r_{actual} = r), \quad r > r_n. \quad (2.4)$$

The secondary users are also uncertain to the fading distribution so this uncertainty has to be taken into account. We want to combine P_{FH} into a performance metric that allows the comparison between different sensing algorithms. This metric is called WPAR.

· WPAR:

$$\text{WPAR} = \int_{r_n}^{\infty} P_{FH}(r) w(r) r dr \quad (2.5)$$

Models for fading uncertainity

The received signal power P can be defined as [6]:

$$P = \bar{p_t} - (l(r) + \bar{S} + M) \quad (2.6)$$

Where p_t is the transmitted power, l(r) is the loss due to the attenuation at a distance r, S is the shadowing loss and M is the fading loss. It is possible to distinguish between nominal and quantile models for M and S.

The uncertainity in the primary users location, the fading and shadowing result in a larger no-talk zone. Once a secondary transmitter detects a spectrum hole according to the metrics it has established, it sends a short request-to-send (RTS) to the receiver. If the receiver gets the message successfully, it is checked that the channel is free to use and it answers with a clear-to-send (CTS) message.

2.3 LIMITS AND TRADEOFFS OF COGNITIVE RADIO

Since CR is a new technology, many challenges and fundamental limits are still not solved. Some design tradeoffs have to be improved in order to satisfy the basic requirements [9, 10].

2.3.1 GEOGRAPHICAL AND DETECTION TRADEOFFS

In section 2.2.3 it has been explained that the difficulties in establishing the exact location of primary users and the shadowing and multipath effects increase the no-talk zone considerably. In occasions, the no-talk zone is so large that spectrum opportunities are lost even if the secondary user transmission will not cause any interference to primary users.

The performance of the detector is limited by the capability to detect signals of the same. It can be demonstrated that at low SNR, it is hard to detect zero-mean signals even if the modulation is known. To improve the performance of the detector, a pilot signal can be transmitted. With transmitting a pilot signal with the transmissions, a suboptimal detector can be constructed reducing the required number of detection samples.

2.3.2 NOISE UNCERTAINTY AND QUANTIZATION

Ideally, the receiver's noise is Gaussian and the variance of the same is known. However, in reality, the noise is almost Gaussian and the variance is unknown. This implies that when the SNR is very low the detector cannot detect the signal [10].

$$SNR_{wall} = 10\log_{10}\left[10^{(x/10)} - 1\right] \qquad (2.7)$$

The use of a pilot signal solves the problem of the noise because it is not probable that the noise picks an exact pilot signal.

Moreover, most of the receivers have an Analog to Digital (A/D) converter after the Radio Frequency (RF) block. The sampling or quantization of the analog signal derives in additional received noise. In the same way as explained in the previous section, this additional noise induces an SNR loss in non-coherent detectors.

2.3.3 SINGLE RADIO DETECTOR: LIMITS ON ROBUST SENSING

We have seen that the use of a coherent detector solves the problem of noise uncertainty in non-coherent detectors. However, the shadowing and fading effects cannot be managed by the coherent detector. Furthermore, complexity and clock-instability impose more limits to the coherent detection. These limits mean that the processing gain is also limited and that cooperation is necessary to achieve optimal performance [11, 12, 13].

One way of facing the single detection problems and the opportunity losses is the coordination between different secondary users. This cooperation tries to exploit the diversity of different radios.

This means that one radio may be faded or shadowed but it is improbable that all the radios are faded at the same time. As shown in Figure 2.11, cooperation can reuse more spectrum bands so that more channels are reused. The improvement is larger when the number of users under cooperation M is larger [7].

Although the probability of having multipath in all the users is low, shadowing may happen at the same time in several users (rain etc.). This fading correlation between users increases the spatial overhead.

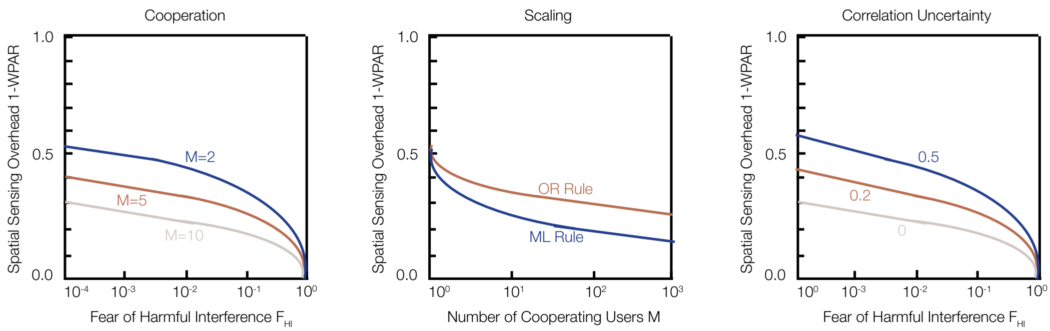

Figure 2.11 Single radio detector limits [7]

Another way to avoid single detection problems is to use a geo-location database system to locate primary users and calculate the white spaces.

2.4 DATABASE SYSTEM

Geo-location databases are an alternative to sensing techniques in CR [14, 15]. They offer a practical solution to monitor capabilities and primary transmissions. In fact, they have been set as the primary tool by the FCC.

A secondary user has access to a database system to get information about the primary users in a given area. The secondary users interact with the database to know which frequencies are free in a certain time and in a certain location and determine the transmission power they are allowed to use. To offer the white space distribution over the frequency bands, the geo-location database needs some parameters and information such as primary user's frequency of operation, transmitted power, size and type of transmit antenna etc. These parameters are provided to the database system by primary users.

Although the database systems solve the problems of location uncertainties, fading effects and detection errors of sensing techniques, they have the tradeoff of synchronizing and updating the database appropriately.

A more precise reuse of the spectrum is possible thanks to database systems and the US FCC has presented it as the main tool to detect white spaces. In Europe on the other hand, a use of both a database system and sensing techniques has been proposed. To analyze the operation of a database system in more detail, the specifications of different regulatory commissions are commented below [16-29].

2.4.1 OFCOM

In November 2009, the Ofcom (http://www.ofcom.org.uk/) published a consultation about the implementation of geo-location databases in order to optimize the efficiency and flexibility of new wireless technologies [16, 17]. Commercial devices and implementations are not expected until 2014 and WSDs are not supposed to replace regular broadband communications but to supplement short-range communications such as home routers, rural broadband communications etc. From 2014 onwards, Ofcom expects the emergence of WSD following a classic S-shape penetration curve.

A geolocation database will be used to inform WSDs about the available channels. The procedure is the following: Ofcom will provide a "master" device with a list of different databases, the secondary user will choose a database and send its location and communication parameters. Finally, the database will answer to the WSD with a list of available channels and the characterization parameters of each channel. To calculate the vacant channels, the database should be aware of the primary licensed usage of the spectrum and should make use of algorithms specified by Ofcom. The master WSD will provide the available white space list to the required slave devices. The master device could be the wireless router in a home and the slave devices could be devices such as laptops connected to the router (the so called "SuperWifi").

The database list on the Ofcom website will be formed by commercial entities (they should apply Ofcom to get permission to publish their database on the website).

In the "Digital dividend: cognitive access. Statement on licence-exempting cognitive WSDs using interleaved spectrum" published on the 1st of July 2009 [18], Ofcom concluded that there are three possible mechanisms to determine the vacant TV channels. The three methods are sensing, geolocation and beacon transmission as mentioned in previous sections.

The beacon transmission was discarded because of the necessary expensive infrastructure and spectrum inefficiency due to the beacon's unexpected propagation. Since the sensing technique does not need any infrastructure and the geolocation resolves the cost and impossibility of sensing low power signals, Ofcom considered enabling the implementation of both mechanisms.

Five key issues must be considered when implementing a database system:

· <u>The information to be provided by the device to the database(s):</u> Ofcom suggested that this can be decided by the device (just location, device type, preferences etc.).

· <u>The information returned from the database(s) to the device:</u> a list of free channels and allowed transmit power levels for each geographical location.

· <u>The frequency of update of the database(s) and hence the periodicity with which devices will need to re-consult:</u> it was suggested that the devices should consult the databases every two hours.

· The modelling algorithms and device parameters to be used to populate the database(s): there are some recommendations depending on the transmission model, device sensitivity, methodology etc.

· The maintenance of the database(s): who and on what terms should be responsible for the maintenance of the databases.

Operation

As mentioned, Ofcom will provide a list of available databases on a website. Licensing information of the Digital Terrestrial Television (DTT) coverage plan should be included in the databases in terms of predicted signal level for each 100m x 100m (pixeled database as it will be discussed later) in the UK. Other parameters will be needed to adapt to the pixel location probability, time variation probability, use of non-ideal antennas etc.

Database providers will have access to this DTT information and they will be informed about any modification in the DTT coverage parameters. In order to enable database providers to have real-time information about the Programme Making and Special Events (PMSE) usage, a link between the database providers and PMSE licensing data is necessary.

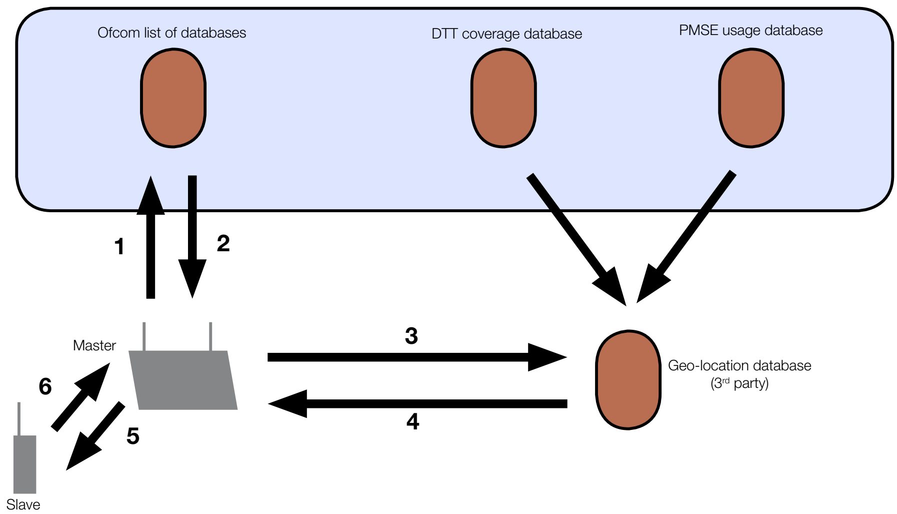

Figure 2.12 Ofcom database operation architecture

As shown in Figure 2.12, the database system works as follows:

1. The master device connects to the Ofcom website.

2. The Ofcom website offers to the master devices a list with registered databases (they are valid within 24 hours).

3. The master device selects a geo-location database from the list and sends some parameters such as location (in terms of latitude-longitude), accuracy of that location to a 95% certainty (in meters), model type (in terms of model number and manufacturer) and height above the ground level.

4. The database answers with the vacant frequency bands represented by the following parameters: start and stop frequencies for the channel, allowed power level, channel validity time, if additional sensing is needed (in affirmative case the sensing level in dBm and type of primary usage). The database gets this information from the licensing information (DTT coverage plan, PMSE usage).

5. The master device can provide the available frequency information to slave devices.

6. The slave devices answer to the master device with data confirmation.

Some problems can appear because of the distance between master and slave devices. Since the slave devices are separated from the master device a certain distance, they could be closer to primary users and cause interference. To avoid it, the location uncertainty of the WSD is increased. However, in some cases the solution is more complex.

Regulatories to be exempt of license

After some discussions, it was decided that it was more convenient to WSDs to be exempt of license. In order to get the legal framework information for their operation, they should consult a Statutory Instrument (SI).

The master devices must follow the following regulatory in order to be exempt of license:

· Determine its location with an accuracy of 95% certainty. The accuracy represents the maximum area within a slave device could be located.

· Consult a list of databases provided by the Ofcom and select one of them (after 24hours the list has to be consulted again).

· Send the location, location accuracy, model and height above the ground level to the database.

· Receive from the database the frequency bands, allowed power level, space and time validity.

· Operate according to these parameters, stopping its transmission when the validity time expires or it moves out from the valid area.

· Share the available spectrum as evenly as possible between the competitor users.

· Provision the out of band performance indicating the emitted power to adjacent channels.

· Manage the slave devices appropriately sending the required parameters.

· Keep control of the slave devices and stop their transmission when the master device has to stop transmitting.

Obtaining a Database listing

It is thought that it is more convenient to the industry to develop the geo-location databases because the Ofcom has no enough Information Technology (IT) resources to supply all the necessary data processing. The commercial entities interested in running a database must apply for a listing in the website to Ofcom. However, the geo-location databases should respect some requirements in order to ensure the proper operation of primary users:

· It has to have access to real-time DTT and PMSE licensing information.

· It should use propagation algorithms and interference parameters approved by Ofcom.

· It should be able to answer to master devices with the minimum parameters mentioned before (frequencies, power level, time validity etc.).

· It must provide a response within 10 seconds.

· It should provide the minimum parameters equally to all devices (some special parameters can be provided to special devices).

· It must modify the algorithms or certain parameters within a week after the Ofcom notification. When interference is severe, Ofcom can order to erase part of the database and it has to be done within one hour.

Database providers are not responsible for interference caused because of DTT and PMSE database errors, inaccuracy of given algorithms and WSD messages. However, they are responsible when the interference happens as a consequence of incorrect implementation of the algorithms, failure when updating the database or changing the algorithms.

An harmonization process is taking place in Europe and Ofcom is participating in the European Conference of Postal and Telecommunications Administrations (CEPT) working group SE43. However, the discussions are in a early steps so the final regulatory and measures are not clear yet.

2.4.2 FCC

In September 23rd 2010, the FCC (http://www.fcc.gov/) allowed the use of TVWSs for "Super Wifi" technologies. The new was taken with expectation since it is the first significant spectrum modification for unlicensed use in 20 years [19].

According to the Second Memorandum Opinion and Order (Second MO&O) [20], additional sensing technologies to database access systems are not required any more and wireless microphone users must certify that they will use all available channels between 7 and 51 before requesting registration in TV bands, being these registrations public.

In order to ensure the protection of primary users, several measures are taken by the FCC. Two free Ultra High Frequency (UHF) channels are reserved for wireless microphones and other secondary users in all areas of the country and a reasonable distance is maintained between secondary users and TV WSDs.

FCC states that all fixed devices have to register their location in a geo-location database. Database administrators are allowed to charge fees for registering. Furthermore, all the devices must include adaptable power control in order to transmit with the minimum power level possible. To avoid interference and ensure the proper operation of the devices, the equipments must be certified by the FCC Laboratory and it has the right to remove any harmful equipment from the market. A TV database is required to contain two main type of data:

· Licensed services operating in the TV bands.

· Location of registered wireless unlicensed microphones that work regularly.

Database administrator assignment

In January 26th 2011, FCC has published an order related to the assignation of TV white space database administrators [21]. In the order, nine entities (Comsearch, Frequency Finder Inc., Google Inc., KB Enterprises LLC and LS Telcom, Key Bridge Global LLC, Neustar Inc., Spectrum Bridge Inc., Telcordia Technologies, and WSdb LLC) are assigned as administrators of TVWS geo-location databases. These administrators are the responsibles of developing the database systems required to identify the vacant channels in a certain location and time.

Before selecting the database administrators, the Office of Engineering and Technology (OET) requested the interested entities to address some details about how they were going to regulate the database. They had to send a proposal including the following information:

· The expertise of the entity in administrating a TV geo-location database and its plan for a five-year period.

· The intended database performance scope and how they were going to synchronize data from different databases.

· Diagrams of the database system and description and interaction between different blocks or functions.

· Information about other entities implementing geo-location databases and the business relationship with these entities.

· Procedures that will be used to communicate WSDs with the database system, procedures to verify that a device can communicate properly with the database and security measures to avoid access and modification of the database from unauthorized users.

FCC received 9 proposals in answer to the request. Although the multiple database administrator decision implies some coordination issues, it is for public interest to have several entities developing business models for a new technology. FCC is also considering extending the database system operation model to other frequencies, being important to find potential frequency bands.

The nine parties have been accepted as database administrators under the condition that they have to modify their proposals according to the specification modifications and they have to work closely and in coordination with the agency. In order to ensure the correct implementation and development of the database systems, significant testing and oversight will be needed. In fact, the OET has organized some mandatory workshops [22, 23, 24] (some of them already took place) and determined some milestones to submit reports to be sure of the consistency and compliance with the rules. Before actual implementation of the database system in the TV bands, a real testing will be required (not less than 45 days).

Once a database administrator passes the testing process successfully and it is approved by OET, the OET will announce public availability of the database for a period of 5 years.

2.4.3 COGEU

COGEU (http://www.ict-cogeu.eu/) is a project that analyzes COGnitive radio systems for efficient sharing of TV white spaces in EUropean context. It tries to investigate the technology, business and regulatory/policy domains to take advantage of the digital switchover and develop the cognitive access to TVWSs in Europe. It also defines new methodologies for TVWS equipments coexisting at the same time with the European DBV-T/H standard [25, 26, 27, 28].

According to the COGEU D2.1 paper [25], local sensing and geo-location database systems will be combined in order to compute the maximum transmit power and relax sensing constraints, increasing potenal of TVWS market. The key parameters given by the regulatory bodies are the following:

- Location accuracy: nominally 100 metres.
- Transmit power: as specified by the database.
- Transmit-power control: required.
- Bandwidth: unlimited.
- Out-of-band performance: less than 46 dBm.

COGEU considers a centralized model with a broker and some players. The spectrum broker decides the amount of spectrum bandwidth assigned to each player (Figure 2.13).

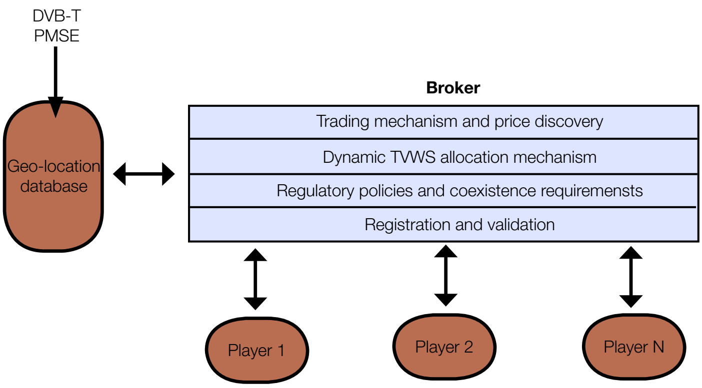

Figure 2.13 COGEU database architecture

Before reaching the conclusion of implementing a combination of sensing and geolocation database, COGEU has compared the sensing and geolocation requirements imposed by different regulators (Figure 2.14).

Parameters	Ofcom	FCC
DTT sensing threshold	-120 dBm	-114 dBm
Time between sensing	< 1sec	<= 1min
Tx adjacent-channels	4 dBm	16 dBm
Tx non-adjacent channels	17 dBm	20 dBm
Database frequency update	<= 2hour	< 1day
Location accuracy	100 meters	50 meters

Figure 2.14 Comparisson between different regulators

Thanks to a database system, part of the complexity related to the sensing is transferred to the database, reducing the computations in WSDs. The main problem of geo-location databases is that they can just protect registered users. In Europe, PMSE works without registering in many cases and in these cases, the sensing is the only method that can protect them. That is why COGEU combines both sensing and geo-location database systems.

According to COGEU, databases must satisfy some requirements.

· The database has to provide a list with the available channels for each pixel (100mx100m) in addition to the maximum transmit power for each channel.

· The database must be available 99.99% of the time (same as Ofcom).

· I should be updated within a period of 24 hours.

· The minimum period within WSDs have to update their information from the database is 2 hours (same as Ofcom).

· To protect PMSE applications, two channels are reserved for microphones in every location.

· A priority system can be adopted by the database: public systems will have the highest priority, followed by incumbent systems and by TVWS systems.

· The database must be able to stop secondary transmission when the regulatory asks.

· The broker can update the database with information coming from the sensing of the players.

The regular frame structure given by COGEU is the following (Figure 2.15):

Figure 2.15 Regular frame structure by COGEU

The topology of the COGEU database system is a two level hierarchy with a central database (CDB) with data from the whole country and local database (LDB) with data from regional WSDs. This topology allows the coexistence of several database systems in the same region. COGEU proposes a first access to the geolocation database using the existing radio interfaces such as WiFi, WiMax or Long Term Evolution (LTE).

Since the Digital Video Broadcasting (DVB) transmitter data are static, the COGEU approach tries to make the calculations in advance for static data to reduce complexity and delays.

Unlike other regulatory policies, COGEU presents a secondary spectrum trading of TVWSs. This means that the regulatory bodies assign certain TVWSs for unlicensed use in some areas. The rest of the TVWSs are assigned in a secondary spectrum market.

WSDs that want to access to a certain spectrum band in a certain region need to register with a database. In order to make sure that geo-location databases are accessed only by accepted users or devices, some basic information of the device (serial number etc.) will be stored during the registration process. When the device accesses the database again to ask for available channels, the serial number will be asked again as a security measure.

2.5 STANDARIZATION

Due to the emergent importance of CR and TVWS, some standards are being developed. There are still many points and issues to determine but the standard teams are working and researching in order to regulate the TVWS access. One of the standards that may have possibilities to be adopted in teh future is the IEEE 802.11af [30].

2.5.1 IEEE 802.22

IEEE 802.22 is a standard for cognitive wireless regional area networks (WRANs) [31, 32, 33]. It is still under development in the IEEE 802 LAN/MAN Standards Committee. The goal of this standard is to use CR techniques to share geographically unused spectrum allocated to the television broadcast service, on a non-interfering basis, to bring broadband access to hard-to-reach low-population-density areas typical of rural environments. IEEE 802.22 WRANs are designed to operate in the TV broadcast bands while ensuring that no harmful interference is caused to the incumbent operation and low-power licensed devices such as wireless microphones.

The white space may consist of unused frequencies or unused fragments of time in a given location. There are several methods that can be used by a CR network to be aware of its spectral environment. The two methods used with IEEE 802.22 for spectral awareness are geo-location/database and spectrum sensing. The 802.22 network quickly modifies its operating frequency so as to only operate on channels unused by licensed transmissions. Thus, the 802.22 network must both quickly identify which channels are allowed to use, and move to a new unused channel if the current operating channel becomes occupied by a licensed transmission.

The application for the IEEE 802.22 WRAN standard will be providing wireless broadband access to a rural area of typically 17–30 km or more in radius (up to a maximum of 100 km) from a base station (BS) and serving up to 255 fixed units of customer premises equipment (CPE) with outdoor directional antennas located at nominally 10 m above ground level, similar to a typical VHF/UHF TV receiving installation.

2.5.2 IEEE SCC41

The activities of IEEE Standardization Coordinating Committee 41 (SCC41), "Dynamic Spectrum Access Networks" aim at facilitating the development of research ideas into standards to use the research results for public use. Now the IEEE SCC41 has become the premier forum for standardizing concepts related to CR [34].

The objective of IEEE SCC41 is to develop standards supporting new technologies for next-generation radio and advanced spectrum management. IEEE SCC41 deals with an area of technological convergence, where radio engineering, wireless networking, computer science, software engineering, network management, and other disciplines contribute to the success of CR deployment.

IEEE SCC41 is formed by several Working Groups (WGs), each of them responsible for drafting a standard for a specific topic described below:

· IEEE 1900.1: Definitions and Concepts for DSA - Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management.

· IEEE 1900.2: Recommended Practice for the Analysis of In-Band and Adjacent Band Interference and Coexistence between Radio Systems

· IEEE 1900.4: Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks

· 1900.4a: Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks - Amendment: Architecture and Interfaces for Dynamic Spectrum Access Networks in White Space Frequency Bands

· 1900.4.1: Interfaces and Protocols Enabling Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Networks

· IEEE 1900.5: Policy Language and Policy Architectures for Managing CR for DSA Applications

· IEEE 1900.6: Spectrum Sensing Interfaces and Data Structures for DSA and Other Advanced Radio Communication Systems.

2.5.3 ETSI RRS

ETSI Reconfigurable Radio Systems Technical Committee (RRS TC) is performing work that is complementary to the IEEE SCC41 and IEEE 802 activities, with a focus on the following: SDR standards beyond the IEEE scope, CR/SDR standards addressing the specific needs of the European Regulatory Framework, and CR/SDR TVWS standards adapted to the digital TV signal characteristics in Europe [34].

ETSI RRS is formed by several WGs:

· WG1: focuses on system aspects and develops proposals from a system aspects point of view for a common framework in RRS TC to ensure coherence among the different RRS TC WGs and to avoid overlapping and gaps between related activities.

