3D Wireless Networks Simulator
- visualization of Radio Frequency propagation
for WLANs

A dissertation submitted to the
University of Dublin, Trinity College,
in a partial fulfilment of the requirements for the degree of
Master of Science in Computer Science

Przemyslaw Madej

May 2006

DECLARATION

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work.

Przemyslaw Madej,

May 31, 2006

PERMISSION TO LEND AND/OR COPY

I, the undersigned, agree that the Trinity College Library may lend and/or copy this thesis upon request.

Przemyslaw Madej,

May 31, 2006

ACKNOWLEDGEMENTS

Many thanks are due to my supervisor, Dave Lewis, for his invaluable advice and for the benefit of his experience throughout the course of the project.

To my ubicomp classmates, thanks for making last years to remember, especially to Maciej Wieckowski for continued support and encouragement.

Lastly, I would like to thank my family for their continued support over the course of my school and college years.

Przemyslaw Madej

University of Dublin, Trinity College

May 2006

ABSTRACT

The entire process of designing the wireless system still remains the big challenge. Prediction of signal propagation in different environments is essential in wireless network planning. The main goal of propagation modelling is to determine the probability of acceptable performance of a system based on radio signal propagation. If the results of the modelling are much different form reality it can dramatically increase costs, in the worst case the entire network must be redesigned and rebuilt or simply when constraints of the model were to restrictive some elements can overlap their functionality.

Moreover, current research projects within the ubiquitous computing clearly presents demand for suitable test environments. Here, modelling and visualisation of the radio wave signal propagation is one of the many features that could make life easier. Especially in the area of development where the evaluation of the pervasive environments is always limited by common set of problems involving cost and logistics of implementation of such systems.

This dissertation describes a simulator called WiFi Simulator, a unique combination of a wireless system planning tool and a popular game, that has been developed to support engineers, network administrators, researches and regular wireless system users testing and analysing radio wave signal propagation in a virtual 3D environment. Based on a 3D games engine, the simulator has been designed to maximise usability and flexibility while minimising working knowledge of the game engine. The primary focuses of the research are visualization aspects of data representation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

This dissertation describes a simulator called WiFi Simulator that has been developed to support engineers, network administrators, researches and regular wireless system users testing and analysing radio wave signal propagation in a virtual 3D environment. The primary focuses of the research are visualization aspects of data representation. The graphics sound and network connection features of a game engine have been exploited to support the core simulator.

This chapter is intended as a precursor to the main dissertation. In order to fully understand the dissertation research, design and implementation it is felt that an introduction to the main technologies involved is essential.

1.1 Why to model signal propagation?

The entire process of designing the wireless system still remains the big challenge. Prediction of signal propagation in different environments is essential in wireless network planning. The main goal of propagation modelling is to determine the probability of acceptable performance of a system based on radio signal propagation. If the results of the modelling are much different form reality it can dramatically increase costs, in the worst case the entire network must be redesigned and rebuilt or simply when constraints of the model were to restrictive some elements can overlap their functions.

In general, the fundamental task of modelling is to predict the location of the base stations carefully, so they will build the coverage area adequately to our needs. The only problem is how many of base stations we will have to use and how we will place them to minimise the cost of the undertaking.

Unfortunately, the problem is not straight. It depends on the many different factors both on the construction site and equipment used. Due to too many different parameters, that often change, simple approximations are not enough. Modern buildings are built with a use of different types of materials, later equipped with various types of office furniture which all have a huge impact on performance. The attenuation, reflections, multipath and many other different phenomena make prediction a very sophisticated process. It is too much time consuming to conduct experiments on the spot. Even the process of measuring costs money and time. Of course there is always a possibility to place a lot of strong base stations that will for sure cover needed area, but this is not the proper solution for this problem. In this case a tool that is able do the job in short time is needed.

The ideal solution would be a program that can provide a very good design on the spot. It could be even a handheld device or application that we can run on our laptop which can produce approximate results. Even if the answer contains small errors or is not detailed enough, could be used as a guidelines in the further work and create an image how the propagation in particular scenario could look like.

1.2 3D games in this context

The 3D games market is a rapidly developing area of a game industry. Companies all over the world invest huge amount of money to develop the newest game environment that will be capable of recreation of the surrounding world on the regular personal computer. Realistic virtual environment is a prime factor of developing a First Person Shooters (FPS) graphics engine, where the user impersonates main character of the game. New graphics hardware provided new capabilities, allowing new engines to add various novel effects, such as particle effects, fog, coloured lightning, as well as increase texture and polygon detail. Many games featured large outdoor environments, vehicles, advanced physics and many more.

Additionally, according to games developers a new era of game engines can be expected within the end of year 2006. New games based on the newest engines will likely to include some of the technology showcased in existing technology demos, including realistic shader-based materials with predefined physics, environments with objects (vegetation, debris, human made objects such as books or tools) universally destructible and interactive levels, procedural animation, cinematographic effects (depth of field, motion blur, etc.), realistic lighting and shadowing. John Carmack, the lead programmer for id Software [1], has repeatedly stated his opinion that it will likely be possible by year 2010 to do a real-time video-realistic rendering of a static real-world-like environment. A foretaste of the look of the newest technologies is visible on the Figure 1, where the real-time rendered scene from the Unreal 3 game engine [2] is presented.

Fig. 1 Screenshot form demo of Unreal 3 game engine; source: Epic Games Inc.

As a result, when such a powerful virtual reality tools are provided why do not use them in purpose of academic research and creation of realistic simulator of our world. The general aim in this case would be use one of existing 3D graphics game engines and then exploit it to provide a realistic user experience, and later on change the source code to provide an easily configurable test-bed for researchers. In addition, the multi-player style implementation of these games provides potential for multiple researchers to interact in a single experiment.

1.3 Vision

Before pursuing any course of research it is necessary to clearly define research goals and discuss the motivations behind such goals. The main objective of this project is to develop a fully operational simulator for RF signal propagation. In other words, an application that will allow testing the setup of wireless RF devices under different circumstances in realistic form before it will be introduced into real life. The application should be capable of carrying out real-life simulation in three dimensions virtual world. Simulation should represent world on different levels: from invisible physics to visual 3D representation of the specific objects. This approach allows conducting experiments in a simulated environment with a technique that is closest one to the real world representation developed for computer systems. When it comes to environments modelled itself, the area of simulation will be limited to multi-level indoor environments only.

Probably the most challenging part of developing the application will be a way of visualisation of the RF signal propagation. This visual effect needs to meet specific requirements. First of all it should give a necessary feedback to the user and stay in the same time non-disorientating for other simple user activities like moving around. What is more it should very intuitive, so the user will associate its functionality being partially unaware of it.