· WG2: focuses on SDR technology with a particular interest in radio equipment architecture and proposes common reference architectures for SDR/CR radio related interfaces etc.

· WG3: focuses on cognitive management and control.

· WG4: focuses on public safety and collects and defines the related RRS requirements from relevant stakeholders in the public safety and defense domain.

2.5.4 IETF PAWS

Internet Engineering Task Force (IETF) is developing a WG called Protocol to Access White Space database (PAWS) with the goal of defining the device-database interface for TVWS database systems (https://www.ietf.org/mailman/listinfo/paws).

According to PAWS, a WSD must access a database to get the list of available channels for its location. There could be several databases serving WSDs so that a database to database interface may be useful. However, PAWS initial work is focused in the user/device interface with the database.

Devices may be able to connect to the database directly or indirectly via the Internet or private IP networks. This interface needs to be: radio/air interface agnostic (802.11af, 802.16, 802.22, LTE etc.), spectrum agnostic (the protocol should be able to use in any spectrum band), globally applicable (applicable in any country ensuring uniformity) and it must address regulatory requirements (the interface has to be flexible to adapt to different regulatories).

PAWS pretends to specify both a database identification mechanism (how can a device know what database it has to connect to) and contents of the queries and responses (XML is an option). This messaging between the device and the database needs to be secure (authentication, integrity of the content, prevent from man-in-the-middle attacks etc.), requiring some authentication and security measures.

The WG is still working and developing the initial proposal for the Protocol.

2.6 COMMERCIAL DEVELOPMENTS

Although the TVWS communication systems are still in research period, little by little some commercial developments are appearing. Some companies or entities have already started to implement real geo-location database systems to access and test TVWSs. In this section, some of these commercial systems are commented.

2.6.1 MICROSOFT

Microsoft has created WhiteFiService, a research platform to plan your white space network. It provides APIs that wireless devices can use to determine the available white spaces in a given location (http://whitespaces.msresearch.us/).

The data about primary transmitters (tower location, TV channel, transmit power, antenna height etc.) are provided by the FCC Commision's Consolidated Database System (CDBS) and the data about the terrain from NASA. The attenuation from the transmit tower is calculated following some algorithms. In this way, with the device location as input data, WhiteFiService provides the available channels in that location.

The API is flexible; in fact, the user can determine the propagation model (Longley Rice, Egli, Okumura-Hata and Free Space), the detection threshold, terrain source etc. The results were validated over a distance of 1500 miles in Massachusetts and Washington. They are trying to spread their results to border areas and include TV towers from other countries.

It was created using ASP .NET. The TV tower data was last updated in March 2010 but they are working on an improvement to get a daily update.

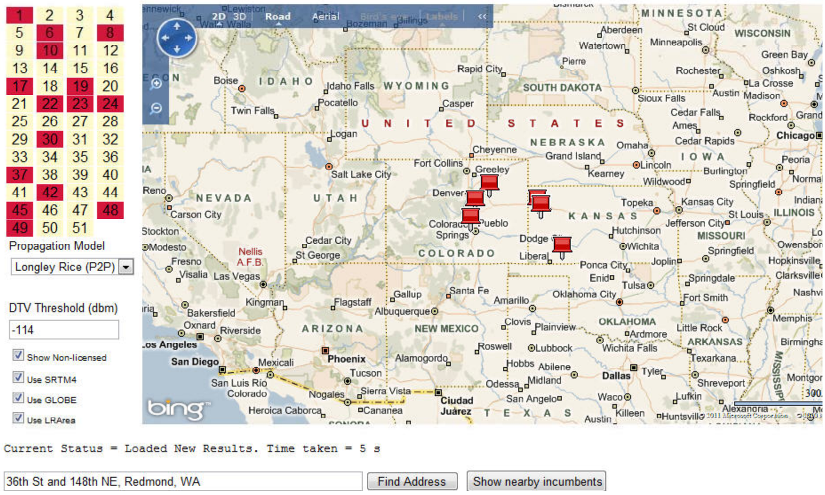

Figure 2.16 Microsoft API example

In Figure 2.16, an example of the WhiteFiService is shown. The boxes on the left side correspond to TV channels. The yellow boxes represent available channels while the red boxes represent occupied channels. The corresponding TV towers appear in the map.

2.6.2 SPECTRUM BRIDGE

Spectrum Bridge is a company that provides wireless networks with the ability of dynamically access the available spectrum via a database system. It tested an experimental White Space Database system for the FCC in January 2009 and has been operating a complete online white space solution since January 2009 (http://www.spectrumbridge.com/WhiteSpacesSolutions/showmywhitespace.aspx).

Spectrum Bridge already has several commercial or at least real-state implementations in collaboration with other companies:

· Logan, Ohio: Google and Spectrum Bridge, Inc., together with the Hocking Valley Community Hospital announced the deployment of the first TVWSs broadband trial network for healthcare providers in Logan, Ohio. This network deployment garnered Spectrum Bridge the RCR Wireless Ecosystem Award for the best specialty deployment in 2010.

· Plumas-Sierra County: Spectrum Bridge, in partnership with Plumas-Sierra Electric Cooperative and Google, launched the nation's first "SmartGrid" wireless network trial, utilizing TVWSs spectrum.

· Wilmington, NC: Spectrum Bridge's "Smart City" solution is able to access available TVWSs spectrum and manage the deployed "Smart City" network by dynamically assigning non-interfering frequencies to WSDs throughout the service area to effectively provide bandwidth where and when it is needed.

· Claudville, VA: In partnership with Dell and Microsoft, Spectrum Bridge launched the nation's first TVWSs broadband network.

· <u>Cambridge, England:</u> Spectrum Bridge, in conjunction with OFCOM, is launching a TVWS network in Cambridge, England.

In Figure 2.17 and Figure 2.18, examples of the Spectrum Bridge TVWS finder are shown.

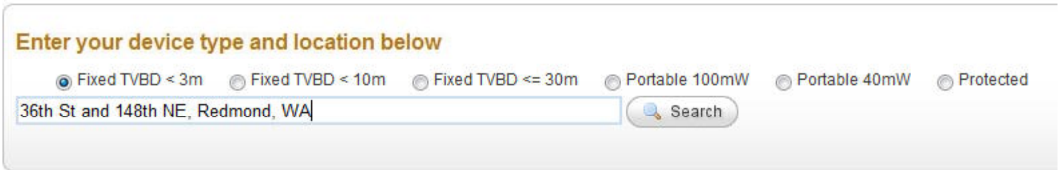

Figure 2.17 Spectrum Bridge API example

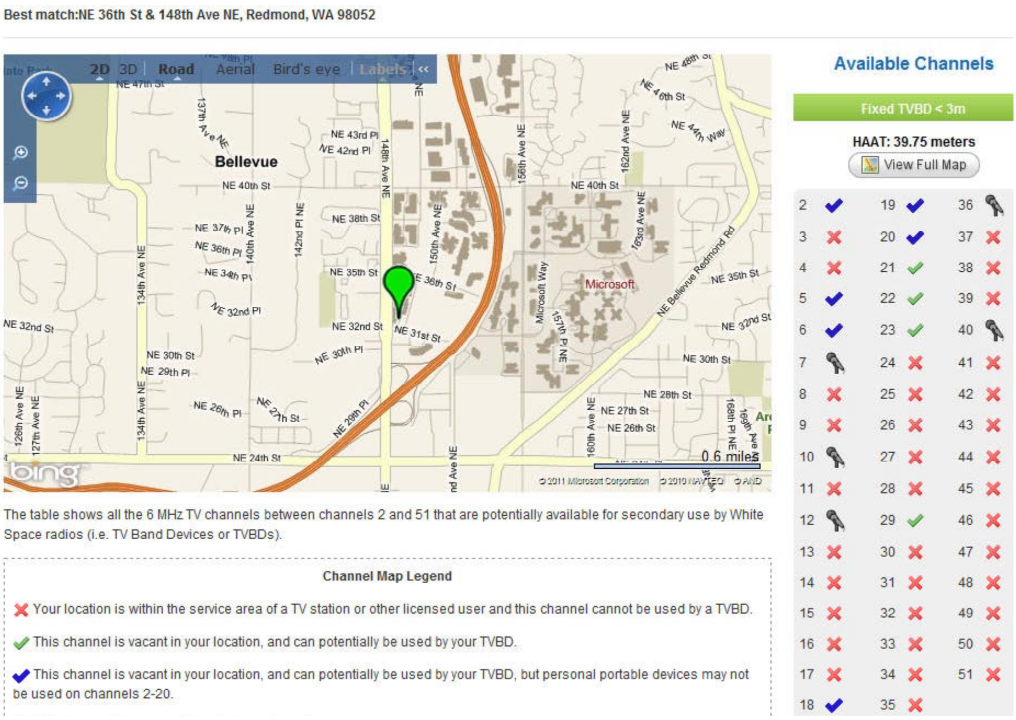

Figure 2.18 Spectrum Bridge API example II

3. THEORETICAL STUDY

General background information on CR, TVWS communications and geo-location database systems was introduced in the previous section. However, to perform and implement a database system properly, a more theoretical study is required.

This section covers the additional theoretical study part that could be interesting when developing a geo-location database system.

Since in the USA several entities or companies are currently developing a database system, being their activities regulated by the FCC and guided through various official workshops [22-24], the analysis of these database administrator entity proposals might be helpful for a correct implementation of the geo-location database system. Although there are nine (maybe ten with the late incorporation of Microsoft) database administrator entities and the proposals of all have been analyzed [36-44], just three (Google, Spectrum Bridge and KB Enterprises LLC and LS telcom) are exposed in Section 3 due to their similarities.

3.1 PROPOSALS

In January 26th 2011, FCC published an order related to the assignation of TV white space database administrators [21]. In the order, nine entities (Comsearch, Frequency Finder Inc., Google Inc., KB Enterprises LLC and LS Telcom, Key Bridge Global LLC, Neustar Inc., Spectrum Bridge Inc., Telcordia Technologies, and WSdb LLC) are assigned as administrators of TVWS geo-location databases. These administrators are the responsible of developing the database systems required to identify the vacant channels in a certain location and time. From March 2011, the 9 database administrators are attending to different workshops to develop their database system proposal.

3.1.1 GOOGLE

In January 4th 2010, Google presented its proposal to the FCC addressing the details requested by the same [38].

Expertise in administrating a TV geolocation database

Google states that as a member of the White Spaces Coallition and a founder of the White Spaces Database Group, it has been a supporter of the unlicensed efficient use of the white spaces from the beginning. Its work with coallition members, comission staff and in the development of testing systems gives to Google the necessary expertise and experience to be a database administrator. Google has many engineers working on the Internet software and database design.

Google proposes a flexible and market-driven approach, without limitation in the number of database providers and database architecture specification, considering an open-architecture model. In particular, they propose a clearinghouse model.

Business plan

Regarding to the five-years business plan, Google has access to enough funds and capitals to develop and maintain the proposed database systems for five years. In case that they do not run the database system for the five-years period, they will transfer the database, IP and URL addresses and the list of the registered fixed WSDs to another entity.

Although the FCC allows charging fees for registering, Google is not planning to do it according to its philosophy of maintaining user costs as low as possible.

Database performance scope and architecture

Google says that it can provide an end-to-end database system with the basic functions of data repository, registration, and channel availability request/answer. This architecture is represented in Figure 3.1.

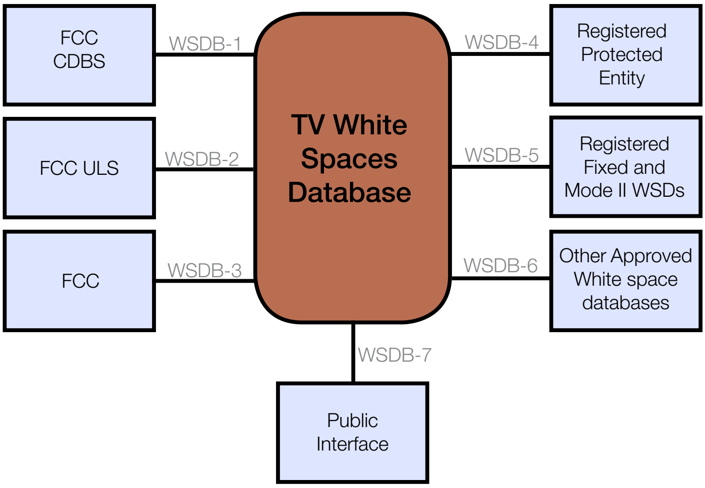

Figure 3.1 Google database strucutre

Managing data repository

The database will include three types of information: location and channel of protected facilities recorded in the FCC CDBS and ULS databases (DTV stations, analog class A stations, low power TV LPTV stations, TV translator and booster stations, broadcast auxiliary service BAS stations, private land mobile radio service PLMRS stations, commercial mobile radio service CMRS stations, offshore radiotelephone service ORS stations), location and channels of protected facilities not recorded in the FCC databases (cable TV headends, TV translator receive sites, licensed wireless microphones, other low power auxiliary stations), information about location and identification of secondary WSDs (geographic location, FCC ID serial number, responsible person information).

In order to interact with these three information types, the database will have three interfaces with FCC based on IP. WSDB-1 and WSDB-2 to directly connect the database to CDBS (updated weekly by FCC) and to ULS (updated daily by FCC). WSDB-3 will be use for other purpose communications with the FCC.

Query

In order to get the available channel list, the WSD will connect the database through WSDB-5 to register in the database. The database will calculate the available channels and return a list with them to the WSD. Interference is avoided by protecting the TV stations using transmitter parameters from CDBS and FCC contour data (http://www.fcc.gov/ftp/Bureaus/MB/Databases/tv_service_contour_data/). The Database will also check for WSDs in the borders with Mexico and Canada where there may be prohibited operations.

Google thinks that anyone should be able to have access to the available TV channel list. The interface WSDB-7 will offer public access to non-confidential information from the Database.

Synchronization with other DB

Google states that it will cooperate with other administrators to create a process to provide to each other everyday information about registered cable TV headends, TV translator station receive sites, wireless microphones and other low power auxiliary stations, and fixed WSDs. This process will take place through a common interface called WSDB-6 in Figure 3.1.

To avoid that the obligation to cooperation among different databases causes delay in database operations, Google proposes that a database operator should require a minimum synchronization interface before accepting any protected entity registrations and beginning operations.

Protection against unauthorized access and security measures

The main security purpose of the database system is to ensure that the WSDs receive information from the validated database administrator and that no-one is trying to impersonate it sending invalid information. In consequence, the interfaces WSDB-4 and WSDB-5 that connect the database with the WSDs using the Internet should be protected using different measures:

· Public key infrastructure (PKI): is the same technology used on the Internet to perform transactions where the database provides to the WSD a certification certified by a well-known certificate authority.

· Transport security: Since the interfaces can be defined as HTTP interfaces carrying XML contents, the transport layer security can be performed in the same way as in web pages providing authentication, integrity and confidentiality (TLS, SSL).

· Device management: to register in the Database: the WSD will authenticate itself by using a shared secret determined when enrollment takes place.

· RPE management: Before registration, the identity of the RPE will be verified and the contact information will be confirmed issuing a shared secret.

· Other security measures: Data replication across multiple secure locations to ensure uninterrupted access in the event of natural disaster, accident, or attack, Intrusion detection and monitoring systems, Anti-malware detection software on file servers and personal computers, Denial of Service protections etc.

3.1.2 SPECTRUM BRIDGE

Expertise in administrating a TV geo-location database

Spectrum Bridge is a company with sufficient support and technical expertise to develop a wireless database system. They created the first cellular billing clearinghouse, created the first online exchange for secondary market spectrum and had pioneering leadership roles in multiple telecommunications technologies from soft switches to ad-hoc networks [42].

It developed a trial version in January 2009 and demonstrated a complete online white space version in June 2009 setting the first white space network in Virginia in September 2009. Its success has made them to start deploying additional white space networks.

Business plan

It is predicted that the white space business will be a multi-billion dollar business market by 2014, providing wireless communications to last and middle mile broadband access.

Spectrum Bridge proposes a two fee component business model: a per-unit device fee and a value added service fee. The first one will be kept as low as possible to encourage secondary users to use TVWSs. The device fee will be paid the first time the WSD registers with the database. The second ones will be fees for additional services to the minimum ones.

Database performance scope and architecture

The proposed architecture and interfaces by Spectrum Bridge in Figure 3.2 will be updated to support FCC requirement changes.

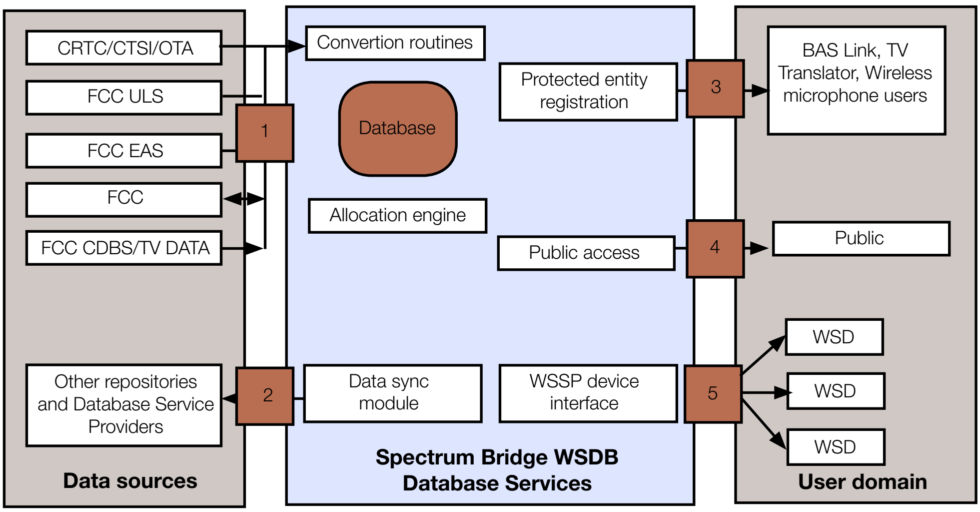

Figure 3.2 Spectrum Bridge database structure

Data sources

Necessary data to calculate the white spaces in a given location and time (transmitter location, channel number, call sign, ERP, antenna pattern etc.). The information is provided by the following primary data sources: FCC contour files (Digital television stations, Class A, FCC Consolidated Database System, FCC ULS database, FCC EAS database, 47 CFR Part 15 Subpart H – Television Band Devices).

Conversion routines

Routines that translate the information from the data sources to the contours stored in the database.

Protected entity registration

Provides the protected access for some designated entities (microphones, cable headends, other BAS links, TV translators) by storing an exclusion zone in the database.

Public access

It is a web based interface (www.showmywhitespace.com) that provides public access to the available white space information for a given location. The database system will use the calculations or algorithms specified by FCC at any location for each WSD type.

WSDB device interface

It is the electronic Internet based interface for WSDs to register in the database and query the database for free TVWSs.

Data sync module

It is used to share information about protected entities with other databases or the FCC. Since there will be different databases and each one will register unique data, it is convenient to share the registered information among authorized databases, data repositories and regulatory entities. The SBI solution offers a root certificate authority with respect other authorized databases using a PKI. An individual certificate for each database can also be provided to help to establish a SSH. In addition, Simple Object Access Protocol (SOAP) will be used to share data among databases in real-time.

The TV transmitters' data from the FCC is already fixed and it changes slowly, not being necessary to share these data in real-time. Estimated data storage requirements for all registered information does not exceed 500 MB, being adequate to use a secure file transfer protocol (FTP over SSH).

Protection against unauthorized access and security measures

Server architecture and maintenance

In order to ensure the functionality of the database system when some physical errors occur the Spectrum Bridge Data Center is designed with remote offsite secure facility, redundant connectivity, controlled environment, monitoring, managed security, hot standby, automated data backup and restore etc.

The initial database requirement is estimated in 500 GB. The additional data related to the service is less than 100 GB.

Data protection and privacy

Spectrum Bridge proposal offers itself to provide all the functions and services required by the FCC, including the core database, channel calculations, WSD and FCC interfaces, support to protected entities etc.

3.1.3 KB ENTERPRISES LLC AND LS TELCOM

Expertise in administrating a TV geolocation database and business plan

KB Enterprises LLC ("KBE") is a consultancy providing telecommunications policy consulting as well as spectrum auction design and software implementation.

Their experience in telecommunications began within the FCC and more recently they have worked for wireless telecommunications companies and government regulators around the world providing software solutions and tools for companies and regulators for spectrum assignment.

LS telcom AG (LS telcom) is the world's leading vendor of spectrum management solutions supporting more than 75 countries worldwide. Since it was founded in 1992, LS telcom has provided network planning software to operators and spectrum management solutions to national regulators.

KB/LS has the required expertise to work as a TVWS database manager, having all the necessary elements, developers and media.

KBE/LS will provide all the necessary elements to build the database system and it is planning to charge an initial fee to devices for registering with the database. This fee payment will be implemented in a non-discriminatory way offering flexible payment options [39].

Database performance scope and architecture

Managing data repository

The primary data will be imported and updated from FCC and ULS databases. Licensed devices that are non-registered at the FCC will also be able to register with the database system. Secondary wireless users will also have to register with the database when asking for available white spaces. In order to control the user access, basic information about the equipment (serial number etc.) will be stored in the registration process and it will be used for verification.

Synchronization with other DB

To enable the proper operation of the system, communication between different databases will be synchronized and determined.

Protection against unauthorized access and security measures

The information and access to the database system will be protected in different ways. There will be several firewalls, the communication between the interfaces and the database will take place via web services to avoid direct exposure of the data to the Internet.

The database system is built in a redundant solution to ensure permanent availability of the system, using Oracle Real Application Cluster (RAC) technology. The web servers are implemented using load-balancing technology and the hardware has redundant power supplies and Hot-Plug Raid solutions.

3.1.4 CONCLUSIONS

After analyzing the proposals of the nine selected database administrators, we can conclude that all the proposals are basically similar. The database architecture is almost the same, requiring some data sources, synchronization between different DBs, public access and registration from secondary users and some security measures to ensure the proper operation of the system.

Regarding the business plan, the entities are offered to supply all the necessary elements to build the database system. However, there are some differences in the fee charge. Some entities want to try not to charge fees while others propose a business plan based on several fees.

These are just the initial proposals, so some of them are not very detailed. They indicate the initial scopes and architecture that the database system should have. In March 2011, some workshops started and the proposals will update and develop to adapt to the specifications exposed by the FCC. Although the proposals do not specify many important aspects in detail they are a good starting point and reference when developing an initial database design.

4. IMPLEMENTATION OF TVWS DATABASE SYSTEM

The main part of the thesis is to implement a TVWS database system, what is covered in this section. For that, in Section 4.1 an initial database system based on FCC administrators' proposals has been designed with some extra considerations and characteristics.

The relational database model has been selected for the implementation. In consequence, in Section 4.2 the metastrucuture of the information is determined and all the components and attributes of the same are defined.

In Section 4.3, a physical database is created and some sample data is introduced in the same to make simulations and populate the database system demo. Note that this population of the database should in theory be done automatically through interfaces with different entities (defined in Section 4.1.1). However, the lack of real interfaces and regulatories to control and regulate the interaction with data provider entities makes necessary the manual introduction of sample data. The sample data is intended to be as realistic as possible so that the implementation results are quite approximate to real ones. However, they will not exactly be the same, having to keep it in mind during the development of the system.

In Sections 4.4 and 4.5, the code description and development are exposed (Java programming language is used for that). Finally, Section 4.6 covers the creation of a user interface (web interface) to interact with the geo-location database system.

Par of the code used to create the implementation can be found in Appendix A (MySQL code), Appendix B (Java code) and Appendix C (web interface code).

4.1 INITIAL DATABASE DESIGN

4.1.1 ARCHITECTURE AND INTERFACES

The initial architecture or structure of the database system is shown in Figure 4.1 and it is similar to the architectures proposed by the FCC database administrators.

Architecture

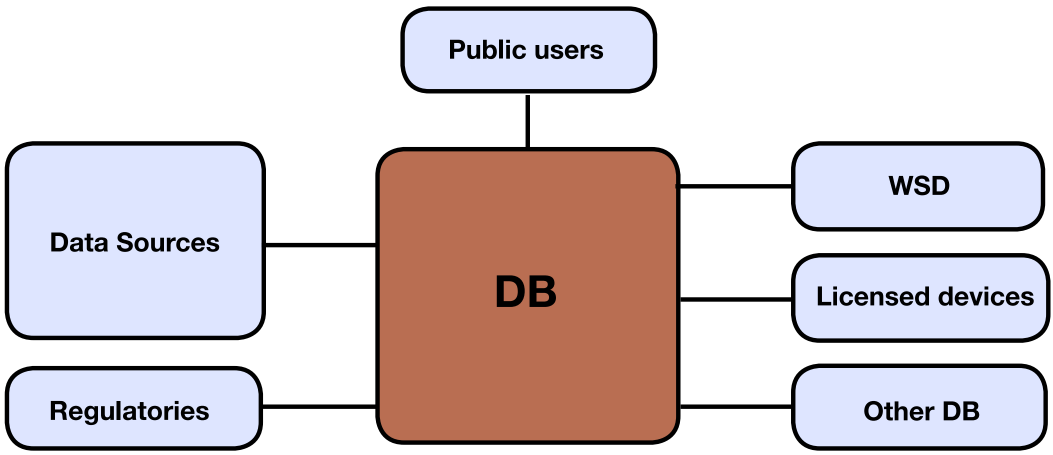

Figure 4.1 Database architecture

DB

The database stores and manipulates the data required to determine the free channels in a location and instant of time. Some calculations need to be done to obtain the available white spaces, and some data is necessary for these calculations. The operation of the database will be explained in more detail later.

Data Sources

The Data Sources provide the required information about primary licensed users that needs to be stored in the database. In theory it should be provided by official or regulated database sources (ULS etc. in the US). Since the database system developed in this thesis is a demo and there are not yet official data sources in Europe, the required data will be introduce to the database manually and it will be sample data. In the future, real Norwegian TV broadcasters' data is supposed to be introduced, but to start with, some modelled information will be inserted.

Regulatories

In the same way as the FCC in the USA, some regulatory entity or policy (ECC i.e.) should control the proper operation of the database system. It should ensure that the database system complies with the established measures and levels and does not provoke interference with primary users. However, this regulatory has not been set in Europe and in the demo there will not exist this communication with a regulatory, the regulatory data will be statically introduced (in the same wa as the rest of the data).

Other DB

In order to improve the performance of the system and make easier the registration of not registered licensed users, some cooperation and information sharing between different databases will be convenient. Once more, we are going to develop a demo so it will not be possible to synchronize with other databases. Nevertheless, it is good to contemplate it.

Licensed devices

Not all the licensed devices are registered in the official data sources. In consequence, theoretically, these non-registered licensed users should access the database to register. In the demo, the information about the non-officially registered licensed users will be introduced manually.

WSD

The WSDs that want to get access to an available free channel will have to first register and identificate with the database and then query for the available free spaces.

Public users

Public users will also be able to access the database to get information about the available white spaces in a given location and time, even if they are not going to make use of them.

Because we do not have real data, actual regulatory entities and WSDs do not exist, the demo will be focused in the implementation of the public access to the database. So from now on we are going to consider that all the necessary data is already stored in the database and that all the active WSDs are registered with the same.

Interfaces

To enable the access, update and exchange of the data mentioned above, 6 interfaces are defined in the database system, they are shown in Figure 4.2. These interfaces will communicate different blocks that form the database architecture with each other.

Interface1

Interface 1 will be used to get and update information about licensed users from the data sources. The update time of this information should be specified by a regulatory entity. Every period of this time, the data sources will send to the database through the Interface 1 the information related to licensed transmit towers (the required information during this communication will be specified later: location of the tower, power, channel etc.). However, the database will be able to ask the data sources for the information any time (exceptions or alarm situations could happen).

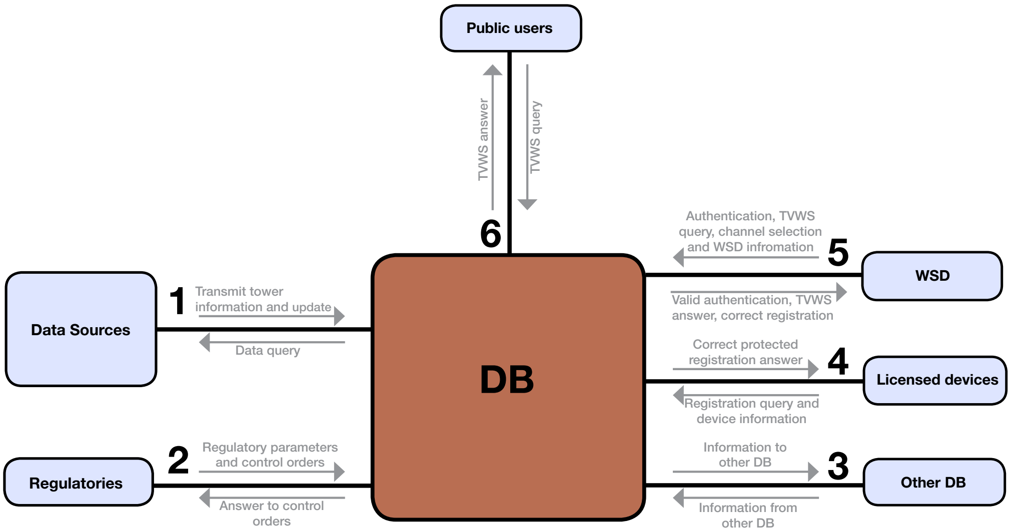

Figure 4.2 Database interfaces

Interface2

Interface 2 will be used to establish the communication between the database and the regulatory. First of all, a regulatory should establish some parameters such as the propagation model to be used, white space decision threshold, validity time etc. to control the operation of the database system. On the other hand, from time to time the regulatory entity will be able to ask for some information about its operation to the database to control if it is respecting the requirements and it is not producing interference to other users. In case that the database system is not respecting some parameters, the regulatory will ask to stop or repair the operation of the problematic parts.

Interface3

The synchronization with other databases will be done using Interface 3. In a scenario with several database systems, it is convenient and positive to share information about registered users between them. To facilitate the information sharing (information about not primary licensed users, WSDs etc.) Interface 3 will be created.

Interface4

Interface 4 enables the protected registration of secondary licensed users. We mentioned before that some licensed users are not registered in the data sources or official databases. In consequence, these users will have to access the database and send information about their location, transmit power, user type etc. through Interface 4.

Interface5

To authenticate WSDs as valid devices, register them with the database, query for available channels and receive the answer, Interface 5 will be used. First, the WSD will have to communicate with the database and authenticate itself as valid device (public key and others). Then, the WSD will ask for available channels and for that it will send its location to the database. The database will respond to the WSD with a list of available channels for the input location. The WSD will select a channel for its operation (or it will decide not to select any channel) and inform the database about its selection sending the number of the selected channel and information about itself (device type and number, transmit power etc.). The database will register the WSD (if necessary) and will indicate that the selected channel is used by this WSD. Notice that the WSD occupation or usage registration is not required in all regulatories (in FCC i.e.). However, in order to keep the database design as flexible as possible to different regulatories and new applications, this option will be available.

Interface6

Public users will access a web interface, Interface 6 in the figure. Public users will be able to apply for available channels in a given location thanks to interface 6. They will send the location as input data and the database will answer with a list of available channels.

4.1.2 OPERATION AND STRUCTURE OF THE DATABASE

We can distinguish between two different types of database systems: pixeled DB or device DB. Both cases will be analyzed in order to choose the most convenient one.

Device DB

In a device DB, the WSD indicates its position to the database system and the database system calculates the available channels for the exact input location of the WSD in real-time. This implies to have to make all the computations in real-time, needing more processing and some delay. However, the structure of the database system is very simple and the results are precise. In Figure 4.3 the operation of a device DB is shown.

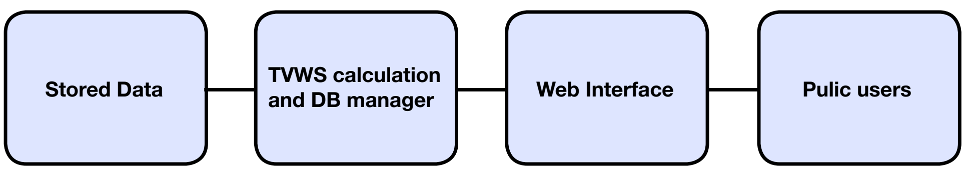

Figure 4.3 Device DB structure

Stored data

As mentioned, some information about the licensed users and WSDs will be registered in a database. The database will contain information about the licensed transmit towers, secondary licensed users and registered WSDs.

TVWS calculation and DB manager

This block of the database system will access and manage the database to get the necessary information to calculate the available free channels for a given time and location indicated by the public user. Once the required data is available, some algorithms will be use to calculate the white spaces.

First of all, a path attenuation relative to the location introduced by the public user will be applied to the transmit power. There are different propagation models: Free Space, Rice, Hata, Egli etc. Once the power that reaches the input location is known, the easiest or simplest way to determine the available white spaces is to use a threshold. A white space decision threshold will be stored (usually determined by the regulatory entity) and if the power in the location is lower than this threshold, the channel will be considered to be free. Otherwise, when the power is higher than the threshold, the channel will be classified as occupied.

Web interface

In order to offer public access to the database, a web interface will be created. It will be a web application that allows introducing the location (address or coordinates) of the user. The application will return to the user a list with real-time calculations of the available channels for that exact location.

Operation

The system will work as follows: a user accesses to the web interface and introduces a location and the optional parameters. The introduced parameters are sent to the DB manager. The DB manager will access the database and get the corresponding data about the transmitters in the input location. The TVWS calculation part will use the data from the database and the input optional propagation parameters to calculate the available free channels for the input location. The resulting list will be provided to the user through the web interface.

Pixeled DB

In a pixeled DB, the map or geographical area is divided into squares or pixels. From the information about transmitters stored in the database, a list with the available channels will be calculated and stored for each pixel in advance. In this way, real-time computation is reduced considerably at the expense of requiring more storage data and loosing some precission.

Since the Ofcom [17], COGEU [27] and ECC [29] propose pixeled DBs, the same database configuration will be used during the thesis. Even though FCC proposes a pixel size of 50m x 50m we are going to take bigger squares (100m x 100m like in Ofcom) to reduce the storage data and make the implementation simpler.

In Figure 4.4 the operation of a pixeled DB is shown. It is the same as the device DB except for that the results are pre-calculated or each pixel.

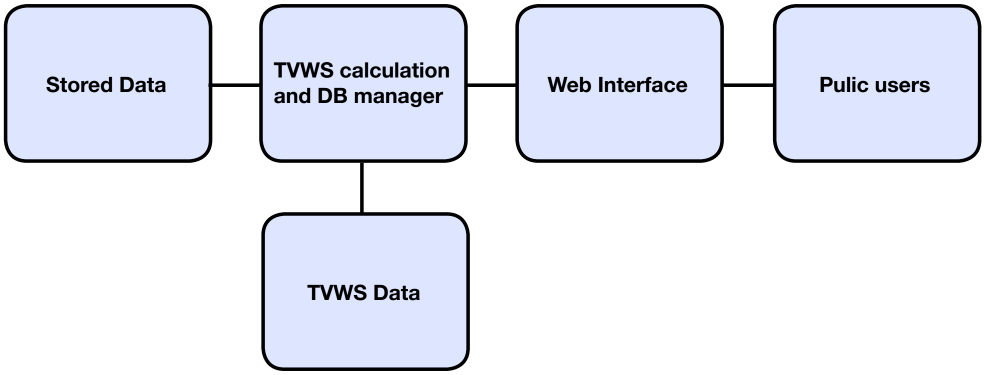

Figure 4.4 Pixeled DB structure

Stored data

In the same way as in the device DB, the stored data corresponds to the necessary data to calculate the available channels in a location. The stored data can be classified in four types: licensed TV tower or primary user data, other licensed secondary user data, WSD data and some regulatory data.

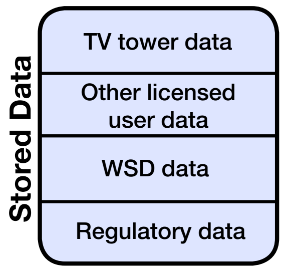

Figure 4.5 Stored data structure

· TV tower data: It will be data related to the licensed primary TV towers or other primary users. The stored data related to primary licensed users will be the following:

- Tower ID (identifies the transmit tower).
- Transmitter type (to indicate if the transmitter is a TV tower or another type).
- Tower location (longitude and latitude in order to know where the transmitter is po sitioned and calculate the propagation loss to the destination).
- Transmitting TV channel (indicates the channel(s) in which the TV tower is transmiting).
- Transmit power or ERP (the power transmitted from the tower, it is used to calcu-late the power that reaches the destination).
- Antenna height or AHHT (height of the tower above the terrain, it is needed to cal culate the propagation loss).
- Antenna pattern (the radiation pattern of the transmit antenna).
- In some proposals or implementations other parameters such as protection rules in incumbent systems, the population and area covered etc. are mentioned.

· <u>Other licensed user data:</u> As explained, apart from the official transmitters, other licensed users such as wireless microphones will exist. These devices should be registered in the database in order to ensure the proper operation of the system. The following parameters will be stored:

- User type (to indicate if it is a wireless microphone or other type of secondary licensed user).
- User name or identification (to ensure the identification or the authentication of the user).
- Location of the device (longitude and latitude where the user is located).
- Transmitting channels (channel(s) that the device is using).
- Transmit power.

· <u>WSD data:</u> data related to unlicensed secondary WSDs. When a WSD is using a channel it should be registered to the database to know that that channel is not available. The following data about WSDs will be stored:

- Device type (indicates if the WSD is a fixed WSD or a personal/portable WSD).
- WSD identifier.
- WSD manufacturer serial number (these two parameters are used to authentication and security).
- Location of the device (coordinates of the WSD).
- Transmit channel (the channel that the WSD is using in that moment).

· <u>Regulatory data:</u> some required parameters are established and indicated by regulatory entities. Some of these parameters are the following:

- Regulatory ID (defines or identifies the regulatory body).
- Propagation model or algorithms (propagation model(s) such as Free Space, Rice, Egli etc. that should be used to calculate the propagation loss and the received power in destination).
- White space availability threshold (the threshold value to decide if a channel is avail able or not, if the received power is lower that the threshold, the channel will be considered available and vice versa).
- Some other information such as regulatory controls to stop or modify some device operation, administrative values, updating and valid periods etc not determined yet since the regulatory entity has not been named yet.

To create the database, the relational database management system MySQL will be used (http://www.mysql.com/). It is useful in full-featured database management systems and in web applications as our demo. It has the advantage that many different programming languages, including Java, have libraries for accessing MySQL databases.

<u>TVWS calculation and DB manager</u>

This block is similar to the device DB block and is represented in Figure 4.6. From time to time (this period of time will be determined by regulatory entities) the DB manager will access and manage the database to get the necessary information to calculate the available free channels. In other words, the DB manager will access the stored data and calculate the available channels for all the pixels. The results will be stored in the database using MySQL.

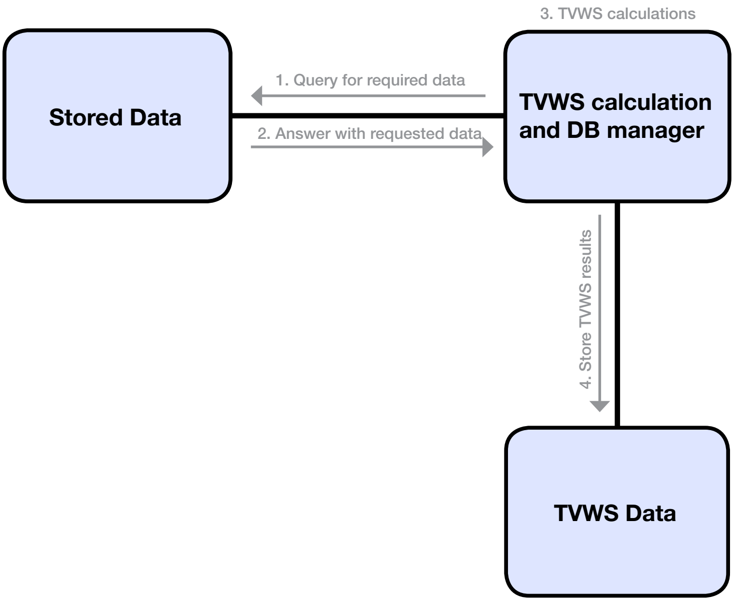

Figure 4.6 Calculation and DB manager structure

To calculate the available channels for a location, first of all, a path attenuation relative to the pixel location will be applied to the transmit power. There are different propagation models: Free Space, Hata, Rice, Egli etc. Once the power that reaches the pixel location is known, the easiest or simplest way to determine the available white spaces is to use a threshold. If the power in the location is lower than this threshold, the channel will be considered to be free. Otherwise, when the power is higher than the threshold, the channel will be classified as occupied. Transmit towers, secondary licensed users and WSDs (in case that they are registered) that are using a channel in a pixel have to be taken into account when calculating the available channels.

Even if the update periods are not determined yet, it seems logical that the licensed user calculations should not be updated very often while the updates related to WSDs should be done more frequently because they will vary more frequently.

The DB manager, apart from getting the required information from the stored data, will also access to the TVWS data when a public user queries for the available channels in a location.

The application will be constructed using the object-oriented Java programming language since it is a general-purpose, low dependencies popular platform for application software. To develop the software we will use the Eclipse Integrated Development Environment (IDE). It allows working with database management systems, different frameworks and different languages such as Java, PHP etc. To make the work easier, an open-source framework called Spring will also be used (http://www.springsource.org/). It helps to implement the standard structure of the application by inheriting parts from previous applications.

TVWS data

Once the white spaces for all the pixels are calculated they will be stored in a database. In this way, when a public user wants to know the free spaces in a location, the DB manager will directly access the TVWS data and get the available channel list for the given location, without any real-time computation. In the TVWS database the following data will be stored:

· Pixel identification or location.

· Available channel number (the numbesr of the channels that are available for that pixel).

· Other optional data such as maximum transmit power allowed or validity of the information.

Once more, MySQL will be use to create and administrate the database.

Web interface

In order to offer public access to the database, a web interface will be created. It will be a web application that allows introducing the location (address or coordinates) of the user and it will return to the user a list with the available channels for the pixel where the location is situated. Apart from the Spring framework, HTML based JSP will also be used to create the web interface. It will collect public users' location data and it will return the corresponding available TVWSs.

Operation

Similar to the device DB, the public user will query for available channels in a location. This location will have to be translated to a pixel (the precision of the location will be lost and substituted by a representative pixel). In this case, since all the TVWS calculations have been done in advance, the DB manager will get the input pixel identificator and will access the TVWS data to get the corresponding available channels for that pixel. The list will be presented to the public user using the web interface.

4.2 DATABASE METASTRUCTURE

MySQL is a relational database system where the data is organized in different tables with different attributes. Each table has a unique primary key that identifies every object/raw of the table uniquely. Each table might also have a foreign key that relates the table to another table.

The general structure of the database for our database design is shown in Figure 4.7. The database will be formed by twelve different entity groups: TV towers, Antennas, Other licensed users, WSD, Measurements, WSD occupancy, Antenna occupancy, Other occupancy, Regulatories, Pixels, Estimated results and Channels.

The attributes of each entity group are represented in one color and the darkest attribute represents the unique key of the group. Every row of an entity group has to be identified uniquely by this key. The relationships between different groups are indicated with a different color in the border line of the attribute.

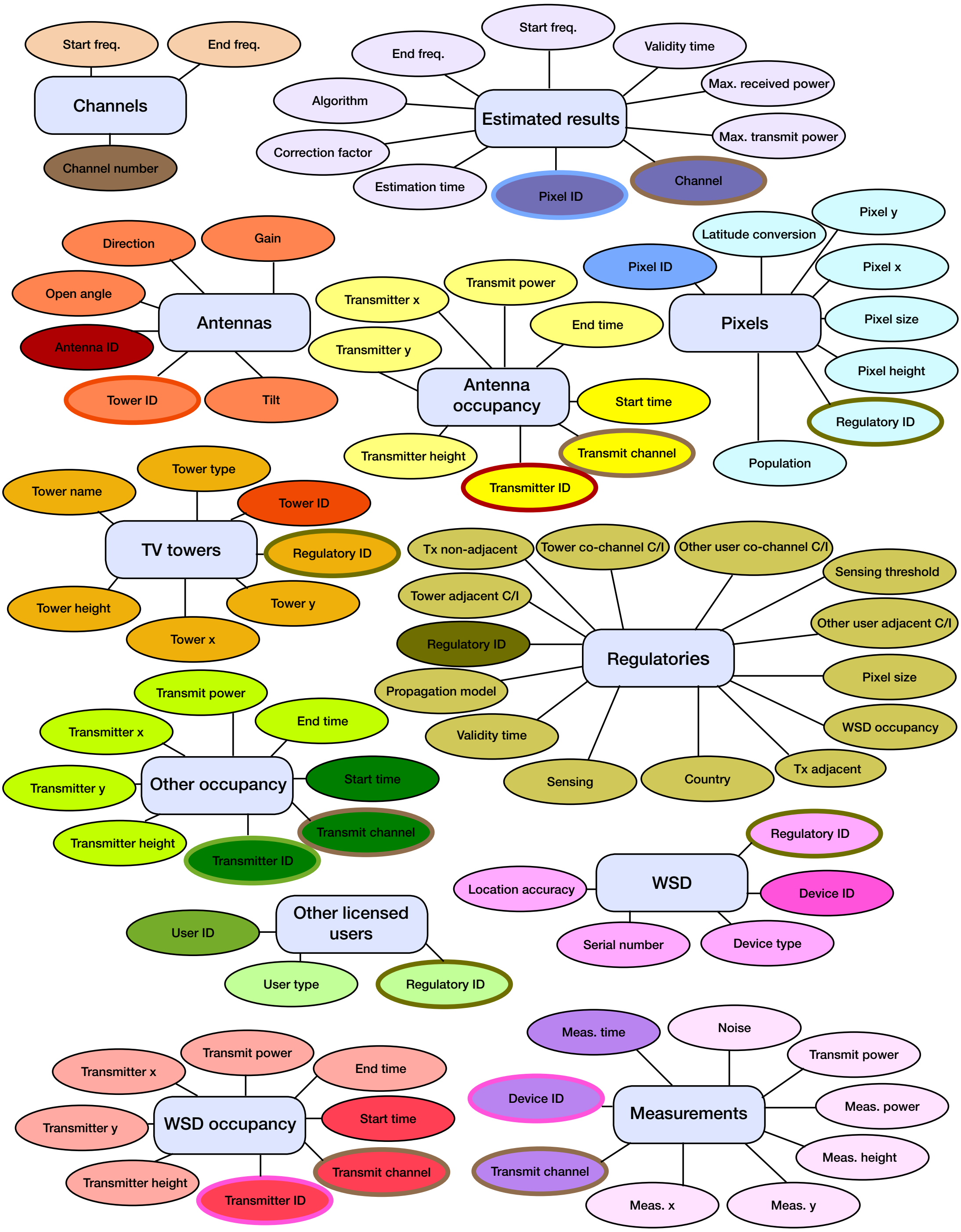

Figure 4.7 Database general metastructure

4.2.1 ENTITY GROUPS

Regulatories

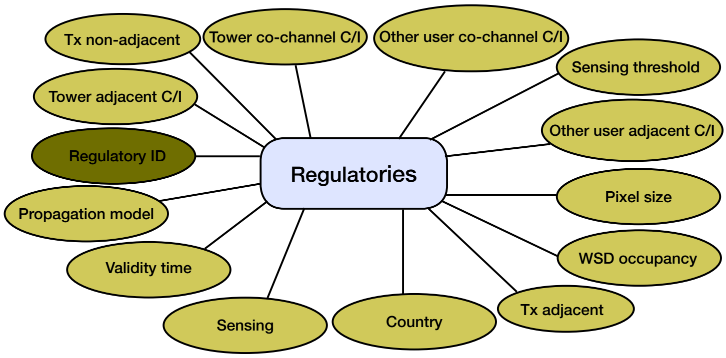

Figure 4.8 Regulatories entity group

The Regulatories table will store information about the different entities that regulate and control the operation of white space databases (Figure 4.8). The regulatory policies are not yet officially defined in Europe (FCC in USA). Each country will have one regulatory and this entity group allows the development of the database system through several countries. The attributes of the entity can not be fixed because the regulatories are not established and specified. However, these are the initially proposed attributes that trie to support most of the different implementation and functionality possibilities:

· Unique key – Regulatory ID: It is an identifier that identifies a certain regulatory policy uniquely. It could be the name of the regulatory or a identifying number.