Furthermore, the entire system should be capable of recognising different types of dynamic changes of the surrounding environment, from simple changes like opening doors to chain-reaction consequences. Because of the nature of problem, prediction results should be produced in a real time and every kind of interaction between user and the environment should be immediately visible. Only in this way user will be able to receive demanded feedback. What is more, the ubiquitous character of the system does not allow for delegating job to remote objects, what in effect will cause unnecessary delays. All results should be produced on the spot.

In addition to all features previously described the ‘multi-user’ functionality should not be omitted. The client-sever approach will allow to make the most of the system. In the same time a various number of users will be able to fully explore possibilities of the simulator.

What is more, entire system should be easily used as a plug-in or extension to existing context-aware adaptive systems. As a result an entire functionality could be straightforwardly reused in other applications.

Research will not involve a development or implementation of any propagation algorithms. For development purposes one of the existing propagation tools will be used.

In conclusion, the main research objective of this dissertation is to build fully operational 3D simulator of RF signal propagation. The primary focuses are visualization aspects of data representation.

1.3.1 Potential uses of the platform

The key reason that stands for usage of the system is a prediction of signal propagation in different environments. This task is essential in wireless network planning. That is why we can divide the application of the system into four areas:

pre-design: where user wants to see the possible wave propagation within area of investigation before actual design, where results do not have to be accurate, but give an overall idea of the problem;

radio planning: where user is designing and planning actual setup of base stations;

optimisation: where user wants to correct and optimise existing configuration of wireless system;

and finally **evaluation**, which is more connected just to observation and analysis of existing systems. It gives real-time feedback about advantages and disadvantages of existing wireless layout.

All of them are very flexible and overlap between each other, and even in some cases are hard to distinguish. Within previously described functions user is able to conduct radio coverage analysis, check the strength of the signal within specific areas, check service availability and compare measured and predicted signal levels. In addition, extra features could be implemented in order to visualise less relevant information such as packet loss, accuracy of triangulating location reporting or assessing the impact of the above with a particular mobile application. Some of them may require integration with other environments like context aware systems.

1.3.2 Research objectives

The objective of the project was to develop a simulator to satisfy the following objectives:

- Allow to conduct simulations in a realistic 3D environment.
- Allow to conduct simulations of wave propagation within multi-floor indoor environments.
- Visualise RF signal propagation with use of special 3D graphic effects (e.g. fog or mist)
- Allow for interaction with the surrounding environment and be capable of dynamic changes within it.
- Must be usable, in particular, this means a straightforward initial setup procedure and an easy mechanism to configure and run simulations and produce results on the spot.
- Allow for a ‘multi-client’ approach (client-server architecture)
- Allow for a possible integration with other applications, like context aware systems

1.4 Document structure

Chapter 2: State of The Art

Chapter 2 outlines the current state of research in the fields of existing simulators and planning tools for indoor wireless systems. The second section of the chapter provides the reasoning behind supporting WiFi simulator with a game engine. This section also describes the choice of the specific game engine. The final section of the chapter introduces ubiquitous computing simulator TATUS that has been developed to support research and development of adaptive software for ubiquitous computing environments, which strongly influences this research.

Chapter 3: Design

The second part of the chapter discusses the design of the simulator. The chapter is divided accordingly to the developed components of the system

Chapter 4: Implementation

Chapter 4 explains how components of the system are implemented by modifying the game engine. The chapter first presents the relationships between system components followed by the implementational details behind each system element.

Chapter 5: Results, Evaluation & Discussion

Chapter 5 presents results, evaluation and discussion of developed simulator. The first part of this chapter is connected to the evaluation of the core simulator, where the second contains an evaluation of the separate application responsible for converting different formats of the maps used in simulations.

Chapter 6: Further Work & Conclusions

Chapter 6 presents ideas for future development of the simulator followed by conclusions about the success of its design and implementation.

Chapter 2: The State of The Art

The purpose of this chapter is to outline the current state of research in the fields of existing simulators and planning tools for indoor wireless systems. Each section presents the research conducted for this dissertation and includes an evaluation and general conclusions. Briefly describe the main functions and differences that distinguish them from the others. In the end it summarises the problem and presents the most important conclusions. All information in this chapter were gathered from white papers, documentation and other descriptions and are based on own experiments and tests on a demo/evaluation versions provided by producers of the software.

2.1 Indoor propagation

Understanding of indoor propagation of electromagnetic waves is vital to design fully operational and effective wireless network. Because of significantly differences from the typical outdoor environment, planning the indoor use of wireless systems became one of the biggest design challenges, called even by the most of the RF engineers as a ‘Black Art’.

Simulating indoor propagation is complicated due to many different factors, which are not present in outdoor environments or simply the scale allows us to omit them. The main problem is an incredibly large variability in building layout and materials used while building. Moreover, the simplest indoor space can change drastically by small things like movement of people, doors opening and closing, even smallest changes in a décor. In this chapter I will try to present the basics of modelling the indoor propagation.

2.1.1 Basic radio propagation

Electromagnetic wave propagation is described by Maxwell’s equations, which state that a changing magnetic field produces an electric field and a changing electric field produces a magnetic field. Thus electromagnetic waves are able to self-propagate. In free space, radio waves emanate from a point source of radio energy in all directions creating in a result a spherical wave front. Unfortunately, free space model is only an idealized model which is hard to apply in a real world. Generally, propagation is impaired by proximity to the earth, any obstacles in line-of-sight (LOS) and even atmospheric effects.

Contemporary communication systems take an advantage of the three following phenomenon to communicate: reflection, diffraction and scattering. Those three basic mechanisms cause radio signal distortions which makes that signal becomes stronger or lead to propagation losses. What is more, in the real life they create additional radio propagation paths beyond the direct optical "line of sight" path between the radio transmitter and receiver. Following terms are described in [3]

“Reflection: Whenever an electromagnetic wave is incident on a smooth surface (or certain sharp edges), a portion of the wave will be reflected. This reflection can be thought of as specular, where the grazing angle and reflection angle are equal.

Diffraction: Diffraction occurs when the path of an electromagnetic wave is blocked by an obstacle with a relatively sharp edge (as compared to the wavelength of the wave). The effect of diffraction is to fill in the shadow that is generated by the blockage.

Scattering: Scattering occurs when an electromagnetic wave is incident on a rough or irregular surface. When a wave is scattered, the resulting reflections occur in many different directions. When looked at on a small scale, the surface can often be analyzed as a collection of flat or sharp reflectors.”