· Propagation model: Indicates the propagation model that the regulatory proposes when calculating the path loss for the available channels. Nevertheless, it is not mandatory that the propagation model is specified by the regulatory (depends on the regulatory entity). In case that the propagation model field says "none", a default propagation model saved in the Java program will be used for the calculations.

· Country: It is the operation country of the regulatory.

· WSD occupancy: It will be used in case that white space allocation or reservation is allowed or required. It may exist the possibility for a WSD to reserve different white spaces for different periods of time, in this way the WSD will ensure white spaces for longer periods of time. The occupancy attribute indicates if the white space pre-allocation or registration service is being used. When affirmative, the database system will have to consult the occupancy table and when negative it will directly make the calculations without consulting any additional table.

· Pixel size: the size of the pixels for the calculations specified by the regulatory. The Ofcom and ECC propose a 100m x 100m pixel size while the COGEU project proposes a 200m x 200m pixel size. We can conclude that each regulatory policy may specify different pixel sizes. Moreover, it might happen that a 100m x 100m size is not precise enough for urban areas with high population density and many buildings. In these cases, smaller pixels (50m x 50m i.e.) may be used. The pixel size attribute allows flexibility when establishing the size of the pixels. In fact, different results for the same pixel but with different sizes may be calculated and stored in the database if necessary.

· Validity time: Is the default value for which a database consult is valid. In other words, for how long the data from a database access is valid without reupdating it.

· Tower co-channel C/I: Indicates the required interference protection for signals transmitted by TV towers in the channel of interest. It will determine the minimum power level difference between the received TV signal and the transmit signal by the WSD.

· Tower adjacent C/I: indicates the required interference protection for signals transmitted by TV towers in channels adjacent to the one of interest. It will determine the minimum power level difference between the received TV signal in adjacent channels and the transmit signal by the WSD.

· Other user co-channel C/I: Indicates the required interference protection for signals transmitted by other licensed users in the channel of interest. It will determine the minimum power level difference between the received licensed signal and the transmit signal by the WSD.

· Other user adjacent C/I: indicates the required interference protection for signals transmitted by other licensed users in channels adjacent to the one of interest. It will determine the minimum power level difference between the received licensed signal in adjacent channels and the transmit signal by the WSD.

· Sensing: Indicates if additional sensing is required or not. Ofcom and COGEU i.e. propose a combined geo-location and sensing technology for TVWS detection. FCC on the other hand, only requires geo-location technology. When additional sensing is necessary, the measurements from the WSDs will also have to be taken into account when making calculations.

· Sensing threshold: is the maximum received power level from TV towers allowed to consider a channel free. If received power detected is above this threshold, the channel is occupied, if it is below it is available. (For instance, for FCC this was -114 dBm in 2008 rulings).

· Tx adjacent: is the allowed transmit power if one of the adjacent channels is occupied. (In FCC, this was 40mW for mobile devices, what is equivalent to 16 dBm).

· Tx non-adjacent: is the allowed transmit power if adjacent channels are not occupied. (In FCC, this was 100mW for mobile devices, what is equivalent to 20 dBm).

Note that other parameters may be included in the regulatory entity when the database system is developed in more detail and when the regulatories become official. At the moment it is not easy to determine what values and parameters will the regulatories provide so this is just an approximation.

TV Towers

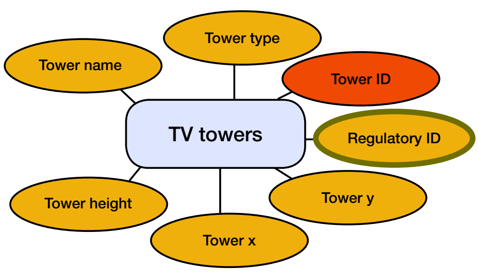

Figure 4.9 TV Towers entity group

The TV towers table will have information about the transmitting TV towers (Figure 4.9). The information we will get about the TV towers will have to be accorded with the TV broadcasters in each region or place so the attributes that I defined here are just initial and temporal, some modifications could be done in the future.

· Unique key – Tower ID: It is a unique identifier for the TV tower. It is used to identify each TV tower uniquely.

· Tower name: is the name given to the tower, making easier its identification.

· Regulatory ID: Informs about the regulatory policy that is in charge of the TV tower or the area where the TV tower works. In consequence, in the future it will be possible to expand the database system to different regions or countries and control the TV towers that form part of each country/region depending on the regulatory entity.

· Tower type: Indicates the type of tower (DVB-T, anlogical etc.).

· Tower x: Is the x coordinate of the TV tower that helps to determine the location of the same.

· Tower y: Is the y coordinate of the TV tower that helps to determine the location of the same.

· Tower height: Is the height of the TV tower above an average level.

Antennas

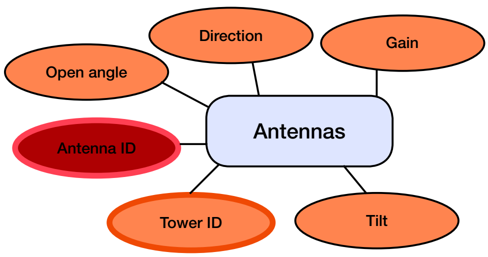

Figure 4.10 Antennas entity group

Each TV tower may have more than one transmit antenna. The Antennas entity group stores information about the particular antennas (Figure 4.10). In the same way as with the TV towers, the attributes of Antennas may vary according to the TV broadcasters.

· Unique key – Antenna ID: The antenna ID is an identifier that identifies each antenna uniquely.

· Tower ID: Indicates the tower to which the antenna corresponds.

· Tilt: The tilt of the antenna, it may be required to calculate the propagation loss.

· Gain: Is the gain of the antenna. The transmit power will be increased by this gain before transmitting.

· Direction: It is the central direction of transmission of the antenna.

· Open angle: Indicates the angle among the central direction that the antenna is able to transmit. Outside this open angle there will not be transmissions from the antenna.

Other licensed users

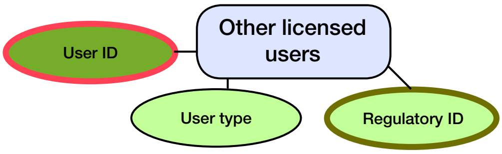

Figure 4.11 Other licensed users entity group

This table contains information about secondary licensed users (Figure 4.11).

· Unique Key – User ID: the User ID identifies the licensed user uniquely.

· Regulatory ID: Informs about the regulatory policy that is in charge of the licensed user or the area where the user is working. In consequence, in the future it will be possible to expand the database system to different regions or countries and control the licensed users that form part of each country/region depending on the regulatory entity.

· User type: indicates the licensed user type.

WSD

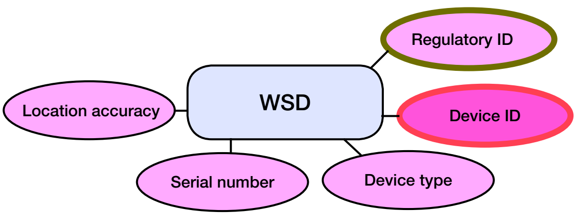

Figure 4.12 WSD entity group

This table contains information about the WSDs that have registered to the database system (Figure 4.12). The information is mainly used to identify and authenticate the device and keep control of the WSDs that make use of the system.

· Unique key – Device ID: Is the identifier that identifies the device uniquely.

· Regulatory ID: Informs about the regulatory policy that is in charge of the WSD or the area where the WSD works. In consequence, in the future it will be possible to expand the database system to different regions or countries and control the WSDs that form part of each country/region depending on the regulatory entity. When calculating the available channels the rules of this regulatory will be used.

· Serial number: The serial number of the manufacturer. It serves for authentication and contact information when needed.

· Device type: Indicates the device type (fixed, personal, portable etc.)

· Location accuracy: Indicates the accuracy of the location provided by the WSD. It is important to have this attribute for calculations when WSD registration is required because depending on the location of the WSD within the pixel and on the accuracy, we may do not know for sure if the device is located in a pixel or in the contiguous one. In the cases when we do not know the exact pixel of the WSD, the calculations should be done for all the possible pixels and choose the most restrictive results to ensure the interference avoidance and proper operation of the system.

WSD, Other users and Antennas Ocuppancy

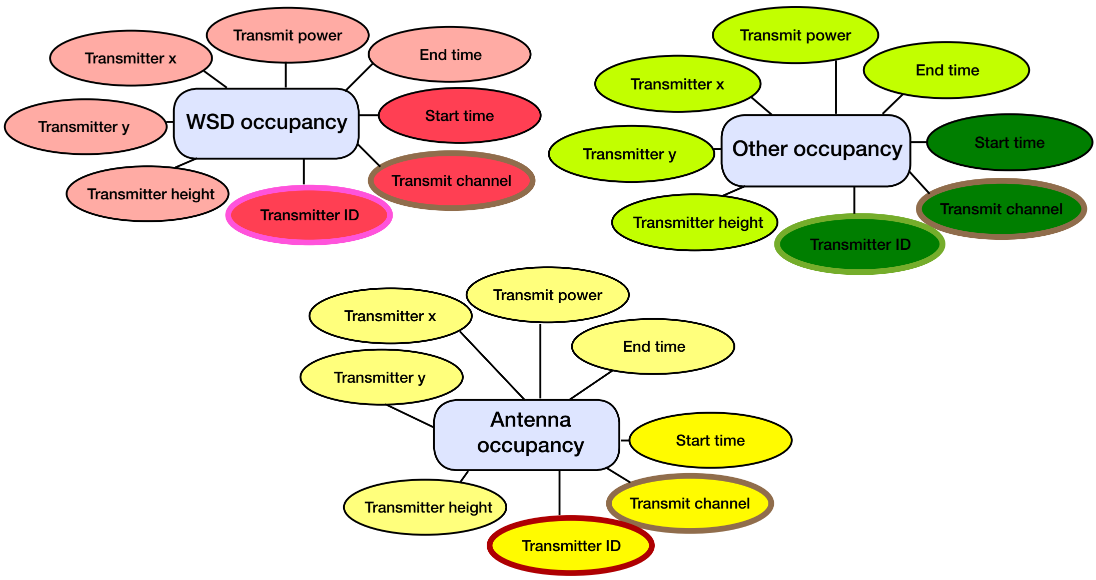

Figure 4.13 Ocuppancy entity groups

These tables have information about the occupancy of different transmitters (antennas, other licensed users and WSDs). In this way, it is possible to identify for all the transmitters the different transmission channels and frequencies for each moment and location (Figure 4.13).

· Unique key - Device ID, Transmit channel, Start time: The device ID is the unique identifier of the transmitter (it can be device ID, antenna ID or other user ID) that identifies which device is using each channel and also allows the access to the different transmitter entity groups to get information about the transmitters. However, the same transmitter can use different channels in different moments. That is why we need the transmit channel (that indicates the channel that the transmitter is using for transmission) and the starting time of transmission of the transmitter in this channel (that indicates when the transmission for the indicated channel started or will start).

· End time: Informs when the transmition in the given channel is going to stop. It used to know when and for how long each channel will be occupied.

· Transmit power: informs about the power that the transmitter is transmitting or will transmit.

· Device x: Is the x coordinate of the transmitter during the occupancy time of the given channel. It helps to determine the location of the transmitter.

· Device y: Is the y coordinate of the transmitter during the occupancy time of the given channel.

· Device height: Is the height of the transmitter abouve an average level during the occupancy time. The transmitter height, x location and y location are used to determine the exact location of each transmitter during the occupancy time. Since the location of the same transmitter can vary from one occupancy to another (portable transmitters), it is important to save the location of each white space access.

The first idea was to put the occupancy table with the pixels; in other words, to save the channels that were occupied or reserved for each instant in each pixel. However, I finally decided to save the reserved channels for each instant by each transmitter because of the location accuracy. In the case of storing the occupancy for each pixel, the location accuracy is very low (with 100m x 100m pixels there might be a almost 100 m error), while storing the occupancy for each pixel limits the error to the location accuracy of the transmitter which is avoided by taking the most restrictive case.

Measurements

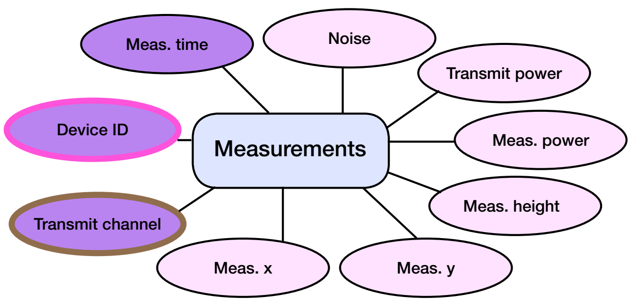

Figure 4.14 Measurements entity group

It could be interesting to save the sensing results of the registered WSDs (Figure 4.14). In this way, the geolocation database results could be complemented by the sensing results to provide more accurate and trustable results. These measurements could also be used by the database to "learn" and adapt the calculation process. The measurements entity group stores information about the sensing of the registered devices.

· Unique key – Device ID, Transmit channel, Measurement time: the device ID indicates the WSD that have taken the sensing measurements. In the same way as with the occupancy, one device can provide measurements for different channels and for different instants of time, being necessary the transmit channel of the measurement and the time when it was taken.

· Measurement x: Is the x coordinate where the measurement was taken.

· Measurement y: Is the y coordinate where the measurement was taken.

· Measurement height: Is the height above an average level where the measurement was taken.

· Measured power: Is the actual received or measured power.

· Transmit power: Indicates the maximum allowed transmit power for the given channel according to the sensing. It is the maximum power that a WSD can use for transmission in this channel, location and time.

· Noise: Is the measured noise in the indicated channel, location and time.

Pixels

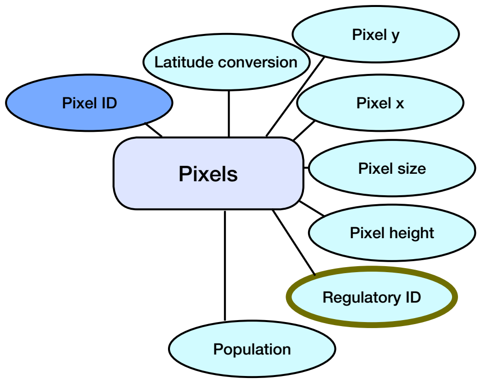

Figure 4.15 Pixels entity group

This entity group has certain information about the pixels (Figure 4.15).

· Unique key - Pixel ID: Is the unique identifier for the pixel.

· Pixel x: Indicates the inferior (westest point) x coordinate of the pixel.

· Pixel y: Indicates the inferior (southest point) y coordinate of the pixel.

· Pixel size: The database system allows flexibility when choosing the pixel size and the same has to be indicated to enable the existence of different sizes of pixels for the same x and y coordinates.

· Pixel height: Is the height of the pixel. It can be calculated taking the heights of the four corners of the pixel and resolving the mean/median.

· Regulatory ID: Informs about the regulatory policy that is in charge of the pixel area. In consequence, in the future it will be possible to expand the database system to different regions or countries and control the pixels that form part of each country/region depending on the regulatory entity.

· Population: Is the population of the pixel that helps to distinguish between rural and urban areas and make some propagation modifications to get more accurate results when calculating the available white spaces.

· Latitude conversion: The location data will be stored in degrees in the database system. A degree-meter equivalence is needed to realize the conversion to calculate distances. As we will discuss later, this equivalence varies from one location to another depending on its relative location to the ecuator. To make the conversion as accurate as possible, each pixel will have this equivalence or translation value stored.

Note that more information about the terrain of the pixel could be added (in the Microsoft case i.e., information about the terrain is taken from the NASA). This physical information of the pixels could be used to make the calculations more exact and reduce the error.

Estimated results

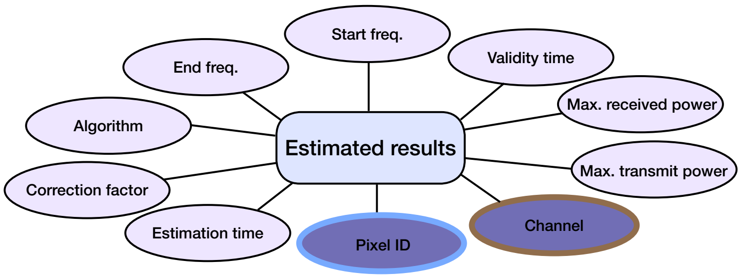

Figure 4.16 Estimated results entity group

It stores the results of the calculations of the system (Figure 4.16). Every certain period of time, the database system calculates the available channels for each pixel and the results are saved in this table. When a WSD or a public user accesses the database system querying for available channels for a certain location, the database will answer with the content of this table for the given location.

· Unique key – Pixel ID, Channel: The unique identifier of each pixel (pixel ID) and the channel number identify each result uniquely. In this way each estimation for each channel of every pixel will be idenfied uniquely.

· Estimation time Is the time when the calculations were done. In the beginning, it was thought to store all the estimations over the time. However, due to the exponential increase of the amount of data, finally it was decided to only save the last estimation, the most recent one. Since all the occupancies/transmissions of all the users are saved in the database, in case that in a given moment previous results are needed, it is relatively easy to make the calculations for past times and calculate the history of estimations. In this way the amount of data stored in the database is reduced considerably.

· Maximum power: Is the maximum allowed transmission power for WSDs in the given channel, pixel and instant of time.

· Maximum received power: Is the actual estimated power in the centre of the pixel, the estimated or calculated received power in that pixel.

· Validity time: Indicates for how long it is possible to use the given channel in the given location. Some regulatories establish the maximum of this validity time in 2 hours. However, if the occupancy reservation is allowed, the calculation of the validity time will become more complex since a channel can be available in the moment but reserved by a WSD for later use.

· Start frequency: Is the starting frequency of the available "channel". It may happen that a channel is being used by another WSD or secondary licensed user. Nevertheless, the WSD will not occupy the whole channel and part of the same could be reused by another WSD. The start frequency value indicates where the available white space starts within the channel. In case that this application is not allowed, the start frequency will indicate the actual channel start frequency.

· End frequency: Is the end frequency of the available "channel", indicates in which frequency the white space ends (in the same way as with the start frequency it can be the same as the channel end frequency or another value within the channel frequency range).

· Algorithm: Informs about the algorithm used for the propagation loss calculation.

· Correction factor: By default it will be set to zero. However, when complementing the estimated results with the measured sensing results of the WSDs, a correction factor can be deduced. If we compare both of the results, it is probable to find a small variation between them. If we do the comparisons for different sensing measurements of different WSDs among the time, the estimated results could be optimized and improved by this correction factor that may correct errors of our propagation algorithm such as mountains, buildings, special environmental conditions etc. How to determine the value of this correction factor has not been decided yet, it is just included for possible future implementations.

Channels

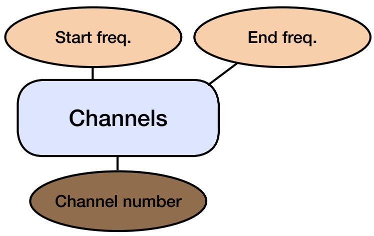

Figure 4.17 Channels entity group

This table saves information about the different channels (Figure 4.17). When working on TV bands, the channels are defined but this table allows the extension of the service to other frequencies.

· Unique key – Channel number: Is the number of the channel.

· Start frequency: Indicates in which frequency the channel starts.

· End frequency: Indicates in which frequency the channel ends.

4.2.2 RELATIONSHIPS BETWEEN ENTITY GROUPS

In Figure 4.18 the relationships between different entity groups are shown. There, the unique keys are represented as well as the attributes (foreign keys) that relate the entity to another one. All the relationships are 1:N relationships. This means that one element from an entity group can be related to N elements to the other group, while each element of the other group is related to just one element from the initial entity group.

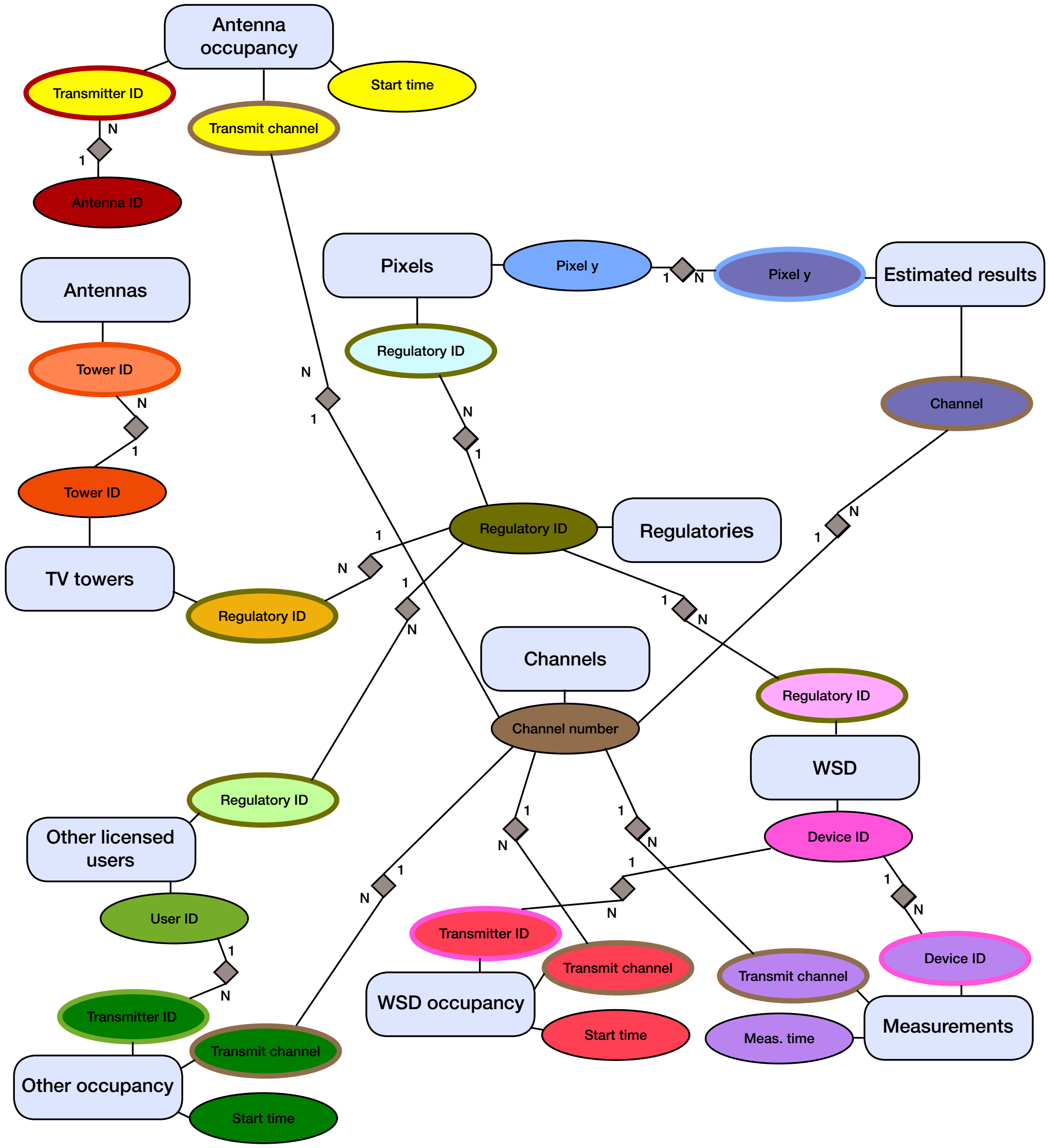

Figure 4.18 Database relational scheme

As shown in Figure 4.18, the Regulatories entity group has a 1:N relationship with TV towers, Other licensed users, WSDs and Pixels. This means that each regulatory may have several TV towers, Other licensed users, WSDs and Pixels. In this way, in each regulatory area, we will have to make calculations for the components related to this regulatory.

The Channels entity group is related to different Occupancies, Measurements and Estimated results with a relationship 1:N. In the channel entity group several information about the channel number from Occupancy, Measurements and Estimated results is stored.

In a similar way, the Antennas, Other licensed users and WSDs have a 1:N relationship with their respective Occupancy entity group. While in Antennas, Other licensed users and WSDs, physical information is stored, the Occupancy table saves information about the transmissions and use of the channels of these transmitters.

WSDs also has a 1:N relationship with Measurements. For each WSD several measurements for different channels and instants of time are stored.

On the other hand, the TV towers entity group is related to Antennas (1:N). Each TV tower can have several antennas. In this way, it is not necessary to save the same information of a TV tower for all the antennas it has.

Finally, the Pixels entity group has a 1:N relationship with Estimated results. This relationship is represented by a composed foreign key (Pixel ID, channel). In Estimated results, several calculations and results (different channels) are saved for each pixel.

In relational database systems, the use of foreign keys to relate different entity groups avoids the repetition of the same information more than once, reducing the stored data.

Note that the database has been designed to be as flexible as possible. Since the regulatory entities have not been defined yet and database systems have not become commercial yet, many modifications may be done during the development and with the practice. In consequence, it is convenient to provide the database with as much flexibility as possible to add new services and applications or extend its use to new ones.

That is why when designing the database system, it was tried to take in consideration as many factors as possible, even if they are not implemented in the demo. Without real data and real specifications, it is quite difficult to implement many applications or added services. Nevertheless, the design tries to keep the system as open as possible to changes and modifications.

4.3 PHYSICAL DATABASE CREATION

Once the initial database system and database metastructure are defined, the actual or physical database has to be created.

For that, as mentioned before, MySQL language and a MySQL server will be used (http://www.mysql.com/).

First of all, the database and its tables should be created with their respective attributes, primary keys and foreign keys. In section 4.2, the required attributes where defined but the attribute data types were not specified. So it is important to appropiately decide what kind of data does each attribute contain (Integer, Float, TimeStamp etc.). When creating the database and its components, the MySQL WorkBench tool was used (http://wb.mysql.com/). The database creation code is available in Appendix A.

In Figure 4.19, the resulting database scheme is shown.

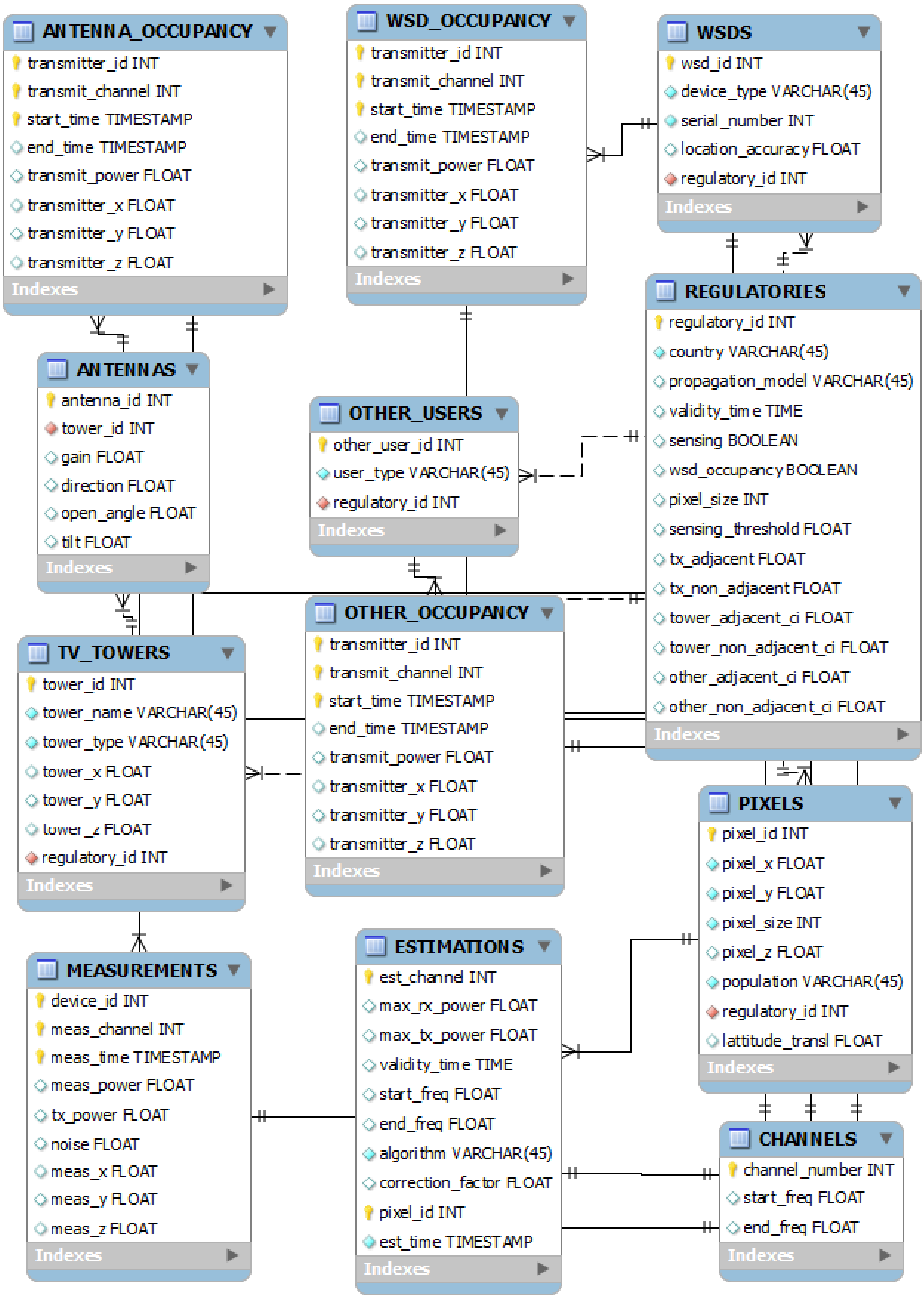

Figure 4.19 MySQL WorkBench database scheme

After the database itself is physically created, the required data for TVWS calculations needs to be stored there. I did not have real data sources and interfaces with them and in consequence, some sample data was introduced manually.

The case of Norway was choosen: first of all, a regulatory for the country was created and then the data related to that regulatory was inserted (TV towers located around Oslo, some secondary users, some sample pixels in Oslo etc.).

Note that even if the TV towers' location and transmit channels for the same are real in the demo, the transmission values such as power, tilt, height etc. are just random sample data. This involves that the results and estimations will be quite approximate but not precise or correct.

4.4 TVWS DATABASE SYSTEM METASTRUCTURE

Now that the initial database system and database metastructure are defined, and a physical database has been populated, the initial code skeleton and class diagrams must be determined.

As mentioned before, we will use Java programming language with the Spring framework (http://www.springsource.org/) to create the main body of the database system. Thanks to the Spring framework, which is open source, the code will be reusable in the future and easier to reconfigure and make modifications (in Spring, the code is based in beans and the implementation of each bean is defined in a very easily modifiable configuration file). In adition to the Spring framework, and in order to map the Java obect oriented domain to the MySQL relational database, the object-relational mapping (ORM) Hibernate persistence layer framework (http://www.hibernate.org/) will be used. In fact, one of the objectives when designing the demo program structure, is to make it as transparent as possible, hiding in the way it is possible the implementations.

That is why we can distinguish between two class diagrams: one for the implementation itself and another one for the interfaces. The interfaces hide the implementations, making it very easy to change from one to another using the Spring framework. The Spring framework uses a configuration file that defines what implementation will be used for an interface. So we just have to modify the Spring configuration file to change the implementation (if we want a non-relational database model, calculate TVWSs in another way etc.), instead of rewritting all the code again.

4.4.1 INTERFACE CLASS DIAGRAM

In Figure 4.20, the Universal Modelling Language (UML) class diagram is shown for the interfaces part. There, the different interfaces that constitute the code and the relationship between them is represented.

The Spring framework structures the code according to the Model, View, Controller (MVC) schema. In our case, we will distinguish the database access and control part, the model part, the service part and the view part.

The database access part Data Access Object (DAO), is formed by objects that manage the access to the database, making it independent from the rest of the application. For each table in the database we will have a DAO interface: RegulatoriesDAO, ChannelsDAO, TvTowersDAO, AntennasDAO, OtherUsersDAO, WsdsDAO, AntennaOccupancyDAO, OtherOccupancyDAO,

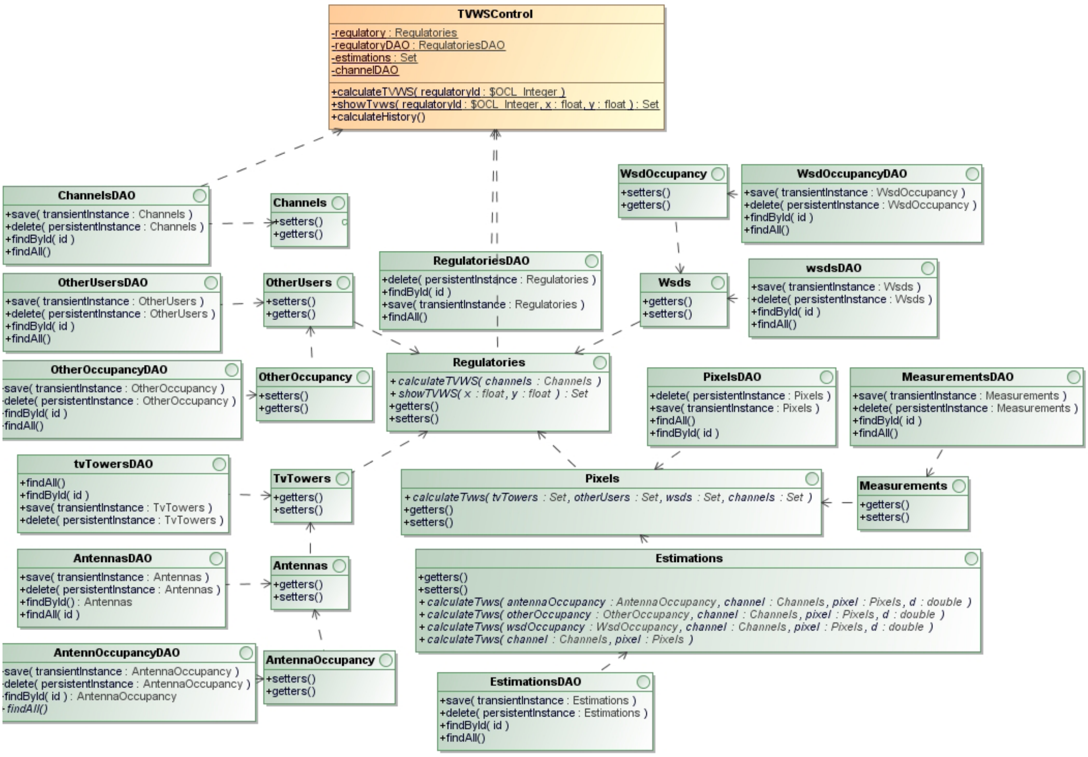

Figure 4.20 Interface UML class diagram

WsdOccupancyDAO, PixelsDAO, EstimationsDAO and MeasurementsDAO. These interfaces are responsible for reading, saving, updating, deleting and finding data fromt he physical database.

The data accessed from the database by the DAO part will be manipulated by the model part. The Regulatories, Channels, TvTowers, Antennas, OtherUsers, Wsds, AntennaOccupancy, OtherOccupancy, WsdOccupancy, Pixels, Estimations and Measurements interfaces constitute the model part. They keep and manipulate data from database tables during execution.

The main functions or operations of the implementation are controlled by the service part, called TvwsControl in our case. It contains the main methods that perform the main functionality of the database system. These methods are CalculateTvws, ShowTvws and CalculateHistory.

· <u>CalculateTvws:</u> This method realizes all the calculations for all the pixels and all the channels within a given regulatory entity. For that, it first loads information about the channels and the given regulatory. It then calles the CalculateTvws method from the Regulatories interface. This Regulatories CalculateTvws method makes the calculations for all its pixels using the TV towers, other users and WSDs that are within its domain. In other words, it iterates over all the pixels and calls the CalculateTvws method from the Pixels interface in each case. The Pixels CalculateTvws method calculates all the estimations for all the channels for a particular pixel (it calls the CalculateTvws method from the Estimations interface for each channel in the pixel). The CalculateTvws method in the Estimations interface calculates if the given channel is free for the given location and if afirmative, it calculates additional parameters such as received power, allowed maximum power, validity time etc. The results are saved in the Estimations table.

· ShowTvws: This method shows the available TVWSs and some information related to them for a given location. Using the location and the regulatory entity as input parameters, it returns a list with the white spaces and their characteristics for the given location. For that, it first loads information about the channels and the given regulatory. It then calls the ShowTvws method from the Regulatories interface, which finds the corresponding pixel for the input location (loosing accuracy) and loads the estimations for that pixel that are saved in the database.

· CalculateHistory: As mentioned before, for functional and storage reasons, it was decided to save in the database the most recent calculation only. This does not suppose big inconvenient or capability loss since all the transmissions among the history are saved in the database and thus, the calculations can be done for past times when necessary. The CalculateHistory method is the responsible for making the historical calculations for statistics, metrics, analysis reports etc. This method is thought to be developed in the future when expanding the features of the application.

In the same way as with the CalculateHistory method, other value-adding methods could be implemented in order to offer more services to the user or WSD. When the geo-location database systems are more developed, this methods and services will be more defined and clear. Thanks to the Spring framework, the interfaces and the open strucuture of the code, it should be easy to add more functionality to the same and reuse the existing one.

Finally, the view part of the application is the web interface that will be developed after ensuring that the service is working (Section 4.6).

4.4.2 IMPLEMENTATION CLASS DIAGRAM

In Figure 4.21, the UML class diagram for the implementation part is represented. This diagram shows the particular implementations of the interfaces described in the previous section. The implementaion classes implement the functionalities of the interfaces.

Regarding to the DAO part, the implementation classes implement the concrete case of MySQL database access using Java DataBase Connectivity (JDBC) connection (save, update, findById, findAll, delete etc.). In case that other database types are wanted to be used, we will just have to define a new implementation class and indicate the change in the Spring configuration .xml file. The hibernate framework will also be used to manage the required transactions to access the database and to define the mapping of the tables.

All the DAO implementations are similar and have similar structure so for simplicity, in Figure 4.21, just the RegulatoriesDAOImpl class has been represented. However, note that one DAOImpl class will exist for each DAO interface.

The model part is implemented so that the data manipulation is performed in a particular way (first loading all the required data from the database to minimize the accesses to the same and then manipulating the loaded data one by one). The calculations, data modifications etc. are done using given algorithms, methods or functions. If we need or want to calculate the TVWSs in another way, we just have to modify the implementations.

See that when the primary key of a database table is composed, a new class is defined to manage data related to the primary key. This is the case of EstimationsId, AntennaOccupancyId, OtherOccupancyId, WsdOccupancyId, MeasurementsId.

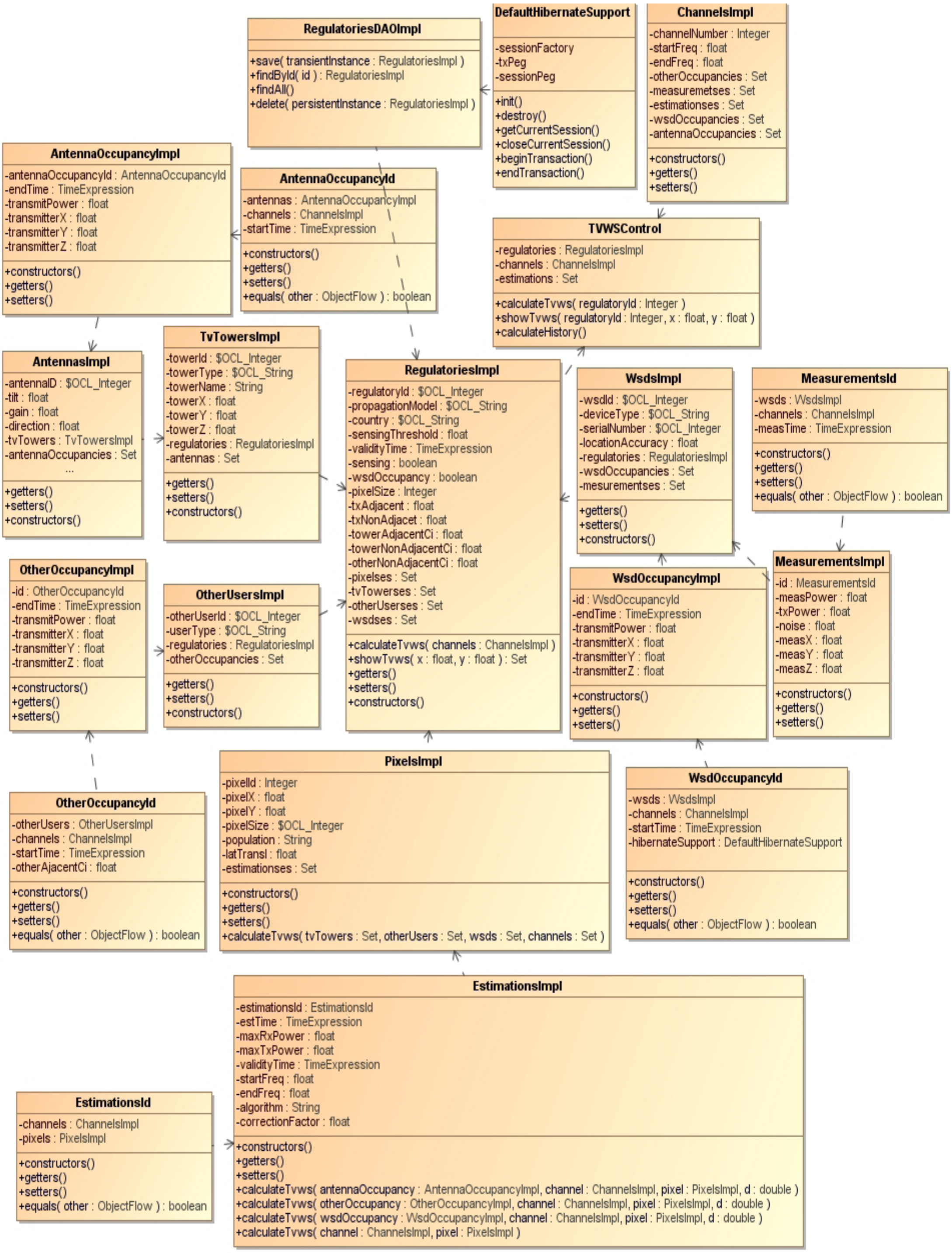

Figure 4.21 Implementation UML class diagram

4.5 CODE DEVELOPMENT

After defining the initial code skeleton the code itself has to be developed. Using the Eclipse tool (http://www.eclipse.org/) and Java object-oriented programming language the code for the TVWS database system application is written. The code is primarily divided into two parts: source (src) and configuration files (META-INF).

4.5.1 SOURCE CODE

The source part of the code is mainly consituted by the UML diagram classes and interfaces commented in section 4.4. It is divided in several packages: dao.interfaces, dao.implementations, model.interfaces, model.implementations, service and hibernate (Figure 4.22).

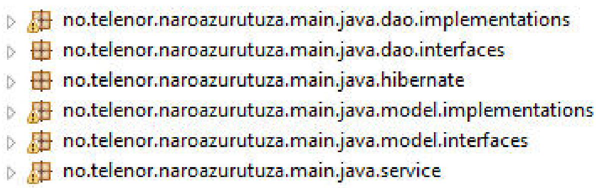

Figure 4.22 Source code structure

Hibernate

This package contains support classes and interfaces for Hibernate transactions (start, end transactions, get, close sessions etc.) for database access (Figure 4.23).

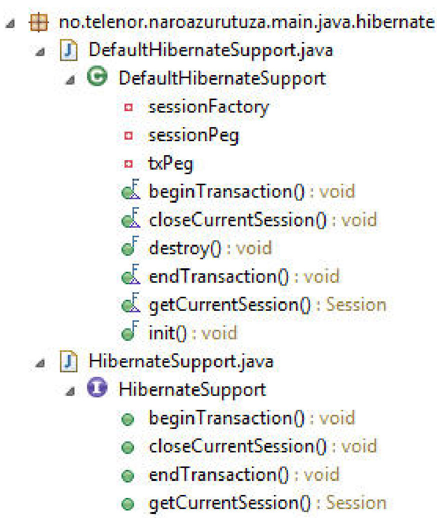

Figure 4.23 hibernate package structure

Dao.interfaces

This package contains the DAO interfaces described in Section 4.4.1, used to interact with and access to the database (Figure 4.24).

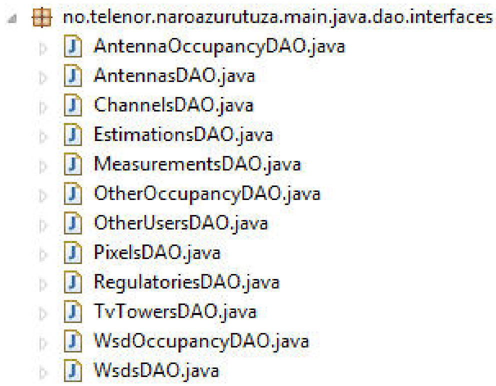

Figure 4.24 dao.interfaces package structure

Dao.implementations

This package contains the DAO classes or implementations described in Section 4.4.2 that are used to implement the access to MySQL relational databases with the help of the Hibernate framework (Figure 4.25).

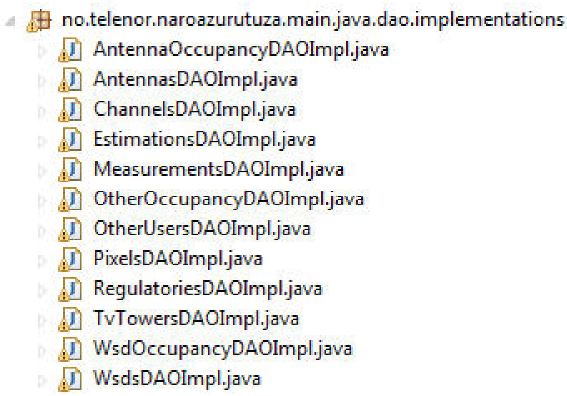

Figure 4.25 dao.implementations package structure

Model.interfaces

This package contains the Model part interfaces described in section 4.4.1 (Figure 4.26).

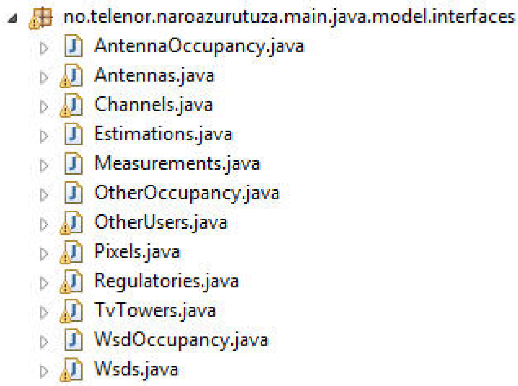

Figure 4.26 model.interfaces package structure

Model.implementations

This package contains the Model part implementations described in section 4.4.2 (Figure 4.27).

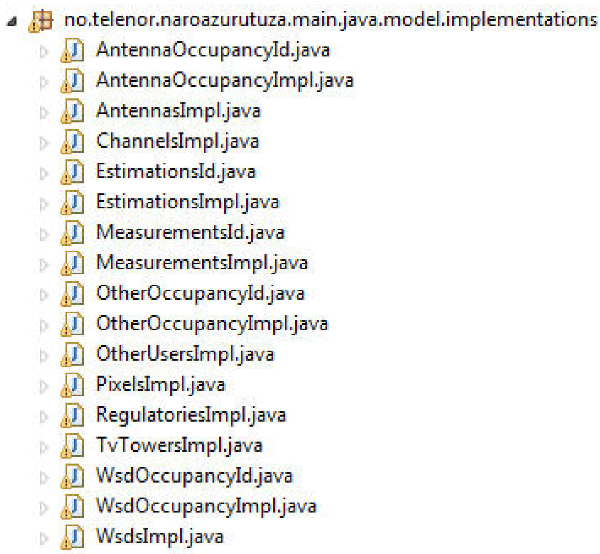

Figure 4.27 model.implementations package structure

Service

This package contains the Service part implementations or classes described in section 4.4.1/2 and that perform the main functionalities of the database system (Figure 4.28).

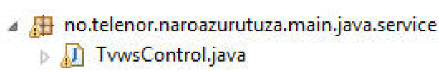

Figure 4.28 service package structure

4.5.2 CONFIGURATION FILES' CODE

There are three main configuration file types in the code structure: the Spring configuration file (applicationContext.xml), the Hibernate configuration file (hibernate.cfg.xml) and the hibernate mapping files (.hbm.xml files) (Figure 4.29).

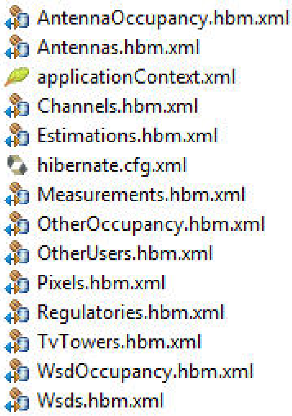

Figure 4.29 configuration files' structure

The interface and configuration code can be found in Appendix B.