2.1.2 Indoor environment

In case of indoor environments multipath is a primary factor. The signal propagated indoor is more likely to come across obstacles what causes propagation impairment. Furthermore, the line-of-sight path usually does not exist and the characteristics of the environment can change rapidly. Of course the required range is completely different. Here, only short distances, around 100 meters or less are taken into consideration. Moreover, more obstacles tend to be on the signal path. Furniture, doors, moving people and wall at the first place lead to significant signal loss. What is more, even the smallest changes in the environment can impact the quality of the propagated signal. All of those obstacles are a potential cause of the multipath reflection, what can produce smearing or cancellation of the signal.

Indoor obstacles were divided into two types: hard partitions – all of the physical and structural components of the building, and soft partitions which are all of the fixed or movable objects that do not extend to a buildings ceiling (furniture, etc.). Both of them are effectively penetrated by the radio signal. Unfortunately, the way how they influence signal propagation is hard to predict.

All those factors make a building that could be free from multipath reflections, diffraction caused by sharp edges or scattering from flat surfaces like walls, almost impossible to design. Even usage of the signal propagation 'friendly' materials like wood or fibre glass would not be much helpful. What is more, most of the existing buildings were designed without thinking about wireless technologies that are going to be used there, and we still need to have at least a rough idea how the signal is going to be propagated.

To determine the signal levels and range of signal losses present in a building number of studies and measurements have been made [4]. Figure below, shows scatter plots of radio path loss as a function of distance in a typical office building for propagation through one through four floors.

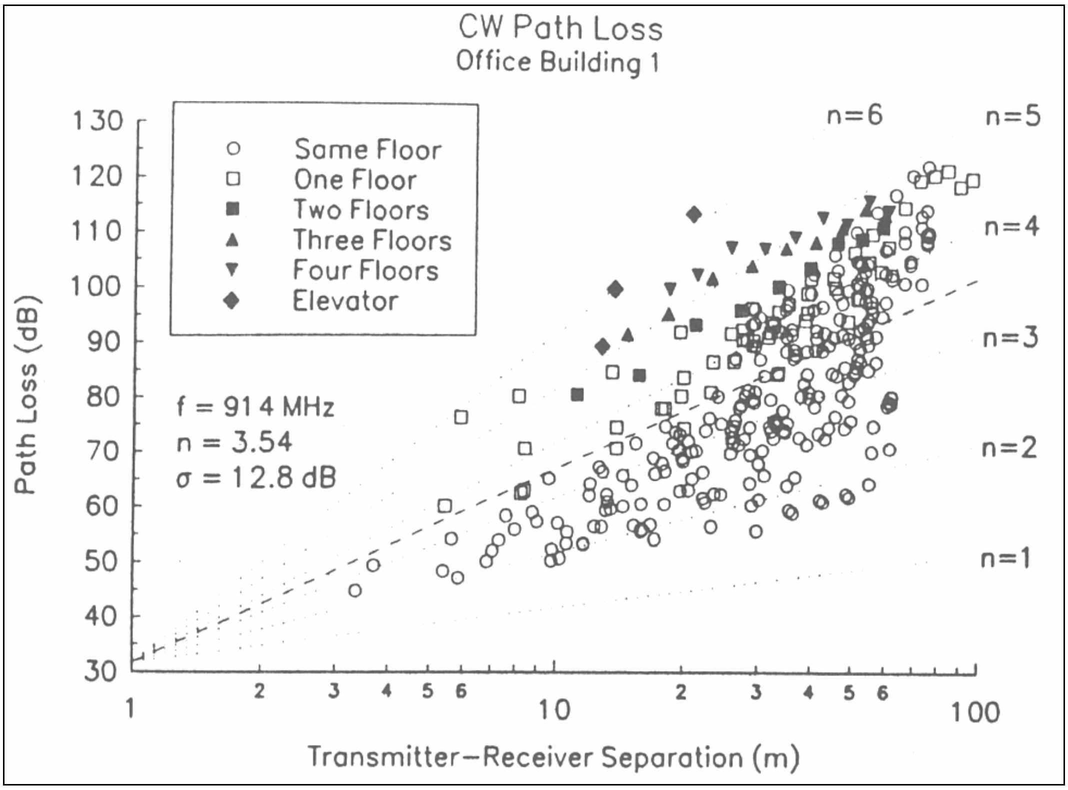

Fig. 2 Path Loss Scatter Plot in a Typical Building. [4]

Analysis of the results of provided data give us a rough idea about the problems connected with indoor propagation. It is clearly visible that generally distance of 10 meters cause losses from 50 to 80 dB.

In the typical indoor scenario we have resources that allow us for losses of approximately 120 dB. According to the data presented above, we can expect the biggest lost in the very first 10 meters. However, the previous figure takes into consideration multi level building propagation, what could be a little bit unreal in common one floor case. Than, the loss within 10 meters could be limited to about 60 dB. Where, after 50 meters we could expect the losses even up to 110 dB, what is almost our entire budget. Simple, it means that in a distance more than a 50 meters form base station we will have huge problems with communication, even with a complete lack of connection.

Fortunately, the problem is not as bad as it looks. In that moment the indoor propagation modelling comes in. It takes into account losses between rooms and through the various radio obstacles and hard / soft partitions within a typical building.

2.1.3 Indoor Propagation Modelling

We can differentiate two most popular types of propagation modelling: site-specific and site-general. First one uses building blueprints with furniture and equipment layout. Modelling process is based on ray-tracing methods, which are trying to predict the path of the signal waves. Need of detailed information makes this method inefficient when it comes to large buildings. What is more, data about moving objects could not be introduced. The second model provides gross statistical predictions of path loss for link design. Its advantages make it popular in initial design and layout of indoor systems. In this section I will try to briefly describe two most popular models: the ITU and the log-distance path loss model.

2.1.3.1 The ITU Indoor Path Loss Model

The ITU model used in site-general indoor propagation path loss prediction [5] is given by:

$$L_{total} = 20\log_{10}(f) + N\log_{10}(d) + Lf(n) - 28\text{dB}$$

Where:

N is the distance power loss coefficient

f is the frequency in MHz

d is the distance in meters (d > 1m)

$Lf(n)$ is the floor penetration loss factor

n is the number of floors between the transmitter and the receiver

2.1.3.2 The Log-Distance Path Loss Model

Another site-general model is called the log-distance path loss model [6] and it is:

$$L_{total} = PL(d_0) + N\log_{10}(d|d_0) + X_S \qquad \text{dB}$$

$PL(d_0)$ is the path loss at the reference distance, usually taken as (theoretical) free-space loss at 1m

$N/10$ is the path loss distance exponent

X_S is a Gaussian random variable with zero mean and standard deviation of σ dB

Path loss is defined by the following equation: $PL = 20\log_{10}((4\Pi d/\lambda))$ dB

Where:

d is distance (same units as λ)

λ is a wavelength (same units as d)

2.1.4 Problems: interference

Another important aspect that should not be omitted when it comes to wireless system operated in indoor environments is the interference. In comparison with the outdoor environments, where distances are much greater, in this case even system that is operation within a range of a few meters or less can have huge impact on the system.

The most common example, which is important for us, is a personal computer operating with a wireless network adapter and a wireless keyboard/mouse. The most common standard for wireless devices is a Bluetooth standard, which uses frequency hopping in the 2.4-GHz ISM band. In the same system setup we have network card that is likely to operate in use of 802.11b or g standard. In this case all devices use the same frequency band, what causes potential interference. What is more, this is not the end. In our scenario our offices usually are full of other equipment that is likely to cause other problems. Even the simplest ones, at the first sight completely innocent, could create unwanted interferences. Among many others the most common are fluorescent lighting and different kinds of office equipment, even monitors. Moreover, the computer itself, exactly all components built with use of many different high-frequency clocks create harmonics, which may fall within the system's passband.

That kind of troubles, which we should keep in our minds while designing indoor wireless systems, clearly show that sometimes communication problems might not be a propagation issues, but rather interference matter. In many cases, just simple change in layout of the pieces of the equipment can solve the problem. It is important to remember that we should take into account and estimate interference scale by computing signal-to-interference ratio. In general, for operation that could be accepted the signal-to interference ratio should be at least as large as required signal-to-noise ratio.

2.2 Existing simulators and Planning Tools for Indoor Wireless Systems

2.2.1 WinProp – by AWE Communications

One of the most sophisticated wave propagation and network planning CNP (Combined Network Planning) suits on the market. Developed in Germany, is a unique system that offers tools for many different standards for outdoor and indoor wireless signal propagation. Coverage, data throughput, interference and blocking can be predicted based on the path loss predictions obtained with ProMan or other prediction tools.

It uses pre-defined network planning modules for the following wireless standards:

- Broadcasting
  - SDMB
- 2G and 2.5G Networks
  - GSM
  - GPRS / EDGE
- 3G and B3G Networks
  - UMTS FDD
  - UMTS TDD
  - UMTS FDD incl. HSDPA
  - TD-SCDMA
  - B3G OFDM Networks
- Wireless LANs (WLAN)
  - IEEE 802.11a
  - IEEE 802.11b
  - IEEE 802.11g
  - HIPERLAN/2
- WiMAX
  - IEEE 802.16

The graphical user interface (GUI) of WinProp runs on all versions of Windows OS.

2.2.1.1 Indoor propagation

Phenomena mentioned in previous sections, like multi-path propagation, reflection, diffraction and shadowing have a major impact on the received power. Based on 3D vector databases with planar objects WinProp is able to compute very fast path loss and wideband properties of the radio links inside buildings. To make it possible it uses different combinations of available propagation models for indoor scenarios. They can be broken down into two types:

- Empirical Models:
  - One Slope Model
  - Motley Keenan Model
  - COST 231 Multi Wall Model
- Ray Optical Propagation Models:
  - Indoor Dominant Path Prediction Model (IDP, 2D and 3D)
  - 3D Standard Ray Tracing (SRT)
  - 3D Intelligent Ray Tracing (IRT)

2.2.1.2 Modelling the multi-floor indoor propagation

The application includes also an option for multi floor buildings predictions. This type of calculation requires calculation on each floor. WinProp can compute predictions on an arbitrary number of heights and can display the result in a 3D views. Thus, interference between floors and coverage problems can be easily predicted. The graph below is an example of the program multi floor building signal propagation prediction.

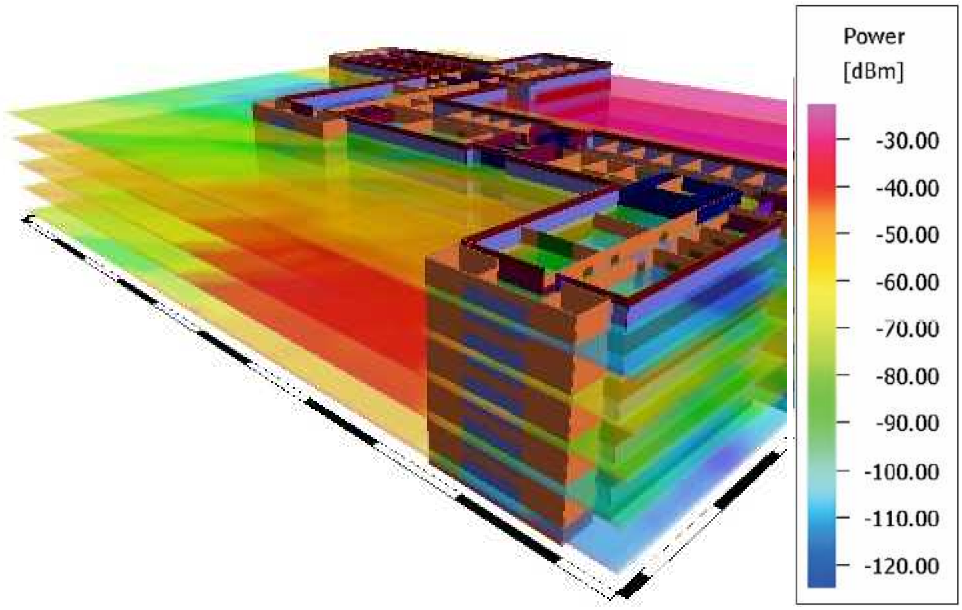

Fig. 3 WinProp: Sample prediction with an indoor prediction model

2.2.1.3 Planning of W-LAN networks

WinProp offers fast and accurate propagation models integrated in a user-friendly GUI application. The user is able to define multiple access points' layout in the building; as a result the coverage area can be computed for each access point separately or for an entire area as well. Based on the predictions of the signal propagation for each access point a set of radio network planning outputs is produced. Following figures present results of computations which allow the user to analyze the WLAN performance:

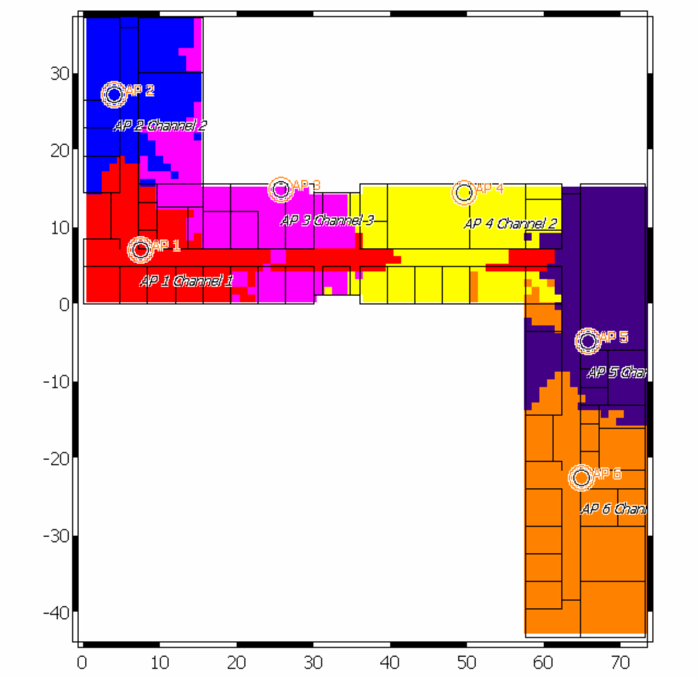

Fig. 4Cell Area/Assignment of MS locations to AP

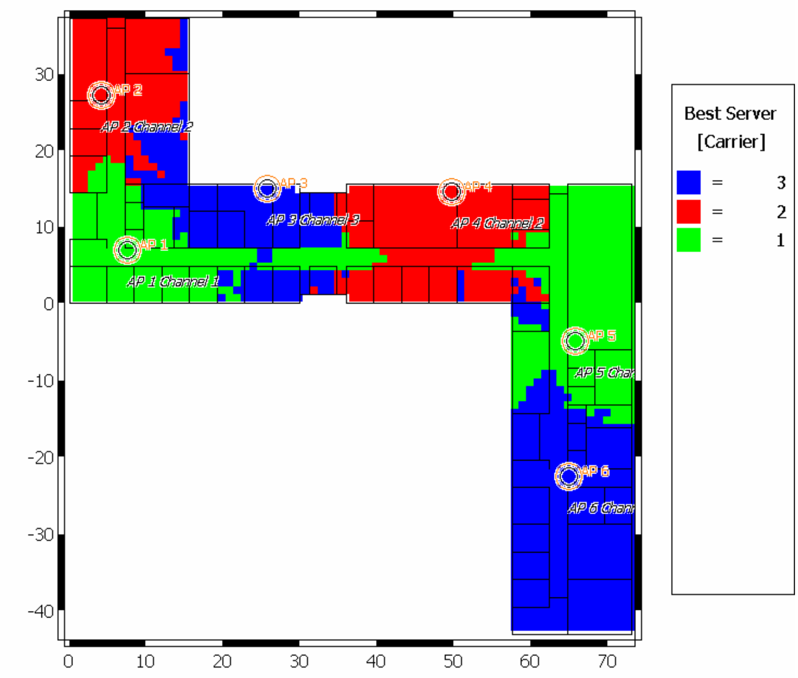

Fig. 5 Best Server Area Assignment of MS locations to carriers

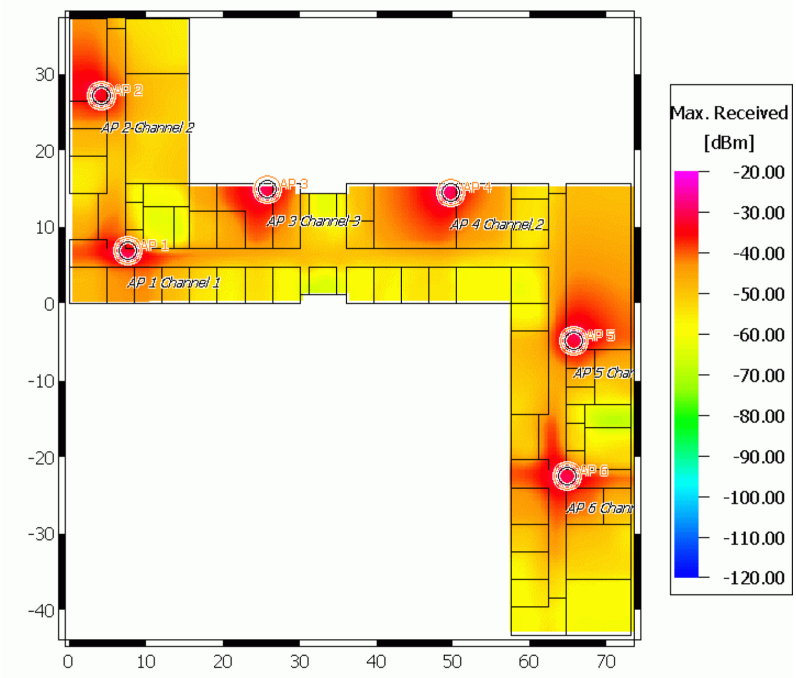

Fig. 6 Maximum Received Rx Power Max. power which
can be received by a mobile station in downlink

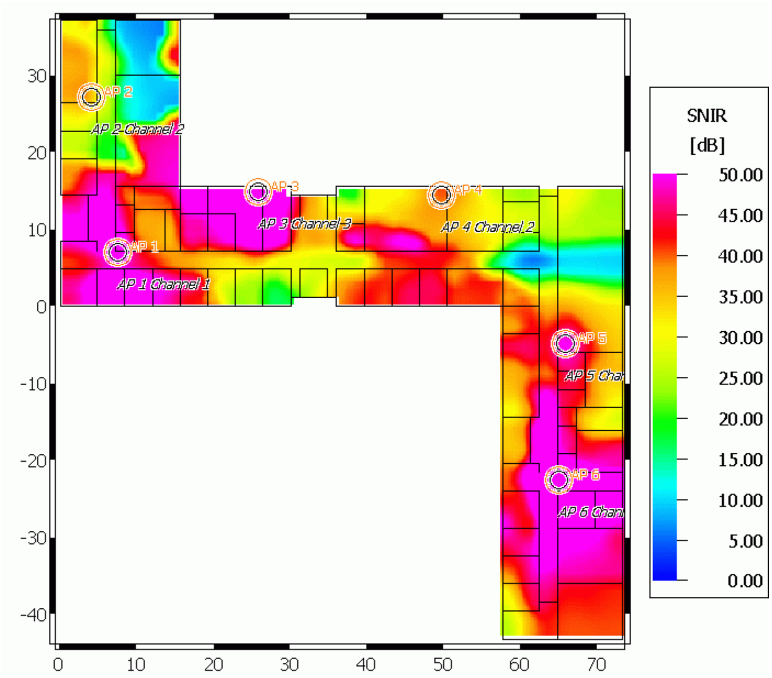

Fig. 7 SNIR (Signal to Noise & Interference Ratio) Signal relative to noise and interference
(on same carrier) Output only for carriers assigned in best server map

2.2.1.4 Network Optimization

WinProp includes as auto optimization module as well for correction and refining WLAN configurations. The module is responsible for adjusting access points' locations automatically to the defined needs. What is more, the optimization modules can consider location dependent traffic, if the users are not homogenously distributed in the scenario. As a result, hot spots can be modelled and considered during the optimization process.