4.6 USER INTERFACE

The last step in the geo-location database system demo development is to create a user interface. The user interface will be a Web interface that makes use of the code described in previous sections.

To create the Web interface, JavaServer Pages (JSP) language, together with HTML and Servlets will be used. JSP is a Java technology that enables the creation of dynamic web pages based on XML or HTML. The JSP based web page will access a Java Servlet to request for required data or operations.

The Web interface will be a very simple web page that allows the visualization of TVWSs in a given location. It will allow to introduce the x and y coordinates and it will show a list with the channels and the allowed transmit power in that location. It will be deployed using Tomcat web server (http://tomcat.apache.org/).

Although in a real user interface will not be included, the demo interface will provide the option of updating the calculations before showing the results in order to be able to check all the functionalities and proper operation of the database system.

We can distinguish between three main parts in the Web interface code: the application, the servlet and the web page code and configuration files.

The application part corresponds to the code developed and commented in Sections 4.4 and 4.5.

The servlet is a Java class (TvwsServlet.java) that interacts with the web page via doPost and doGet methods. Its structure is shown in Figure 4.30. The input parameters are sent from the JSP web page using the doPost method of a form and they are received and analyzed by the servlet, taking the corresponding decissions and executing the corresponding methods.

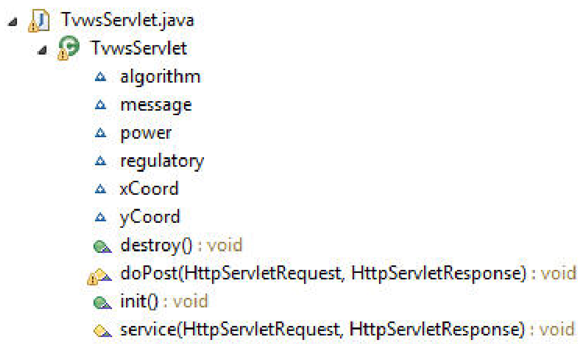

Figure 4.30 Servlet structure

The results from the servlet are passed back to the JSP web page and displayed.

The JSP web page is formed by two .jsp files: index.jsp (main page) and responser.jsp (the response redirection page).

To make the web page recognize the servlet, we have to configure the web.xml file. The JSP and configuration files' structure is shown in Figure 4.31.

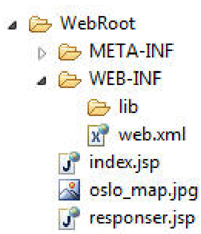

Figure 4.31 Web page files structure

The main page of the user interface is shown in Figure 4.32. We can see that it indicates to enter the x and y coordinates of the location within the Oslo area and it offers the option of recalculating the estimations before displaying the results. Once more, most of the code used to create the Web interface can be found in Appendix C.

Figure 4.32 Main page of user interface

5. DEMONSTRATION/ RESULTS

To prove the proper operation of the geo-location database system, some tests were carried out. This section shows the results and demonstrations from the tests.

First of all, the database access part (DAO part) was checked. After testing that the code was able to access (read, save, delete) the database information correctly, the main application of the database system was created.

In fact, the next step was to make sure that the application was calculating the TVWSs properly (checking that the estimation values were correct) and that it was able to show them.

Finally, the user web interface was created and tested. The results presented by using the user interface were the same as the ones obtained directly from the application.

Even if several additional functionalities could be added to the implementation, after demonstrating the proper operation of the three previous parts (DAO, application and user interface), it can be said that the geo-location database system is working properly.

5.1 DATABASE ACCESS DEMONSTRATION

Because of the simplicity and error reduction when testing the code divided into modules, first the database access part (DAO) code was created and tested. Some queries, data saving and delete operations were performed and proved that the results were in concordance with the expected ones.

In Figure 5.1 some displays of the testing results are showed (the code is in Appendix B).

```
Regulatory saved:
Regulatory Loaded from DB [id=675323,propagationModel=Haat, country=Spain, sensing=false, wsdOccupancy=false,
Regulatory updated:
Regulatory Loaded from DB Again [pixelSize=150]
Regulatory deleted
Pixel saved:
Pixel Loaded from DB [ID=111589 population=rural, pixelZ=4365440.0]
Pixel updated:
Pixel Loaded from DB Again [ID=111589 population=urban]
Pixel deleted
Channel saved:
Channel Loaded from DB [number=19,startFreq=457.0, endFreq=459.0]
Channel updated:
Channel Loaded from DB Again [startFreq=454.0]
Channel deteled
```

Figure 5.1 Database DAO part performance display

5.2 DATABASE APPLICATION DEMONSTRATION

After testing the correct operation of the DAO part, the main functionality was added to the code: TVWS calculations and result display for a given input location (since it is a demo implementation, just a few pixels were implemented and if the input location is not inside the coverage area, the application indicates that the pixel was not found).

The results demonstrate that the calculations are correct since the occupied resulting channels correspond to the channels where there was a TV tower or Secondary user transmission saved in the database and viceversa with free channels (free channels correspond to the channels where there was no TV tower or Secondary user transmission). Moreover, the correct update of the results or estimations is tested by checking that the estimation time saved in the database corresponds to the last time instant when calculations were done. It has also been proved that the parameter values of the calculation results such as pixel size, allowed transmit power, propagation model etc. are in concordance with the regulatory specifications (the regulatory specifications saved in the database).

Finally it has been tested that when an input location (x coordinate and y coordinate) inside the coverage area of the regulatory is introduced for querying for available channels in that location, the application responds with the addecuate corresponding list of estimations for the correct pixel. The values of the estimations are the same as the values of the last calculations performed.

In Figure 5.2 some displays of the testing results are showed for correct input location. In Figure 5.3, the display for incorrect or not inside coverage area location is represented (par of the code is in Appendix B).

```
calculations done
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=20.0, channel=21, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=22, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=24.922, txpower=0.0, channel=23, time=2011-06-08 11:55:51.0
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=26.4726, txpower=0.0, channel=24, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=25, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=20.0, channel=26, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=27, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=27.1747, txpower=0.0, channel=28, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=29, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=31.8373, txpower=0.0, channel=30, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=27.6741, txpower=0.0, channel=31, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=32, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=27.9953, txpower=0.0, channel=33, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=27.6142, txpower=0.0, channel=34, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=35, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=28.2195, txpower=0.0, channel=36, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=37, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=28.5195, txpower=0.0, channel=38, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=39, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=33.3801, txpower=0.0, channel=40, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=27.8282, txpower=0.0, channel=41, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=42, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=28.9394, txpower=0.0, channel=43, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=29.0775, txpower=0.0, channel=44, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=29.5111, txpower=0.0, channel=45, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=34.2143, txpower=0.0, channel=46, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=47, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=48, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=30.2824, txpower=0.0, channel=49, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=50, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=30.0001, txpower=0.0, channel=51, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=34.9915, txpower=0.0, channel=52, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=53, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=54, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=30.7926, txpower=0.0, channel=55, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=56, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=35.6008, txpower=0.0, channel=57, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=35.7188, txpower=0.0, channel=58, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=59, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=31.0858, txpower=0.0, channel=60, time=2011-06-08 11:55:51.
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=16.0, channel=61, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=20.0, channel=62, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=20.0, channel=63, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=20.0, channel=64, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=20.0, channel=65, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=20.0, channel=66, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=20.0, channel=67, time=2011-06-08 11:55:51.0,
Estimations readed for X=10.6878 and Y=59.9166 [propagationModel=hata, rxpower=0.0, txpower=20.0, channel=68, time=2011-06-08 11:55:51.0,
```

Figure 5.2 Application part performance display for correct input location

```
Location not found within regulatory
No results available for input location
No pixel found X=9.6878 and Y=50.9166
```

Figure 5.3 Application part performance display for incorrect input location

5.3 USER INTERFACE DEMONSTRATION

To demonstrate the proper operation of the User Web interface, queries for the same input locations as in 5.2 have been executed and checked that effectively the results are the same.

The user interface returns a table for the input location containing all the channels, indicating the allowed transmit power for each channel (0 if it is occupied and the calculated value when it is free). Occupied channels are represented by red squares while the available ones by green squares (the darkest green indicates that the maximum allowed transmit power is 20 dBm because adjacent channels are also available and the lightest green indicates that the maximum allowed transmit power is 16 dBm because one or both of the adjacent channels are occupied).

If the input location is not found within the regulatory domain, it is indicated that the pixel was not found and all the resulting channels are shown as occupied.

To be able to evaluate the overall correct operation and performance of the database system, the user interface offers the option of recalculating the results before showing them (there is a checkbox that allows this functionality). In fact, when the recalculation option is selected, it is proved that the resulting estimation time is updated to the actual time.

In Figure 5.4 the resulting web page is showed for a valid input location (the same as in Section 5.2). In Figure 5.5 the resulting web page for invalid input location is represented. Once more, the code can be found in Appendix C.

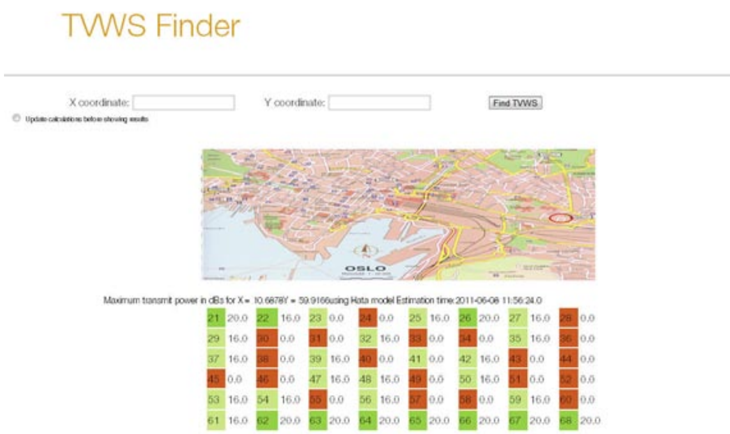

Figure 5.4 User interface performance display for correct input location

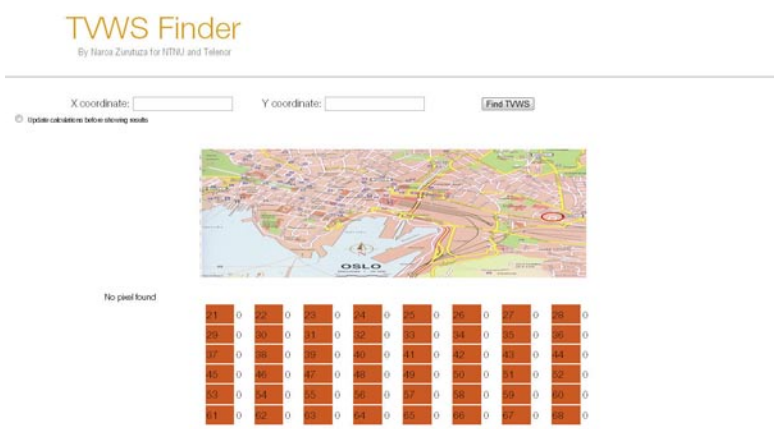

Figure 5.5 User interface performance display for incorrect input location

6. CONCLUSIONS

After researching in TVWS communications and implementing a geo-location database system, it can be concluded that it may offer a really profitable (both economically and in terms of communications QoS) alternative to traditional static communications, solving the lack of spectrum issue at the same time.

Analyzing the results from the database system, it has been demonstrated that putting extra efforts and time in developing a detailed and complete initial design, its later implementation can be simplified considerably. In this way, we do not require large amount of resources and infrastructure, being the results quite good. The small amount of necessary resources for its implementation makes the geo-location database system a good candidate for commercial developments. However, it should be noted that with more detailed propagation models using terrain data, the required computing resource will become higher. This will also lead to better utilization of white spaces and to more accurate and trustable results.

Nevertheless, due to the unprecission of propagation models, the results may not always fit with reality. This estimation error is despreciable in most of the cases but it may happen that in some occasions additional sensing is required for more trustable results (this is the case that Ofcom and COGEU propose). The cooperation with sensing might be an interesting added feature for future implementations (considered in the thesis's database design and commented in future work Section 7). It should be noted that the simple propagation model used in this paper will result in more false negatives reported from the database than false positives, thus violation of interference requirements is less probable, being the errors less severe (primary users' operation will not be affected).

Another problem that geo-location database systems have to face currently, is the lack of official standards or regulatory policies. Even if several attempts and projects are being developed, none of them is official yet (at least in Europe). This makes it difficult to determine the exact parameters required in the database system and the restrictions that the regulatories will impose. However, the database system was design to be easily reconfigurable and thus, once the official regulatories and standards are established, it will not be difficult to adapt the operation of the system to their rules.

Note that geo-location database systems are in their initial development steps and in consequence just the basic functions have been defined. In the future and with the experience, several additional features could be added to the basic functionality of the system, improving QoS and efficiency.

We can conclude that the scope of the thesis was successfully achieved: the state of the art on CR and TVWS communications was first analyzed and deep knowledge on the subject was acquired. This knowledge was used to successfully implement a geo-location database system that addresses the main requirements for proper TVWS detection and primary users' protection. Finally, to offer public access to the implementation, a web interface was created and tested. In this way and thanks to the open-source Spring framework, a publicly available geo-location database system was created. The code of the same can be publicly accessed so that third users or interested entities can reuse it and improve it in their future implementations. This will hopefully be a contribution to the research on TVWS communications and hopefully also useful for commercialization of TVWS database systems.

7. FUTURE WORK

When implementing the geo-location database system, some problems or some aspects remain unsolved due to the lack of necessary resources or specifications. Future work will intend to solve these problems and to implement the additional aspects found. In this section the most important considerations are given.

Future work will also try to adopt official standards and regulatory policies to the database system in order to support their specifications.

7.1 IMPLEMENTATION OF INTERFACES

In the initial database design (Section 4.1), 6 interfaces with other entities such as data source providers, WSDs, other database systems etc. were defined. However, in the actual database system just the user interface (Interface 6) has been implemented. The next step will be to implement the rest of the interfaces to get real data from real data sources, allow real TVWS provisioning to WSDs and enable synchronization with other database administrators.

7.2 COOPERATION WITH SENSING

The database design specified in Section 4.1, defines a table in the database for sensing measurements from WSDs. This feature was just considered, not implemented in the demo. It is thought to allow WSDs to send the database sensing measurement information.

These measurements could be used to get more accurate results and to reduce the propagation model errors. For that, some comparisons between the estimated results and the measurements should be done and the results from the comparisons should be modelled.

A way to implement this cooperation with sensing measurements, is the use of the "correction factor" attribute reserved for estimated results. From the modelled comparisson results, a correction factor could be derived. In this way, when calculating available TVWSs, the correction factor will be applied to the estimations, improving the resulting values and reducing the model propagation inaccuracy errors (mountains, buildings, fading, metheorological conditions etc.).

Another way to deal with propagation model inaccuracies is to actually measure the received power in each pixel. Since the location and transmission power of TV towers is known and varies slowly over time, it could be possible to measure the received power in each pixel and calculate and save the real path loss for that pixel from each TV tower. In consequence, it would be possible to apply the actual path loss when calculating TVWSs instead of using inaccurate propagation models. The only problem with this method is that secondary and WSD transmissions may vary more frequently over time not being possible to measure the path losses for all of them and still being necessary the propagation models. However, the TV towers case, which is the most problematic, is solved.

7.3 INITIAL PIXEL CREATION

In the demo database implementation, all the data was introduced manually (due to the lack of official data source providers and interfaces with them). Once the interfaces with data providers are created, most of the data provisioning problem will be automathized. However, the pixel data part is still not solved.

A regulatory entity has many pixels in its domain and it is not efficient to introduce all of them manually. A method to introduce the pixels corresponding to each regulatory has to be defined. This is just needed the first time they are created, from that moment onwards, its pixel will have assigned a regulatory identifier and each regulatory entity will be able to access all the pixels in its domain.

One possible solution is to use the MSB algorithm: each regulatory entity will provide the starting pixel/pixels (in case that the regulated country has islands or its composed by more than one physically separated regions, one starting pixel for each region or area will be needed) and several border pixels (indicating where does the regulatory domain end). The algorithm will take the starting pixel and it will check if its adjacent pixels (surrounding 8 pixels) correspond to boder pixels. If not, the same procedure will be repeated for the adjacent pixel. If yes, iteration will stop in the pixel (leaf or end pixel). In this way, starting from one single pixel, all the pixels covered by the border area will be iterated and created.

We found a way to iterate over all the pixels within a regulatory domain but the values of some pixel attributes are still difficult to determine. This is the case of “population” and”latTransl” attributes. The “population” attribute indicates if the pixel corresponds to a rural, urban or suburban area. This information should be provided by the regulatory or the regulatory should provide a source that stores population information for several areas within the regulatory. For each pixel, the population area to which it corresponds has to be determined.

The “latTransl” attribute provides the equivalence parameter to convert decimal latitude degrees into meters. The problem with “latTransl” is that the conversion parameter varies from one location to another, depending on its relative positition to the ecuator. The closest the location is to the ecuator, the bigger “latTransl” value will be (bigger radio representing 360º). The further the location is from the ecuator, the smaller wil be the value (smaller radio representing 360º). To calculate the “latTransl” value, the radio of the location r (in meters) has to be dividided with 360. To calculate r, knowing the ecuator radio of the Earth R = 6378 Km and the distance d from the ecuator plane to the location plane, Pitagoras theorem has to be applied: $r = sqrt(R^2-d^2)$. This problem does not exist with the longitude conversion since the longitude radios are the same for all locations (6356 Km).

It was decided to implement the lattitude conversion attribute in the pixels entity instead of in the regulatory because in large countries or physically extense contries (specially in the North part), the value of “latTransl” may be different from one location within the country to another.

7.4 IMPROVEMENT OF PROPAGATION AND HISTORY CALCULATION

One future work could be the improvement of the propagation model. It could be really interesting to implement a detailed propagation model taking real terrain data into consideration. This would probably take some time, but it should not be too difficult when we have the geo-location. Such terrain data are also freely available from terrain databases. Though, we have to pay for the more accurate ones. It could also be interesting to implement other well known propagation models, and compare the different propagation models with each other.

On the other hand, the CalculateHistory function mentioned before should be implemented in order to allow the calculation of past estimations and the creation of statistics, reports etc.

7.4 SECURITY FEATURES

In the database system implementation, security features were not taken into consideration. However, in real database systems it is essential to develop several security measures such as user authentication (through interfaces with WSDs and Other users), privacy, encryption etc. Future work should focus on developing the lacking security features of the geo-location database system.

Bibliography

[1] Qing Zhao and Brian M. Sadler, "A Survey of Dynamic Spectrum Access (signal processing, networking and regulatory policy)", IEEE Signal Processing Magazine, May 2007.

[2] Qing Zhao Ananthram Swami, "Networking Cognitive Radios for Dynamic Spectrum Access", MILCOM Tutorial San Diego, 18 November 2008.

[3] Ian F. Akyildiz, Won-Yeol Lee, Mehmet C. Vuran, Shantidev Mohanty, "Survey on Spectrum Management in Cognitive Radio Networks", IEEE Communications Magazine, 2008.

[4] Ian F. Akyildiz, Won-Yeol Lee, Kaushik R. Chowdhury, "CRAHNs: Cognitive Radio ad hoc Networks", Elsevier Science Publishers B. V., January 2009.

[5] Ian F. Akyildiz, Won-Yeol Lee, Kaushik R. Chowdhury, "Spectrum Management in Cognitive Radio ad hoc Networks", IEEE Network, July/August 2009.

[6] Anant Sahai, Shridhar Mubaraq Mishra, Rahul Tandra, "What is a Spectrum Hole and What Does it Take to Recognize One?", Proceedings of the IEEE, May 2009.

[7] Anant Sahai, Shridhar Mubaraq Mishra, Rahul Tandra, and Kristen Ann Woyach, "Cognitive Radios for Spectrum Sharing", IEEE Signal Processing Magazine, January 2009.

[8] Shridhar Mubaraq Mishra, Rahul Tandra, Anant Sahai, "Coexistence with Primary Users of Different Scales", DySPAN, 2nd IEEE International Symposium, 2007.

[9] Anant Sahai, Rahul Tandra, Shridhar Mubaraq Mishra, Niels Hoven, "Fundamental Design Tradeoffs in Cognitive Radio Systems", TAPAS'06, August 2006.

[10] Anant Sahai Niels Hoven Rahul Tandra, "Some Fundamental Limits on Cognitive Radio", University of California Berkeley.

[11] Ian F. Akyldiz, Brandon F. Lo, Ravikumar Balakrishnan, "Cooperative Spectrum Sensing in Cognitive Radio Networks: a Survey", Physical Communication, November 2010.

[12] Yunxia Chen, Qing Zhao, and Ananthram Swami, "Distributed Spectrum Sensing and Access in Cognitive Radio Networks With Energy Constraint", IEEE Transactions on Signal Processing, February 2009.

[13] Michael Maskery, Vikram Krishnamurthy, and Qing Zhao, "Decentralized Dynamic Spectrum Access for Cognitive Radios: Cooperative Design of a Non-Cooperative Game", IEEE Transactions on Communications, March 2009.

[14] CLAROS Project, "Geolocation database for white spaces searching in UHF band", Centro Tecnolóxico de Telecomunicacións de Galicia, 2010.

[15] Farpoint Group White Paper Document FPG 2010-299.1, "Rethinking Spectrum Scarcity: Database-Driven Cognitive Radio", Ashland MA, September 2010.

[16] Ofcom, "Digital Dividend: Geolocation for Cognitive Access: Discussion on using geolocation to enable licence-exempt access to the interleaved spectrum", United Kingdom, November 2009.

[17] Ofcom, "Implementing Geolocation", United Kingdom, November 2010.

[18] Ofcom, "Digital dividend: cognitive access Statement on licence-exempting cognitive devices using interleaved spectrum", United Kingdom, July 2009.

[19] FCC 11-4 MO&O, Washington D.C, January 2011.

[20] FCC 10-174 Second MO&O, Washington D.C, September 2010.

[21] FCC 11-131A1 Order, Washington D.C, January 2011.

[22] White Space Database Administrators Group WSDAG, "Workshops", Columbia Maryland, March 10 2011.

[23] FCC, "TV Bands Database Administrator Workshop", Columbia Maryland, April 2011.

[24] FCC, "TV White Spaces Databases and Database Administrators Guide, 2011.

[25] COGEU D2.1 ICT-248560, "European TV White Spaces Analysis and COGEU use-cases", March 2010.

[26] COGEU D3.1 ICT-248560, "Use-cases Analysis and TVWS System Requirements",2010.

[27] COGEU D3.2 ICT-248560, "Initial Architecture for TVWS Spectrum Sharing Systems", December 2010.

[28] COGEU D4.1 ICT-248460, "Spectrum Measurements and Anti-interference Spectrum Database Specification", October 2010.

[29] ECC Report 159, “Technical and Operational for the Possible Operation of Cognitive Radio Systems in the White Spaces of the Frequency Band 470-790 MHz”, Cardiff, January 2011.

[30] IEEE 802.11af Draft Standard for Information Technology, “Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - TV White Spaces Operation”, NY, January 2011.

[31] IEEE 802.22 Draft, “Standard for Wireless Regional Area Networks Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Policies and Procedures for Operation in the TV Bands”, NJ, December 2010.

[32] IEEE 802.22 Part 22.1, “Standard to Enhance Harmful Interference Protection for Low-Power Licensed Devices Operating in TV Broadcast Bands”, NY, November 2010.

[33] IEEE 802.22 Paper, “The First Cognitive Radio Wireless Regional Area Network Standard”, IEEE Communications Magazine, January 2009.

[34] IEEE SCC41, “Efforts and Other Activities in Standarization and Research in Cognitive and Dynamic Spectrum Access Networks”, IEEE Communications Magazine, 2010.

[35] ETSI RRS, “Status and Future Directions on Software Defined Radio and Cognitive Radio Standards”, IEEE Communications Magazine, September 2010.

[36] Comsearch, “Proposal to Be Designated As a TV Band Device Database Manager”, January 2011.

[37] Frequency Finder, “Proposal to Be Designated As a TV Band Device Database Manager”, January 2011.

[38] Google, “Proposal by Google Inc. to Provice a TV Band Device Database Management Solution”, January 2011.

[39] KBE and LS Telcom, “White Spaces Proposal by KB Enterprises and LLC and LS Telcom”, January 2011.

[40] Key Bridge Global LLC, “Proposal to Administer a TV Bands Database”, January 2011.