2.2.1.5 Building Databases

Propagation of signal in the indoor environments requires detailed information about the walls and object inside the buildings. To make results as much accurate as it is possible WinProp uses 3D vector databases for its propagation models. Moreover, it includes an additional tool specially designed for building databases that are used in a process of signal propagation. For most indoor planning tools, the handling of the building data is the most critical part. This CAD tool allows the generation of building databases within a few minutes based on scanned bitmaps or CAD data.

2.2.2 WiSE - A Wireless System Engineering Tool – by AT&T Bell Laboratories

WiSE is a simple system for signal wave propagation in indoor and microcell environments. Consist of three main functionalities: prediction, optimization and interaction. First option in responsible for computing predictions as an output, where building information (wall locations and composition), system parameters, and access points' locations are input data. Its main aim is to determine the wireless system performance. In a prediction computation process the basic propagation model with a several modifications is used. Second feature, given parameters and system requirements compute alternative layout of base-stations to optimize the network planning process. The last part of the system is the user interface. It displays plan, elevation, and perspective views of a building and shows in living colour the power received at each location for a given base-station site. Examples of application in use are showed below:

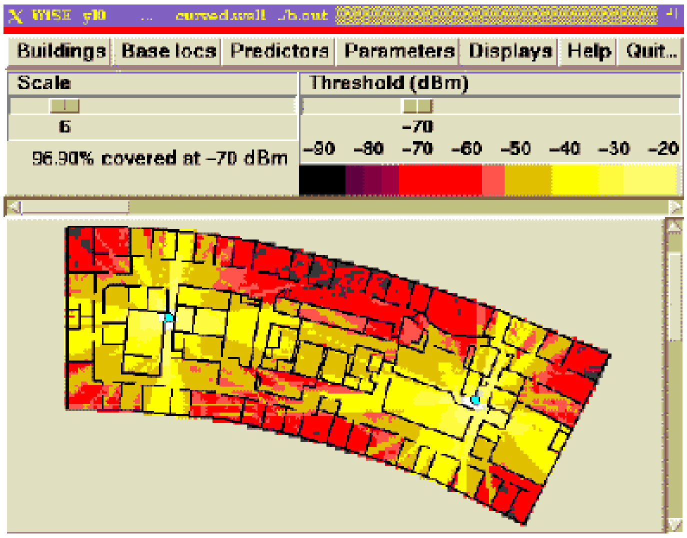

Fig. 8 WiSE: Predicted coverage of a small office building with two base stations.

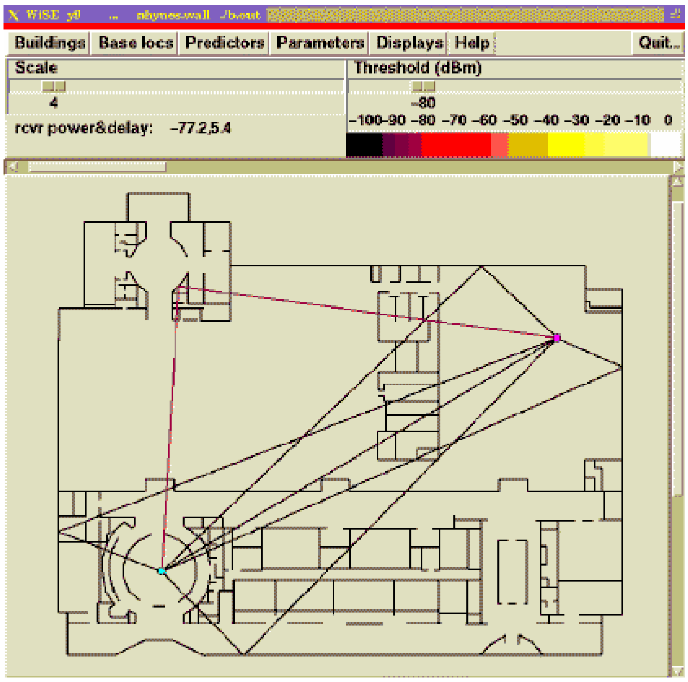

Fig. 9 WiSE: Ray tracing

WiSE runs on UNIX systems with X-Windows and in a more restricted form on PC's under Microsoft Windows.

2.2.3 Volcano – by Siradel

Volcano is a propagation prediction software suit developed by French company called Siradel. More than 10 years of experience lead to a creation of very sophisticated set of tools connected mostly with outdoor signal propagation. Among many sophisticated tools such as optimized ray-tracing/UTD, automatic tuning, multi-resolution capabilities, indoor penetration, wideband propagation application includes as well mini-, micro- and pico-cellular propagation simulation. Application is capable of calculating multi-floor cases as well.

To achieve accurate predictions Volcano uses its unique deterministic models developed through the years. They are applicable to rural, suburban, urban and indoor environments, thus enhancing the accuracy and computation speed of the radio planning tool that is interfaced with it.

The application computes how radio signal is propagated taking into consideration all environment constrains. As a result it produces a field strength coverage map that represents a signal propagation which allows users to place base stations and antennas in locations to achieve the optimum result. Examples of files produced are shown below:

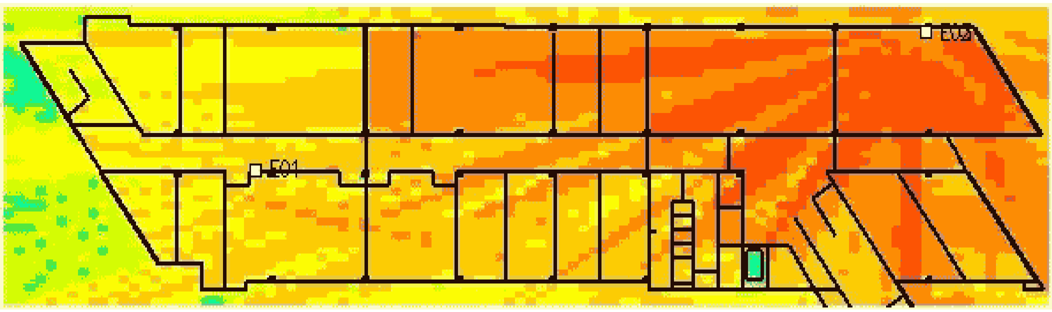

Fig. 10 Volcano: radio attenuation predicted around the omni-directional transmitter E02 (top right corner) with a directional receiving antenna.

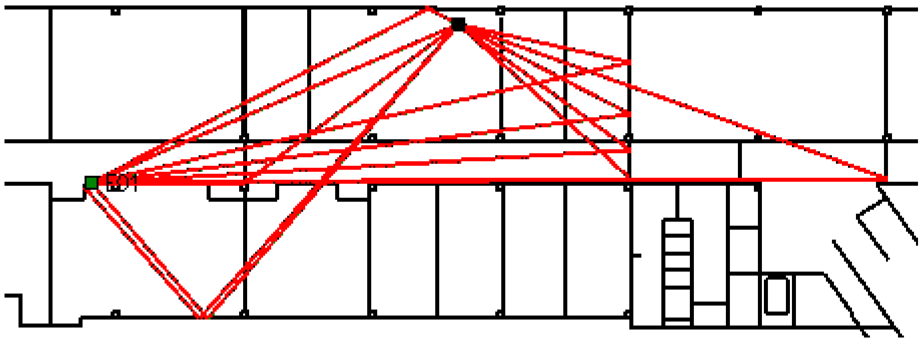

Fig. 11 Volcano: trajectory of the main radio contributions between the transmitter E01 (in the left) and one particular receiver location (in the top).

2.2.3.1 Modelling the multi-floor indoor propagation

An additional option of the Volcano is a multi-floor prediction. The application uses a specific model to compute signal propagation in multi-floor buildings, when the transmitter and the receiver ale placed at the different levels. Based on 3D building blueprints a specific algorithms use a ray-tracing technique to calculate the multi-path components reflected or diffracted between the transmitter floor and the receiver floor in a shortest time keeping the results as accurate as it is possible.

Graphs below illustrate an output example of the signal propagation through 4 levels building where the 2.4 GHz isotropic antenna is placed in the entrance hall.

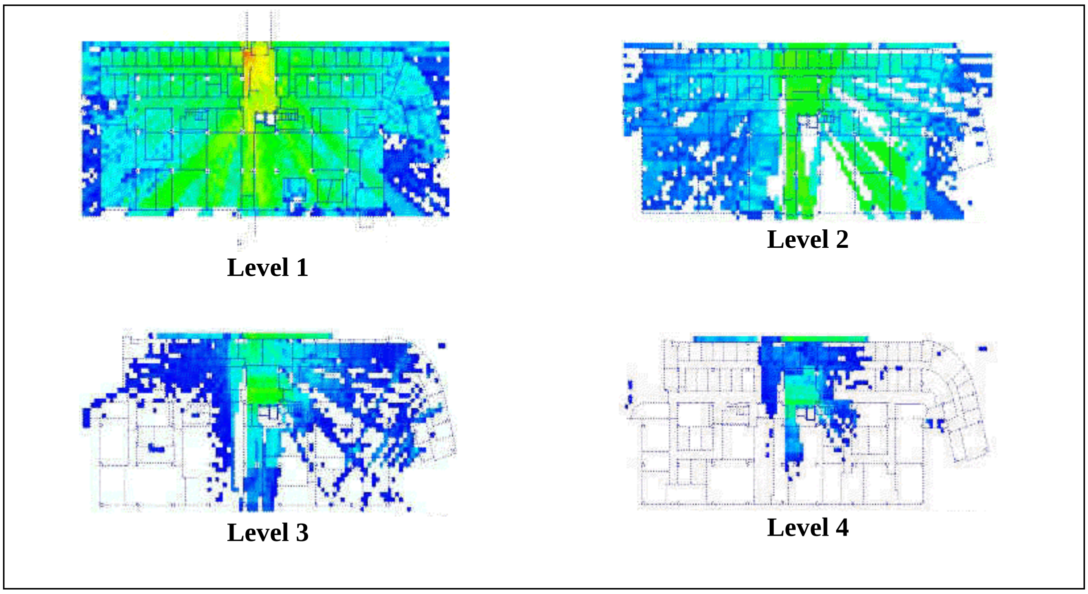

Fig. 12 Volcano: multi-floor signal propagation prediction

2.2.4 CINDOOR - by University of Cantebria, Spain

CINDOOR is a tool developed in one of the Spanish universities specially for predicting radio signal propagation in an indoor and urban micro and picocell environments. Computes coverage and channel performance, explore indoor/outdoor interaction. In a prediction process application uses method based on a full three-dimensional implementation of GO/UTD (Geometrical Optics/Uniform Theory of Diffraction). Moreover, tool is capable of ray tracing carried out by combining Image Theory with Binary Space Partitioning algorithms. It provides features useful in a process of planning a wireless system such as coverage, fading statistics, power delay profile, and associated parameters, such as rms delay spread and the coherence bandwidth. Sample results are shown below:

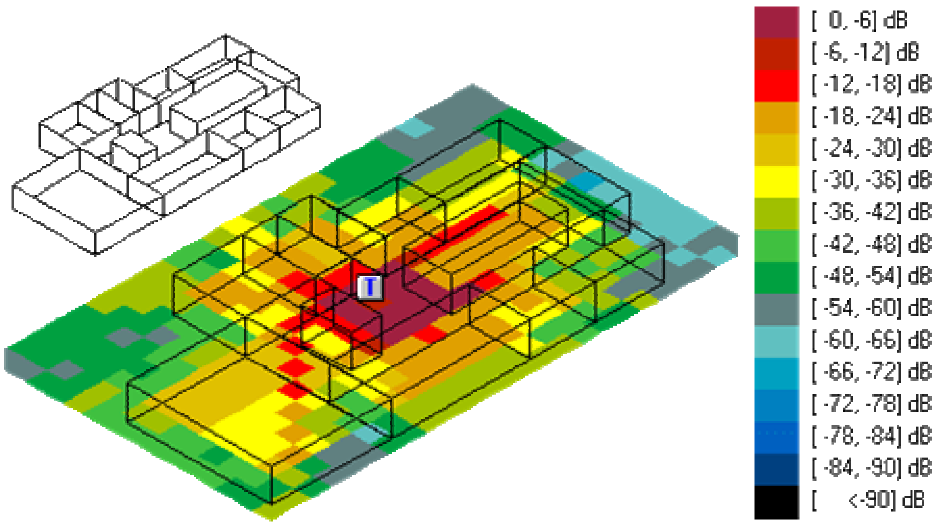

Fig. 13 CINDOOR: coverage map

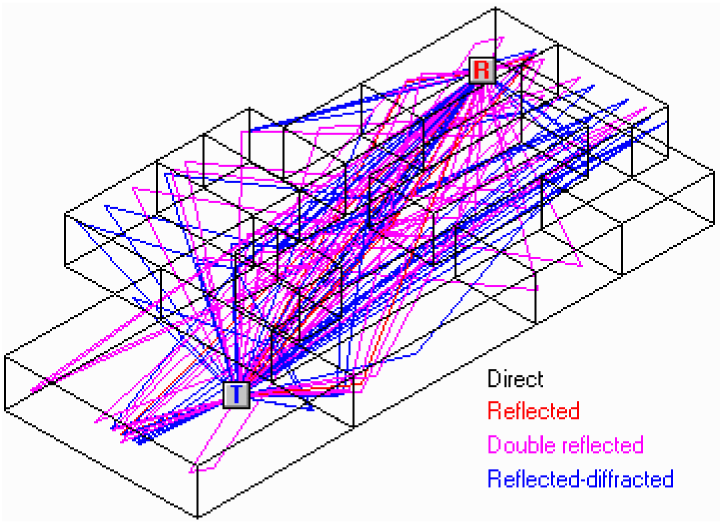

Fig. 14 CINDOOR: Ray tracing prediction

2.2.5 Mesh/LanPlanner – by Wireless Valley Communications

Wireless Valley Communications developed a complex suite for planning of sophisticated wireless environments. It provides design, measurement, optimization, and management engineering features for in-building, campus-wide, and microcell wireless communications systems. The tool suite includes the Predictor, InFielder, and Optimatic modules that work seamlessly for all phases of deployment, maintenance, and optimization of a local wireless system.