[41] Neustar, “Proposal for Designated TV Band Device Database Manager”, January 2011.

[42] Spectrum Bridge, “response to PN DA-09-2479 Proposals for Designated TV Band Database Manager ET Docket No. 04-186”, January 2011.

[43] Telcordia, “Comments of Telcordia Technologies Proposal Seeking to Be Designated a TV Band Device Database Manager”, January 2011.

[44] WSdb LLC, “Proposal to be Designated TVBand Device Database Manager”, 2011.

Appendix A-MySQL Code

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';


CREATE SCHEMA IF NOT EXISTS `tvws` ;

USE `tvws` ;


-- -----------------------------------------------------

-- Table `tvws`.`REGULATORIES`

-- -----------------------------------------------------

CREATE  TABLE IF NOT EXISTS `tvws`.`REGULATORIES` (

  `regulatory_id` INT NOT NULL ,

  `country` VARCHAR(45) NOT NULL ,

  `propagation_model` VARCHAR(45) NULL ,

  `validity_time` TIME NULL ,

  `sensing` TINYINT(1)  NULL ,

  `wsd_occupancy` TINYINT(1)  NULL ,

  `pixel_size` INT NULL ,

  `sensing_threshold` FLOAT NULL ,

  `tx_adjacent` FLOAT NULL ,
```

```
  `tx_non_adjacent` FLOAT NULL ,
  `tower_adjacent_ci` FLOAT NULL ,
  `tower_non_adjacent_ci` FLOAT NULL ,
  `other_adjacent_ci` FLOAT NULL ,
  `other_non_adjacent_ci` FLOAT NULL ,
  PRIMARY KEY (`regulatory_id`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `tvws`.`TV_TOWERS`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`TV_TOWERS` (
  `tower_id` INT NOT NULL ,
  `tower_name` VARCHAR(45) NOT NULL ,
  `tower_type` VARCHAR(45) NOT NULL ,
  `tower_x` FLOAT NULL ,
  `tower_y` FLOAT NULL ,
  `tower_z` FLOAT NULL ,
  `regulatory_id` INT NOT NULL ,
  PRIMARY KEY (`tower_id`) ,
  INDEX `fk_TV_TOWERS_REGULATORIES1` (`regulatory_id` ASC) ,
  CONSTRAINT `fk_TV_TOWERS_REGULATORIES1`
    FOREIGN KEY (`regulatory_id` )
    REFERENCES `tvws`.`REGULATORIES` (`regulatory_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `tvws`.`ANTENNAS`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`ANTENNAS` (
  `antenna_id` INT NOT NULL ,
  `tower_id` INT NOT NULL ,
  `gain` FLOAT NULL ,
  `direction` FLOAT NULL ,
  `open_angle` FLOAT NULL ,
  `tilt` FLOAT NULL ,
  PRIMARY KEY (`antenna_id`) ,
  INDEX `fk_ANTENNAS_TV_TOWERS` (`tower_id` ASC) ,
  CONSTRAINT `fk_ANTENNAS_TV_TOWERS`
    FOREIGN KEY (`tower_id` )
    REFERENCES `tvws`.`TV_TOWERS` (`tower_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `tvws`.`OTHER_USERS`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`OTHER_USERS` (
  `other_user_id` INT NOT NULL ,
  `user_type` VARCHAR(45) NOT NULL ,
```

```
  `regulatory_id` INT NOT NULL ,
  PRIMARY KEY (`other_user_id`) ,
  INDEX `fk_OTHER_USERS_REGULATORIES1` (`regulatory_id` ASC) ,
  CONSTRAINT `fk_OTHER_USERS_REGULATORIES1`
    FOREIGN KEY (`regulatory_id` )
    REFERENCES `tvws`.`REGULATORIES` (`regulatory_id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `tvws`.`WSDS`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`WSDS` (
  `wsd_id` INT NOT NULL ,
  `device_type` VARCHAR(45) NOT NULL ,
  `serial_number` INT NOT NULL ,
  `location_accuracy` FLOAT NULL ,
  `regulatory_id` INT NOT NULL ,
  PRIMARY KEY (`wsd_id`) ,
  INDEX `fk_WSDS_REGULATORIES1` (`regulatory_id` ASC) ,
  CONSTRAINT `fk_WSDS_REGULATORIES1`
    FOREIGN KEY (`regulatory_id` )
    REFERENCES `tvws`.`REGULATORIES` (`regulatory_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `tvws`.`CHANNELS`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`CHANNELS` (
  `channel_number` INT NOT NULL ,
  `start_freq` FLOAT NULL ,
  `end_freq` FLOAT NULL ,
  PRIMARY KEY (`channel_number`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `tvws`.`ANTENNA_OCCUPANCY`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`ANTENNA_OCCUPANCY` (
  `transmitter_id` INT NOT NULL ,
  `transmit_channel` INT NOT NULL ,
  `start_time` TIMESTAMP NOT NULL ,
  `end_time` TIMESTAMP NULL ,
  `transmit_power` FLOAT NULL ,
  `transmitter_x` FLOAT NULL ,
  `transmitter_y` FLOAT NULL ,
  `transmitter_z` FLOAT NULL ,
  PRIMARY KEY (`transmitter_id`, `transmit_channel`, `start_time`) ,
  INDEX `fk_ANTENNA_OCCUPANCY_ANTENNAS1` (`transmitter_id` ASC) ,
  INDEX `fk_ANTENNA_OCCUPANCY_CHANNELS1` (`transmit_channel` ASC) ,
```

```
  CONSTRAINT `fk_ANTENNA_OCCUPANCY_ANTENNAS1`
    FOREIGN KEY (`transmitter_id` )
    REFERENCES `tvws`.`ANTENNAS` (`antenna_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_ANTENNA_OCCUPANCY_CHANNELS1`
    FOREIGN KEY (`transmit_channel` )
    REFERENCES `tvws`.`CHANNELS` (`channel_number` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `tvws`.`OTHER_OCCUPANCY`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`OTHER_OCCUPANCY` (
  `transmitter_id` INT NOT NULL ,
  `transmit_channel` INT NOT NULL ,
  `start_time` TIMESTAMP NOT NULL ,
  `end_time` TIMESTAMP NULL ,
  `transmit_power` FLOAT NULL ,
  `transmitter_x` FLOAT NULL ,
  `transmitter_y` FLOAT NULL ,
  `transmitter_z` FLOAT NULL ,
  PRIMARY KEY (`transmitter_id`, `transmit_channel`, `start_time`) ,
  INDEX `fk_OTHER_OCCUPANCY_OTHER_USERS1` (`transmitter_id` ASC) ,
  INDEX `fk_OTHER_OCCUPANCY_CHANNELS1` (`transmit_channel` ASC) ,
```

```
  CONSTRAINT `fk_OTHER_OCCUPANCY_OTHER_USERS1`
    FOREIGN KEY (`transmitter_id` )
    REFERENCES `tvws`.`OTHER_USERS` (`other_user_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_OTHER_OCCUPANCY_CHANNELS1`
    FOREIGN KEY (`transmit_channel` )
    REFERENCES `tvws`.`CHANNELS` (`channel_number` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `tvws`.`WSD_OCCUPANCY`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`WSD_OCCUPANCY` (
  `transmitter_id` INT NOT NULL ,
  `transmit_channel` INT NOT NULL ,
  `start_time` TIMESTAMP NOT NULL ,
  `end_time` TIMESTAMP NULL ,
  `transmit_power` FLOAT NULL ,
  `transmitter_x` FLOAT NULL ,
  `transmitter_y` FLOAT NULL ,
  `transmitter_z` FLOAT NULL ,
  PRIMARY KEY (`transmitter_id`, `transmit_channel`, `start_time`) ,
  INDEX `fk_WSD_OCCUPANCY_WSDS1` (`transmitter_id` ASC) ,
  INDEX `fk_WSD_OCCUPANCY_CHANNELS1` (`transmit_channel` ASC) ,
```

```
  CONSTRAINT `fk_WSD_OCCUPANCY_WSDS1`
    FOREIGN KEY (`transmitter_id` )
    REFERENCES `tvws`.`WSDS` (`wsd_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_WSD_OCCUPANCY_CHANNELS1`
    FOREIGN KEY (`transmit_channel` )
    REFERENCES `tvws`.`CHANNELS` (`channel_number` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `tvws`.`MEASUREMENTS`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`MEASUREMENTS` (
  `device_id` INT NOT NULL ,
  `meas_channel` INT NOT NULL ,
  `meas_time` TIMESTAMP NOT NULL ,
  `meas_power` FLOAT NULL ,
  `tx_power` FLOAT NULL ,
  `noise` FLOAT NULL ,
  `meas_x` FLOAT NULL ,
  `meas_y` FLOAT NULL ,
  `meas_z` FLOAT NULL ,
  PRIMARY KEY (`device_id`, `meas_channel`, `meas_time`) ,
  INDEX `fk_MEASUREMENTS_WSDS1` (`device_id` ASC) ,
```

```
  INDEX `fk_MEASUREMENTS_CHANNELS1` (`meas_channel` ASC) ,
  CONSTRAINT `fk_MEASUREMENTS_WSDS1`
    FOREIGN KEY (`device_id` )
    REFERENCES `tvws`.`WSDS` (`wsd_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_MEASUREMENTS_CHANNELS1`
    FOREIGN KEY (`meas_channel` )
    REFERENCES `tvws`.`CHANNELS` (`channel_number` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `tvws`.`PIXELS`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`PIXELS` (
  `pixel_id` INT NOT NULL ,
  `pixel_x` FLOAT NOT NULL ,
  `pixel_y` FLOAT NOT NULL ,
  `pixel_size` INT NOT NULL ,
  `pixel_z` FLOAT NULL ,
  `population` VARCHAR(45) NOT NULL ,
  `regulatory_id` INT NOT NULL ,
  `lattitude_transl` FLOAT NULL ,
  PRIMARY KEY (`pixel_id`) ,
  INDEX `fk_PIXELS_REGULATORIES1` (`regulatory_id` ASC) ,
```

```
  CONSTRAINT `fk_PIXELS_REGULATORIES1`
    FOREIGN KEY (`regulatory_id` )
    REFERENCES `tvws`.`REGULATORIES` (`regulatory_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `tvws`.`ESTIMATIONS`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `tvws`.`ESTIMATIONS` (
  `est_channel` INT NOT NULL ,
  `max_rx_power` FLOAT NULL ,
  `max_tx_power` FLOAT NULL ,
  `validity_time` TIME NULL ,
  `start_freq` FLOAT NULL ,
  `end_freq` FLOAT NULL ,
  `algorithm` VARCHAR(45) NOT NULL ,
  `correction_factor` FLOAT NULL ,
  `pixel_id` INT NOT NULL ,
  `est_time` TIMESTAMP NOT NULL ,
  PRIMARY KEY (`est_channel`, `pixel_id`) ,
  INDEX `fk_ESTIMATIONS_CHANNELS1` (`est_channel` ASC) ,
  INDEX `fk_ESTIMATIONS_PIXELS1` (`pixel_id` ASC) ,
  CONSTRAINT `fk_ESTIMATIONS_CHANNELS1`
    FOREIGN KEY (`est_channel` )
    REFERENCES `tvws`.`CHANNELS` (`channel_number` )
```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_ESTIMATIONS_PIXELS1`
    FOREIGN KEY (`pixel_id` )
    REFERENCES `tvws`.`PIXELS` (`pixel_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;



SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Appendix B - Java Code

```
package no.telenor.naroazurutuza.main.java.dao.interfaces;

import no.telenor.naroazurutuza.main.java.model.interfaces.AntennaOccupancy;

import java.util.List;

public interface AntennaOccupancyDAO {

	public abstract void save(AntennaOccupancy transientInstance);

	public abstract void delete(AntennaOccupancy persistentInstance);

	public abstract AntennaOccupancy findById(no.telenor.naroazurutuza.main.java.model.
implementations.AntennaOccupancyId id);

	public abstract List <AntennaOccupancy> findByTransmitPower(Object transmitPower);

	public abstract List <AntennaOccupancy> findByTransmitterZ(Object transmitterZ);

	public abstract List <AntennaOccupancy> findByStartTime(Object startTime);

	public abstract List <AntennaOccupancy> findByEndTime(Object endTime);

	public abstract List <AntennaOccupancy> findAll();

}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import no.telenor.naroazurutuza.main.java.model.interfaces.Antennas;

import java.util.List;

public interface AntennasDAO {

	public abstract void save(Antennas transientInstance);

	public abstract void delete(Antennas persistentInstance);
```

```
        public abstract Antennas findById(java.lang.Integer id);

        public abstract List <Antennas> findByGain(Object gain);

        public abstract List <Antennas> findByTilt(Object tilt);

        public abstract List <Antennas> findByOpenAngle(Object openAngle);

        public abstract List <Antennas> findByDirection(Object direction);

        public abstract List <Antennas> findAll();

}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import no.telenor.naroazurutuza.main.java.model.interfaces.Channels;

public interface ChannelsDAO {

        public abstract void save(Channels transientInstance);

        public abstract void delete(Channels persistentInstance);

        public abstract Channels findById(java.lang.Integer id);

        public abstract List <Channels> findByStartFreq(Object startFreq);

        public abstract List <Channels> findByEndFreq(Object endFreq);

        public abstract List <Channels> findAll();

}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import no.telenor.naroazurutuza.main.java.model.interfaces.Estimations;

public interface EstimationsDAO {

        public abstract void save(Estimations transientInstance);

        public abstract void delete(Estimations persistentInstance);

        public abstract Estimations findById(no.telenor.naroazurutuza.main.java.model.implementations.
EstimationsId id);

        public abstract List <Estimations> findByMaxRxPower(Object maxRxPower);

        public abstract List <Estimations> findByMaxTxPower(Object maxTxPower);

        public abstract List <Estimations> findByStartFreq(Object startFreq);
```

```
		public abstract List <Estimations> findByEndFreq(Object endFreq);

		public abstract List <Estimations> findByAlgorithm(Object algorithm);

		public abstract List <Estimations> findByCorrectionFactor(Object correctionFactor);

		public abstract List <Estimations> findAll();

}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import no.telenor.naroazurutuza.main.java.model.interfaces.Measurements;

public interface MeasurementsDAO {

		public abstract void save(Measurements transientInstance);

		public abstract void delete(Measurements persistentInstance);

		public abstract Measurements findById(no.telenor.naroazurutuza.main.java.model.implementations.
MeasurementsId id);

		public abstract List <Measurements> findByMeasPower(Object measPower);

		public abstract List <Measurements> findByTxPower(Object txPower);

		public abstract List <Measurements> findByNoise(Object noise);

		public abstract List <Measurements> findByMeasZ(Object measZ);

		public abstract List <Measurements> findAll();

}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import no.telenor.naroazurutuza.main.java.model.interfaces.OtherOccupancy;

public interface OtherOccupancyDAO {

		public abstract void save(OtherOccupancy transientInstance);

		public abstract void delete(OtherOccupancy persistentInstance);

		public abstract OtherOccupancy findById(no.telenor.naroazurutuza.main.java.model.implementations.
OtherOccupancyId id);

		public abstract List <OtherOccupancy> findByTransmitPower(Object transmitPower);

		public abstract List <OtherOccupancy> findByTransmitterZ(Object transmitterZ);
```

```
        public abstract List <OtherOccupancy> findByStartTime(Object startTime);

        public abstract List <OtherOccupancy> findByEndTime(Object endTime);

        public abstract List <OtherOccupancy> findAll();

}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import  no.telenor.naroazurutuza.main.java.model.interfaces.OtherUsers;

public interface OtherUsersDAO {

        public abstract void save(OtherUsers transientInstance);

        public abstract void delete(OtherUsers persistentInstance);

        public abstract OtherUsers findById(java.lang.Integer id);

        public abstract List <OtherUsers> findByUserType(Object userType);

        public abstract List <OtherUsers> findAll();
}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import no.telenor.naroazurutuza.main.java.model.interfaces.Pixels;

public interface PixelsDAO {

        public abstract void save(Pixels transientInstance);

        public abstract void delete(Pixels persistentInstance);

        public abstract Pixels findById(java.lang.Integer id);

        public abstract List <Pixels> findByPixelX(Object pixelX);

        public abstract List <Pixels> findByPixelY(Object pixelY);

        public abstract List <Pixels> findByPixelSize(Object pixelSize);

        public abstract List <Pixels> findByPixelZ(Object pixelZ);

        public abstract List <Pixels> findByPopulation(Object population);

        public abstract List <Pixels> findAll();

}
```

```
package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import no.telenor.naroazurutuza.main.java.model.interfaces.Regulatories;

public interface RegulatoriesDAO {

        public abstract void save(Regulatories transientInstance);

        public abstract void delete(Regulatories persistentInstance);

        public abstract Regulatories findById(java.lang.Integer id);

        public abstract List <Regulatories> findByCountry(Object country);

        public abstract List <Regulatories> findByPropagationModel(Object propagationModel);

        public abstract List <Regulatories> findBySensing(Object sensing);

        public abstract List <Regulatories> findByWsdOccupancy(Object wsdOccupancy);

        public abstract List <Regulatories> findByPixelSize(Object pixelSize);

        public abstract List <Regulatories> findBySensingThreshold(Object sensingThreshold);

        public abstract List <Regulatories> findByTxAdjacent(Object txAdjacent);

        public abstract List <Regulatories> findByTxNonAdjacent(Object txNonAdjacent);

        public abstract List <Regulatories> findByTowerAdjacentCi(Object towerAdjacentCi);

        public abstract List <Regulatories> findByTowerNonAdjacentCi(Object towerNonAdjacentCi);

        public abstract List <Regulatories> findByOtherAdjacentCi(Object otherAdjacentCi);

        public abstract List <Regulatories> findByOtherNonAdjacentCi(Object otherNonAdjacentCi);

        public abstract List <Regulatories> findAll();

}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import no.telenor.naroazurutuza.main.java.model.interfaces.TvTowers;

public interface TvTowersDAO {

        public abstract void save(TvTowers transientInstance);

        public abstract void delete(TvTowers persistentInstance);

        public abstract TvTowers findById(java.lang.Integer id);
```

```
        public abstract List <TvTowers> findByTowerType(Object towerType);

        public abstract List <TvTowers> findByTowerZ(Object towerZ);

        public abstract List <TvTowers> findAll();
}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import no.telenor.naroazurutuza.main.java.model.interfaces.WsdOccupancy;

public interface WsdOccupancyDAO {

        public abstract void save(WsdOccupancy transientInstance);

        public abstract void delete(WsdOccupancy persistentInstance);

        public  abstract  WsdOccupancy  findById(no.telenor.naroazurutuza.main.java.model.implementations.
WsdOccupancyId id);

        public abstract List <WsdOccupancy> findByTransmitPower(Object transmitPower);

        public abstract List <WsdOccupancy> findByTransmitterZ(Object transmitterZ);

        public abstract List <WsdOccupancy> findByStartTime(Object startTime);

        public abstract List <WsdOccupancy> findByEndTime(Object endTime);

        public abstract List <WsdOccupancy> findAll();

}

package no.telenor.naroazurutuza.main.java.dao.interfaces;

import java.util.List;

import no.telenor.naroazurutuza.main.java.model.interfaces.Wsds;

public interface WsdsDAO {

        public abstract void save(Wsds transientInstance);

        public abstract void delete(Wsds persistentInstance);

        public abstract Wsds findById(java.lang.Integer id);

        public abstract List <Wsds> findByDeviceType(Object deviceType);

        public abstract List <Wsds> findBySerialNumber(Object serialNumber);

        public abstract List <Wsds> findByLocationAccuracy(Object locationAccuracy);

        public abstract List <Wsds> findAll();
}
```

```
package no.telenor.naroazurutuza.main.java.hibernate;

import org.hibernate.Session;

public interface HibernateSupport {

        /**
     * Returns the current Hibernate session for communicating with the
     * underlying database.
     *
     * @return the current Hibernate session.
     */
    Session getCurrentSession();

    /**
     * closes the current Hibernate session.
     */
    void closeCurrentSession();

    /**
     * Opens a transaction on the current session.
     *
     * @throws IllegalStateException if a transaction is already open. The
     *        transaction will be rolled back and the session state cleaned up.
     */
    void beginTransaction();

    /**
     * Commits the transaction on the current session.
     *
     * @throws IllegalStateException if no transaction is open.
     */
    void endTransaction();
}

package no.telenor.naroazurutuza.main.java.model.interfaces;

import java.sql.Timestamp;

import no.telenor.naroazurutuza.main.java.model.implementations.AntennaOccupancyId;

public interface AntennaOccupancy {

        public abstract AntennaOccupancyId getId();

        public abstract void setId(AntennaOccupancyId id);

        public abstract Timestamp getEndTime();

        public abstract void setEndTime(Timestamp endTime);

        public abstract Float getTransmitPower();

        public abstract void setTransmitPower(Float transmitPower);

        public abstract Float getTransmitterX();
```

```
		public abstract void setTransmitterX(Float transmitterX);
		public abstract Float getTransmitterY();
		public abstract void setTransmitterY(Float transmitterY);
		public abstract Float getTransmitterZ();
		public abstract void setTransmitterZ(Float transmitterZ);
}
package no.telenor.naroazurutuza.main.java.model.interfaces;
import java.util.Set;
public interface Antennas {
		public abstract Integer getAntennaId();
		public abstract void setAntennaId(Integer antennaId);
		public abstract TvTowers getTvTowers();
		public abstract void setTvTowers(TvTowers tvTowers);
		public abstract Float getGain();
		public abstract void setGain(Float gain);
		public abstract Float getDirection();
		public abstract void setDirection(Float direction);
		public abstract Float getOpenAngle();
		public abstract void setOpenAngle(Float openAngle);
		public abstract Float getTilt();
		public abstract void setTilt(Float tilt);
		public abstract Set getAntennaOccupancies();
		public abstract void setAntennaOccupancies(Set antennaOccupancies);
}
package no.telenor.naroazurutuza.main.java.model.interfaces;
import java.util.Set;
public interface Channels {
		public abstract Integer getChannelNumber();
```

```
	public abstract void setChannelNumber(Integer channelNumber);

	public abstract float getStartFreq();

	public abstract void setStartFreq(float startFreq);

	public abstract float getEndFreq();

	public abstract void setEndFreq(float endFreq);

	public abstract Set getOtherOccupancies();

	public abstract void setOtherOccupancies(Set otherOccupancies);

	public abstract Set getMeasurementses();

	public abstract void setMeasurementses(Set measurementses);

	public abstract Set getEstimationses();

	public abstract void setEstimationses(Set estimationses);

	public abstract Set getWsdOccupancies();

	public abstract void setWsdOccupancies(Set wsdOccupancies);

	public abstract Set getAntennaOccupancies();

	public abstract void setAntennaOccupancies(Set antennaOccupancies);

}

package no.telenor.naroazurutuza.main.java.model.interfaces;

import java.sql.Time;
import java.sql.Timestamp;

import no.telenor.naroazurutuza.main.java.model.implementations.EstimationsId;

public interface Estimations {

	public abstract EstimationsId getId();

	public abstract void setId(EstimationsId id);

	public abstract Timestamp getEstTime();

	public abstract void setEstTime(Timestamp estTime);

	public abstract float getMaxRxPower();

	public abstract void setMaxRxPower(float maxRxPower);

	public abstract float getMaxTxPower();
```

```
	public abstract void setMaxTxPower(float maxTxPower);

	public abstract Time getValidityTime();

	public abstract void setValidityTime(Time validityTime);

	public abstract float getStartFreq();

	public abstract void setStartFreq(float startFreq);

	public abstract float getEndFreq();

	public abstract void setEndFreq(float endFreq);

	public abstract String getAlgorithm();

	public abstract void setAlgorithm(String algorithm);

	public abstract float getCorrectionFactor();

	public abstract void setCorrectionFactor(float correctionFactor);

	public abstract void calculateTvws(AntennaOccupancy antennaOccupancy,Channels channel, Pixels pixel, double d);

	public abstract void calculateTvws(OtherOccupancy otherOccupancy, Channels channel, Pixels pixel, double d);

	public abstract void calculateTvws(WsdOccupancy wsdOccupancy, Channels channel, Pixels pixel, double d);

	public abstract void calculateTvws(Channels channel, Pixels pixel);
}

package no.telenor.naroazurutuza.main.java.model.interfaces;

import no.telenor.naroazurutuza.main.java.model.implementations.MeasurementsId;

public interface Measurements {

	public abstract MeasurementsId getId();

	public abstract void setId(MeasurementsId id);

	public abstract Float getMeasPower();

	public abstract void setMeasPower(Float measPower);

	public abstract Float getTxPower();

	public abstract void setTxPower(Float txPower);

	public abstract Float getNoise();

	public abstract void setNoise(Float noise);
```

```
	public abstract Float getMeasX();
	public abstract void setMeasX(Float measX);
	public abstract Float getMeasY();
	public abstract void setMeasY(Float measY);
	public abstract Float getMeasZ();
	public abstract void setMeasZ(Float measZ);
}
package no.telenor.naroazurutuza.main.java.model.interfaces;
import java.sql.Timestamp;
import no.telenor.naroazurutuza.main.java.model.implementations.OtherOccupancyId;
public interface OtherOccupancy {
	public abstract OtherOccupancyId getId();
	public abstract void setId(OtherOccupancyId id);
	public abstract Timestamp getEndTime();
	public abstract void setEndTime(Timestamp endTime);
	public abstract Float getTransmitPower();
	public abstract void setTransmitPower(Float transmitPower);
	public abstract Float getTransmitterX();
	public abstract void setTransmitterX(Float transmitterX);
	public abstract Float getTransmitterY();
	public abstract void setTransmitterY(Float transmitterY);
	public abstract Float getTransmitterZ();
	public abstract void setTransmitterZ(Float transmitterZ);
}
package no.telenor.naroazurutuza.main.java.model.interfaces;
import java.util.Set;
public interface OtherUsers {
	public abstract Integer getOtherUserId();
```

```
		public abstract void setOtherUserId(Integer otherUserId);

		public abstract Regulatories getRegulatories();

		public abstract void setRegulatories(Regulatories regulatories);

		public abstract String getUserType();

		public abstract void setUserType(String userType);

		public abstract Set getOtherOccupancies();

		public abstract void setOtherOccupancies(Set otherOccupancies);

}

package no.telenor.naroazurutuza.main.java.model.interfaces;

import java.util.Set;

public interface Pixels {

		public abstract Integer getPixelId();

		public abstract void setPixelId(Integer pixelId);

		public abstract Regulatories getRegulatories();

		public abstract void setRegulatories(Regulatories regulatories);

		public abstract float getPixelX();

		public abstract void setPixelX(float pixelX);

		public abstract float getPixelY();

		public abstract void setPixelY(float pixelY);

		public abstract Integer getPixelSize();

		public abstract void setPixelSize(Integer pixelSize);

		public abstract float getPixelZ();

		public abstract void setPixelZ(float pixelZ);

		public abstract float getLatTrans();

		public abstract void setLatTrans(float latTrans);

		public abstract String getPopulation();

		public abstract void setPopulation(String population);

		public abstract Set getEstimationses();
```

```
	public abstract void setEstimationses(Set estimationses);

	public abstract void calculateTvws(Set tvTowers, Set otherUsers, Set Wsds, Set channels);
}

package no.telenor.naroazurutuza.main.java.model.interfaces;

import java.sql.Time;
import java.util.Set;

public interface Regulatories {

	public abstract Integer getRegulatoryId();

	public abstract void setRegulatoryId(Integer regulatoryId);

	public abstract String getCountry();

	public abstract void setCountry(String country);

	public abstract String getPropagationModel();

	public abstract void setPropagationModel(String propagationModel);

	public abstract Time getValidityTime();

	public abstract void setValidityTime(Time validityTime);

	public abstract Boolean getSensing();

	public abstract void setSensing(Boolean sensing);

	public abstract Boolean getWsdOccupancy();

	public abstract void setWsdOccupancy(Boolean wsdOccupancy);

	public abstract Integer getPixelSize();

	public abstract void setPixelSize(Integer pixelSize);

	public abstract Float getSensingThreshold();

	public abstract void setSensingThreshold(Float sensingThreshold);

	public abstract Float getTxAdjacent();

	public abstract void setTxAdjacent(Float txAdjacent);

	public abstract Float getTxNonAdjacent();

	public abstract void setTxNonAdjacent(Float txNonAdjacent);

	public abstract Float getTowerAdjacentCi();

	public abstract void setTowerAdjacentCi(Float towerAdjacentCi);
```

```
	public abstract Float getTowerNonAdjacentCi();

	public abstract void setTowerNonAdjacentCi(Float towerNonAdjacentCi);

	public abstract Float getOtherAdjacentCi();

	public abstract void setOtherAdjacentCi(Float otherAdjacentCi);

	public abstract Float getOtherNonAdjacentCi();

	public abstract void setOtherNonAdjacentCi(Float otherNonAdjacentCi);

	public abstract Set getPixelses();

	public abstract void setPixelses(Set pixelses);

	public abstract Set getOtherUserses();

	public abstract void setOtherUserses(Set otherUserses);

	public abstract Set getWsdses();

	public abstract void setWsdses(Set wsdses);

	public abstract Set getTvTowerses();

	public abstract void setTvTowerses(Set tvTowerses);

	public abstract void calculateTvws(Set channels);

	public abstract Set showTvws(float x, float y);

}

package no.telenor.naroazurutuza.main.java.model.interfaces;

import java.util.Set;

public interface TvTowers {

	public abstract Integer getTowerId();

	public abstract void setTowerId(Integer towerId);

	public abstract Regulatories getRegulatories();

	public abstract void setRegulatories(Regulatories regulatories);

	public abstract String getTowerName();

	public abstract void setTowerName(String towerName);

	public abstract String getTowerType();

	public abstract void setTowerType(String towerType);
```

```
	public abstract Float getTowerX();
	public abstract void setTowerX(Float towerX);
	public abstract Float getTowerY();
	public abstract void setTowerY(Float towerY);
	public abstract Float getTowerZ();
	public abstract void setTowerZ(Float towerZ);
	public abstract Set getAntennases();
	public abstract void setAntennases(Set antennases);
}
package no.telenor.naroazurutuza.main.java.model.interfaces;
import java.sql.Timestamp;
import no.telenor.naroazurutuza.main.java.model.implementations.WsdOccupancyId;
public interface WsdOccupancy {
	public abstract WsdOccupancyId getId();
	public abstract void setId(WsdOccupancyId id);
	public abstract Timestamp getEndTime();
	public abstract void setEndTime(Timestamp endTime);
	public abstract Float getTransmitPower();
	public abstract void setTransmitPower(Float transmitPower);
	public abstract Float getTransmitterX();
	public abstract void setTransmitterX(Float transmitterX);
	public abstract Float getTransmitterY();
	public abstract void setTransmitterY(Float transmitterY);
	public abstract Float getTransmitterZ();
	public abstract void setTransmitterZ(Float transmitterZ);
}
package no.telenor.naroazurutuza.main.java.model.interfaces;
import java.util.Set;
```

```
public interface Wsds {

	public abstract Integer getWsdId();

	public abstract void setWsdId(Integer wsdId);

	public abstract Regulatories getRegulatories();

	public abstract void setRegulatories(Regulatories regulatories);

	public abstract String getDeviceType();

	public abstract void setDeviceType(String deviceType);

	public abstract Integer getSerialNumber();

	public abstract void setSerialNumber(Integer serialNumber);

	public abstract Float getLocationAccuracy();

	public abstract void setLocationAccuracy(Float locationAccuracy);

	public abstract Set getMeasurementses();

	public abstract void setMeasurementses(Set measurementses);

	public abstract Set getWsdOccupancies();

	public abstract void setWsdOccupancies(Set wsdOccupancies);

}

package no.telenor.naroazurutuza.main.java.service;

import no.telenor.naroazurutuza.main.java.model.interfaces.*;
import no.telenor.naroazurutuza.main.java.dao.interfaces.*;

import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.ClassPathResource;

import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;

/**
 * Service layer. It provides the services and control for the TVWS database system.
 */

public class TvwsControl {

	private static Regulatories regulatory;
	private static RegulatoriesDAO regulatoryDAO;
	private static ChannelsDAO channelDAO;
	private static Set estimations;
```

```
//calculate TVWS, makes the calculations for an input regulatory
public static void calculateTvws(Integer regulatoryId){

		BeanFactory beanFactory = new XmlBeanFactory(new ClassPathResource(
		"applicationContext.xml"));
		regulatory =(Regulatories)beanFactory.getBean("Regulatories");
		regulatoryDAO=(RegulatoriesDAO)beanFactory.getBean("RegulatoriesDAO");

		//get the regulatory information
		regulatory=regulatoryDAO.findById((Integer)regulatoryId);

		if(regulatory==null)
			System.out.println("Regulatory not found");
		else{
			//get the channels for which estimations have to be done
			channelDAO=(ChannelsDAO)beanFactory.getBean("ChannelsDAO");

			Set channels=new HashSet(channelDAO.findAll());
			if (channels.size()==0)
				System.out.println("No channels exist for calculations");
			else{
				//calculate the tvwss
				regulatory.calculateTvws(channels);
			}
		}
}

//Show TVWSs for an input location
public static Set showTvws(Integer regulatoryId, Float x, Float y){

		Set estimations=new HashSet(0);

		BeanFactory beanFactory = new XmlBeanFactory(new ClassPathResource(
		"applicationContext.xml"));
		regulatory =(Regulatories)beanFactory.getBean("Regulatories");
		regulatoryDAO=(RegulatoriesDAO)beanFactory.getBean("RegulatoriesDAO");

		regulatory=regulatoryDAO.findById((Integer)regulatoryId);

		if (regulatory==null)
			System.out.println("Not regulatory found");
		else{
			//load the estimations for the location
			estimations=regulatory.showTvws(x, y);
			if (estimations.size()==0)
				System.out.println("No results available for input location");
		}
		return estimations;
}

/*calculate history of estimations. for efficiency just most recent estimatinos
 * are saved. this functions calculates the estimations over the history
 */
public void calculateHistory(){

}
```

```
	public static void main(String[] args) {
		//calculateTvws(485323);
		System.out.println("calculations done");

		Estimations estimation;

		estimations= showTvws(485323,(float)7.6878,(float)59.9166);
		Estimations estimationses[]=new Estimations[estimations.size()];
		for(int i=0;i<estimations.size();i++){
			BeanFactory beanFactory=new XmlBeanFactory(new ClassPathResource(
			"applicationContext.xml"));
			estimationses[i]= (Estimations)beanFactory.getBean("Estimations");
		}

		Iterator it1 = estimations.iterator();
		while(it1.hasNext()){
			estimation=(Estimations)it1.next();
			int number = estimation.getId().getChannels().getChannelNumber()-21;
			estimationses[number]=estimation;

		}
		for (int i=0;i<estimations.size();i++){
			System.out.println("Estimations readed for X=10.6878 and Y=59.9166 
[propagationModel="
		+ estimationses[i].getAlgorithm() + ", rxpower="
		  + estimationses[i].getMaxRxPower() + ", txpower=" + estimationses[i].getMaxTxPower()+ 
", channel=" + estimationses[i].getId().getChannels().getChannelNumber()+ ", time=" + estimationses[i].
getEstTime()
		+ ", correctionFactor=" + estimationses[i].getCorrectionFactor()+ ", pixelX=" + estimationses[i].
getId().getPixels().getPixelX()+ ", pixelY=" + estimationses[i].getId().getPixels().getPixelY());
		}
		if(estimations.size()==0){
			System.out.println("No pixel found X=9.6878 and Y=50.9166");
		}

	}
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<beans
	xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xmlns:p="http://www.springframework.org/schema/p"
	xsi:schemaLocation="http://www.springframework.org/schema/beans	http://www.springframework.
org/schema/beans/spring-beans-3.0.xsd">


	<bean id="hibernateSupport"
	class="no.telenor.naroazurutuza.main.java.hibernate.DefaultHibernateSupport"
		init-method="init" destroy-method="destroy">
	</bean>

	<bean id="AntennasDAO"
		class="no.telenor.naroazurutuza.main.java.dao.implementations.AntennasDAOImpl">
		<property name="hibernateSupport">
```

```
                    <ref bean="hibernateSupport" />
          </property>
     </bean>
     <bean id="ChannelsDAO"
          class="no.telenor.naroazurutuza.main.java.dao.implementations.ChannelsDAOImpl">
          <property name="hibernateSupport">
                    <ref bean="hibernateSupport" />
          </property>
     </bean>
     <bean id="EstimationsDAO"
          class="no.telenor.naroazurutuza.main.java.dao.implementations.EstimationsDAOImpl">
          <property name="hibernateSupport">
                    <ref bean="hibernateSupport" />
          </property>
     </bean>
     <bean id="MeasurementsDAO"
          class="no.telenor.naroazurutuza.main.java.dao.implementations.MeasurementsDAOImpl">
          <property name="hibernateSupport">
                    <ref bean="hibernateSupport" />
          </property>
     </bean>
     <bean id="OtherOccupancyDAO"
          class="no.telenor.naroazurutuza.main.java.dao.implementations.
OtherOccupancyDAOImpl">
          <property name="hibernateSupport">
                    <ref bean="hibernateSupport" />
          </property>
     </bean>
     <bean id="OtherUsersDAO"
          class="no.telenor.naroazurutuza.main.java.dao.implementations.OtherUsersDAOImpl">
          <property name="hibernateSupport">
                    <ref bean="hibernateSupport" />
          </property>
     </bean>
     <bean id="PixelsDAO"
          class="no.telenor.naroazurutuza.main.java.dao.implementations.PixelsDAOImpl">
          <property name="hibernateSupport">
                    <ref bean="hibernateSupport" />
          </property>
     </bean>
     <bean id="RegulatoriesDAO"
          class="no.telenor.naroazurutuza.main.java.dao.implementations.RegulatoriesDAOImpl">
          <property name="hibernateSupport">
                    <ref bean="hibernateSupport" />
          </property>
     </bean>
     <bean id="TvTowersDAO"
          class="no.telenor.naroazurutuza.main.java.dao.implementations.TvTowersDAOImpl">
          <property name="hibernateSupport">
                    <ref bean="hibernateSupport" />
          </property>
     </bean>
     <bean id="WsdOccupancyDAO"
          class="no.telenor.naroazurutuza.main.java.dao.implementations.WsdOccupancyDAOImpl">
          <property name="hibernateSupport">
                    <ref bean="hibernateSupport" />
```

```
			</property>
		</bean>
		<bean id="WsdsDAO"
			class="no.telenor.naroazurutuza.main.java.dao.implementations.WsdsDAOImpl">
			<property name="hibernateSupport">
				<ref bean="hibernateSupport" />
			</property>
		</bean>
		<bean id="AntennaOccupancyDAO"
			class="no.telenor.naroazurutuza.main.java.dao.implementations.
AntennaOccupancyDAOImpl">
			<property name="hibernateSupport">
				<ref bean="hibernateSupport" />
			</property>
		</bean>
		<bean id="Antennas"
			class="no.telenor.naroazurutuza.main.java.model.implementations.AntennasImpl">
		</bean>
		<bean id="Channels"
			class="no.telenor.naroazurutuza.main.java.model.implementations.ChannelsImpl">
		</bean>
		<bean id="Estimations"
			class="no.telenor.naroazurutuza.main.java.model.implementations.EstimationsImpl">
		</bean>
		<bean id="Measurements"
			class="no.telenor.naroazurutuza.main.java.model.implementations.MeasurementsImpl">
		</bean>
		<bean id="OtherOccupancy"
			class="no.telenor.naroazurutuza.main.java.model.implementations.OtherOccupancyImpl">
		</bean>
		<bean id="OtherUsers"
			class="no.telenor.naroazurutuza.main.java.model.implementations.OtherUsersImpl">
		</bean>
		<bean id="Pixels"
			class="no.telenor.naroazurutuza.main.java.model.implementations.PixelsImpl">
		</bean>
		<bean id="Regulatories"
			class="no.telenor.naroazurutuza.main.java.model.implementations.RegulatoriesImpl">
		</bean>
		<bean id="TvTowers"
			class="no.telenor.naroazurutuza.main.java.model.implementations.TvTowersImpl">
		</bean>
		<bean id="WsdOccupancy"
			class="no.telenor.naroazurutuza.main.java.model.implementations.WsdOccupancyImpl">
		</bean>
		<bean id="Wsds"
			class="no.telenor.naroazurutuza.main.java.model.implementations.WsdsImpl">
		</bean>
		<bean id="AntennaOccupancy"
			class="no.telenor.naroazurutuza.main.java.model.implementations.
AntennaOccupancyImpl">
		</bean></beans>
```

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
        "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<!-- Generated by MyEclipse Hibernate Tools.                   -->
<hibernate-configuration>

<session-factory>
        <property name="dialect">org.hibernate.dialect.MySQLInnoDBDialect</property>
        <property name="connection.url">
                jdbc:mysql://localhost/tvws
        </property>
        <property name="connection.username">root</property>
        <property name="connection.password">narleras</property>
        <property name="connection.driver_class">
                com.mysql.jdbc.Driver
        </property>
        <property name="myeclipse.connection.profile">
                MySQL Connector/J
        </property>

        <mapping resource="Pixels.hbm.xml" />
        <mapping resource="AntennaOccupancy.hbm.xml" />
        <mapping resource="Antennas.hbm.xml" />
        <mapping resource="Estimations.hbm.xml" />
        <mapping resource="Measurements.hbm.xml" />
        <mapping resource="OtherOccupancy.hbm.xml" />
        <mapping resource="OtherUsers.hbm.xml" />
        <mapping resource="TvTowers.hbm.xml" />
        <mapping resource="WsdOccupancy.hbm.xml" />
        <mapping resource="Wsds.hbm.xml" />
        <mapping resource="Regulatories.hbm.xml" />
        <mapping resource="Channels.hbm.xml" />

</session-factory>

</hibernate-configuration>
```

Appendix C-Web Interface

```
<%@ page language=”java” import=”java.util.*” pageEncoding=”ISO-8859-1”%>
<%
String path = request.getContextPath();
String basePath = request.getScheme()+”://”+request.getServerName()+”:”+request.getServerPort()+path+”/”;
%>

<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01 Transitional//EN”>
<html>
  <head>
    <base href=”<%=basePath%>”>

    <title>TVWS Finder home page</title>
            <meta http-equiv=”pragma” content=”no-cache”>
            <meta http-equiv=”cache-control” content=”no-cache”>
            <meta http-equiv=”expires” content=”0”>
            <meta http-equiv=”keywords” content=”keyword1,keyword2,keyword3”>
            <meta http-equiv=”description” content=”This is my page”>

            <!--
            <link rel=”stylesheet” type=”text/css” href=”style.css”>
            -->

  </head>
  <body>
  <br>           
              <font face=”HelveticaNeueLT Std Med”> <strong></strong></font><font
color=”#d89c21”><strong><font face=”HelveticaNeueLT Std Lt” size=”7”><font style=”font-weight:
normal;”>    TVWS Finder</font></font></strong></font><br>
               
       <font size=”2” color=”#808080”> </font><font face=”HelveticaNeueLT Std Thin” size=”3”
color=”#808080”>   <font face=”HelveticaNeueLT Std” size=”2”>   
         By Naroa Zurutuza for NTNU and Telenor</font></
font><br><br>
  <hr width=”100%” size=”2”><br><br>
  <form method=”post” action=”/tvwsFinder4/TvwsFinder” name=”Find TVWSs”>
   
     <font face=”HelveticaNeueLT Std Lt”>           
             X coordinate:</font> <input type=”text”
name=”xCoord”>         <font face=”HelveticaNeueLT Std Lt”> 
```

```
Y coordinate: </font><input type=”text” name=”yCoord”>         
          
        <font size=”7”><input type=”submit” value=”Find TVWS” name=”find”></font>  
     
  <br>
      
      <font size=”7”><input type=”radio” value=”calculate” name=”calculate”></font> <font size=”1”
face=”HelveticaNeueLT Std Lt”>Update calculations before showing results</font>
                 <br></form><br>    
;                             
   <img border=”0” src=”oslo_map.jpg” width=”620” height=”300”>

                   
                    
            <br>      
         </body>
</html>
```

```
package no.telenor.naroazurutuza.main.java.service;

import java.io.*;
import javax.servlet.*;

import java.sql.Timestamp;
import java.util.Iterator;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import no.telenor.naroazurutuza.main.java.model.interfaces.Estimations;

import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.ClassPathResource;

public class TvwsServlet extends HttpServlet {

		Integer regulatory;
		Float xCoord;
		Float yCoord;
		String power[];
		String algorithm;
		String message=null;
		//@Override
		public void init()throws ServletException{
				super.init();

		}
		public void destroy(){

		}
		protected void service(HttpServletRequest req, HttpServletResponse resp)
						throws ServletException, IOException {
						super.service(req, resp);
```

```
	}
	protected void doPost(HttpServletRequest req, HttpServletResponse resp)
				throws ServletException, IOException {

		boolean error = false;
		TvwsControl tvwsControl = new TvwsControl();
		Estimations estimation;

		regulatory=485323;
		/*if (req.getParameter("regulatory")!=null){
			regulatory= Integer.parseInt(req.getParameter("regulatory"));
		}
		else
			error = true;*/
		if(req.getParameter("calculate")!=null){
			if(req.getParameter("calculate").equals("calculate")){
				tvwsControl.calculateTvws(regulatory);
				//System.out.println(req.getParameter("calculate"));
			}
		}

		if (req.getParameter("xCoord")!=null){
			xCoord = Float.parseFloat(req.getParameter("xCoord"));
		}
		else
			error = true;
		if (req.getParameter("yCoord")!=null){
			yCoord = Float.parseFloat(req.getParameter("yCoord"));
		}
		else
			error = true;
		if(!error)
			{

			tvwsControl.showTvws(regulatory, xCoord, yCoord);
			//Estimations estimationses[]=new Estimations[tvwsControl.getEstimations().
size()];
			/*for(int i=0;i<tvwsControl.getEstimations().size();i++){
				BeanFactory beanFactory=new XmlBeanFactory(new
ClassPathResource(
				"applicationContext.xml"));
				estimationses[i]= (Estimations)beanFactory.getBean("Estimations");
			}*/
			//power=new String[tvwsControl.getEstimations().size()];
			power=new String[48];
			String algorithms[]=new String[tvwsControl.getEstimations().size()];
			String times[]=new String[tvwsControl.getEstimations().size()];
			Float rx[]=new Float[tvwsControl.getEstimations().size()];
			Float tx[]=new Float[tvwsControl.getEstimations().size()];
			String validity[]=new String[tvwsControl.getEstimations().size()];
			Integer size = tvwsControl.getEstimations().size();
			String length = size.toString();
			Timestamp timestamp = new Timestamp (0);;
			//System.out.println(channels.length);

			if(tvwsControl.getEstimations().size()!=0){
```

```
                        Iterator it1 = tvwsControl.getEstimations().iterator();
                        while(it1.hasNext()){
                                estimation=(Estimations)it1.next();
                                int     number     =     estimation.getId().getChannels().
getChannelNumber()-21;
                                Float pow = estimation.getMaxTxPower();
                                power[number]=pow.toString();
                                System.out.println(power[number]);
                                algorithms[number]=estimation.getAlgorithm();
                                times[number]=estimation.getEstTime().toString();
                                validity[number]=estimation.getValidityTime().toString();
                                rx[number]=estimation.getMaxRxPower();
                                tx[number]=estimation.getMaxTxPower();
                                //estimation.g
                                timestamp = estimation.getEstTime();
                        }
                        message = "Maximum transmit power in dBs for X = "+ xCoord +"Y = "
+ yCoord + "using Hata model Estimation time:"+timestamp;
                }
                else{
                        for(int i=0; i<48;i++){
                                Integer number = 0;
                                power[i]=number.toString();

                        }
                        message = "No pixel found";
                }


                req.setAttribute("power", power);
                req.setAttribute("message", message);

                //req.setAttribute("length", length);

                req.getRequestDispatcher("/responser.jsp").forward(req, resp);
        }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
        xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <display-name></display-name>
  <servlet>
        <display-name>JAX-RS REST Servlet</display-name>
        <servlet-name>JAX-RS REST Servlet</servlet-name>
        <servlet-class>
                com.sun.jersey.spi.container.servlet.ServletContainer
        </servlet-class>
        <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
        <servlet-name>JAX-RS REST Servlet</servlet-name>
        <url-pattern>/services/*</url-pattern>
```

```
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
        <description></description>
        <display-name>TvwsFinder</display-name>
        <servlet-name>TvwsFinder</servlet-name>
        <servlet-class>
        no.telenor.naroazurutuza.main.java.service.TvwsServlet
        </servlet-class>
        </servlet>
        <servlet-mapping>
        <servlet-name>TvwsFinder</servlet-name>
        <url-pattern>/TvwsFinder</url-pattern>
        </servlet-mapping>
</web-app>
```