Algorithms and prediction models used allow visualizing site-specific coverage, capacity, and the physical location and configuration of all installed infrastructure on building blueprints and campus maps. The following features minimize design and deployment costs of the network building process:

- Site-specific models that visualize the physical location and configuration of all installed network equipment
- Automated placement and configuration of access points
- Highly accurate coverage and capacity predictions

Moreover, suite in a site-specific modelling uses unique database system. It allows converting drawing files or paper floor plans into multiple-story building databases that can be used later on. In addition, the RF characteristics of walls and other obstructions can be easily added. Information can be easily imported from the following sources as well:

- An existing AutoCAD, Visio, or bitmap drawing
- A scanned image, digital photograph or PDF file
- A free-hand or electronic sketch of any site

The tool provides predefined functions for designing IEEE 802.11a/b/g and multi-band systems in the most complex environments with great accuracy. It supplies information such as RSSI (Received Signal Strength Indicator), SIR (Signal to Interference Ratio), SNR (Signal to Noise Ratio), and throughput and bit-error rate. Furthermore, software includes additional aid for network optimisation and measurement. As a result it plots graphic outcomes on a site-specific 2D/3D models. Examples are shown below:

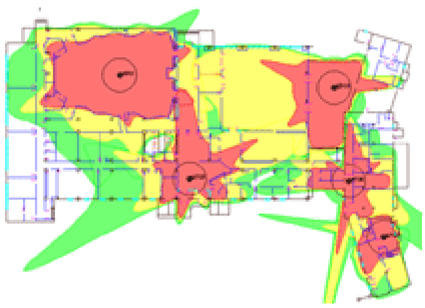

Fig. 14 LANPlanner: Signal coverage prediction

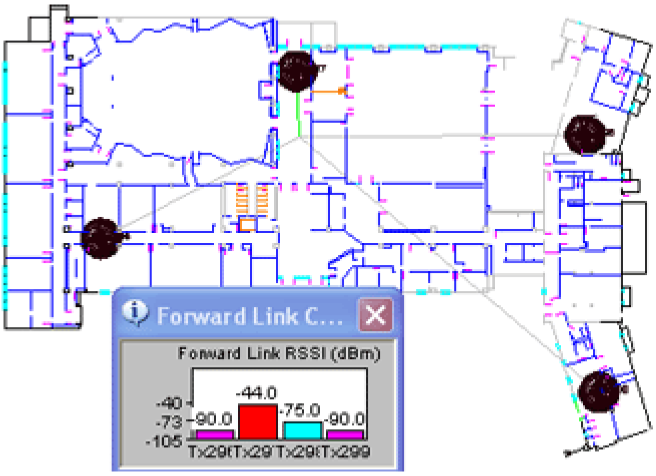

Fig. 15 LANPlanner: predicted signal power from multiple access points.

2.2.6 EDX Signal Pro – by Comarco Wireless Technologies

EDX SignalPro is a complete software suit offering a complete set of planning tools for wireless communication systems. EDX SignalPro contains propagation prediction tools for wide area service prediction, link analysis, point-to-point and point-to-multipoint studies. One of the modules, called MIM (Microcell/Indoor Module), is specially designed for microcell and indoor wireless communication systems. It includes multi-site coverage and interference analysis, multiple point-to-point link analysis, a comprehensive set of propagation models, full mapping capabilities, and full access to terrain, groundcover, building, demographic, traffic, and other databases. It uses both two-dimensional and three-dimensional ray-tracing propagation models which have been optimized for accuracy and efficient performance in the indoor environment. Universal database provided with a suite includes detailed information about the RF characteristics of walls and other obstructions. Results of the computations can be displayed in a different ways, and several studies can be displayed simultaneously. Some of the results examples are shown below:

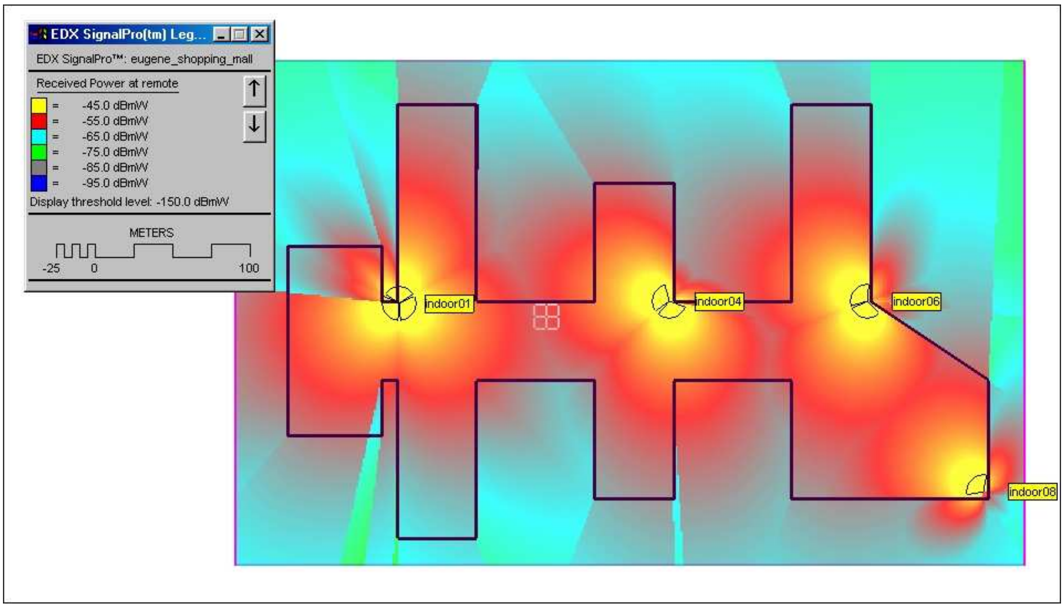

Fig. 15 EDX Signal Pro: Predicted coverage area

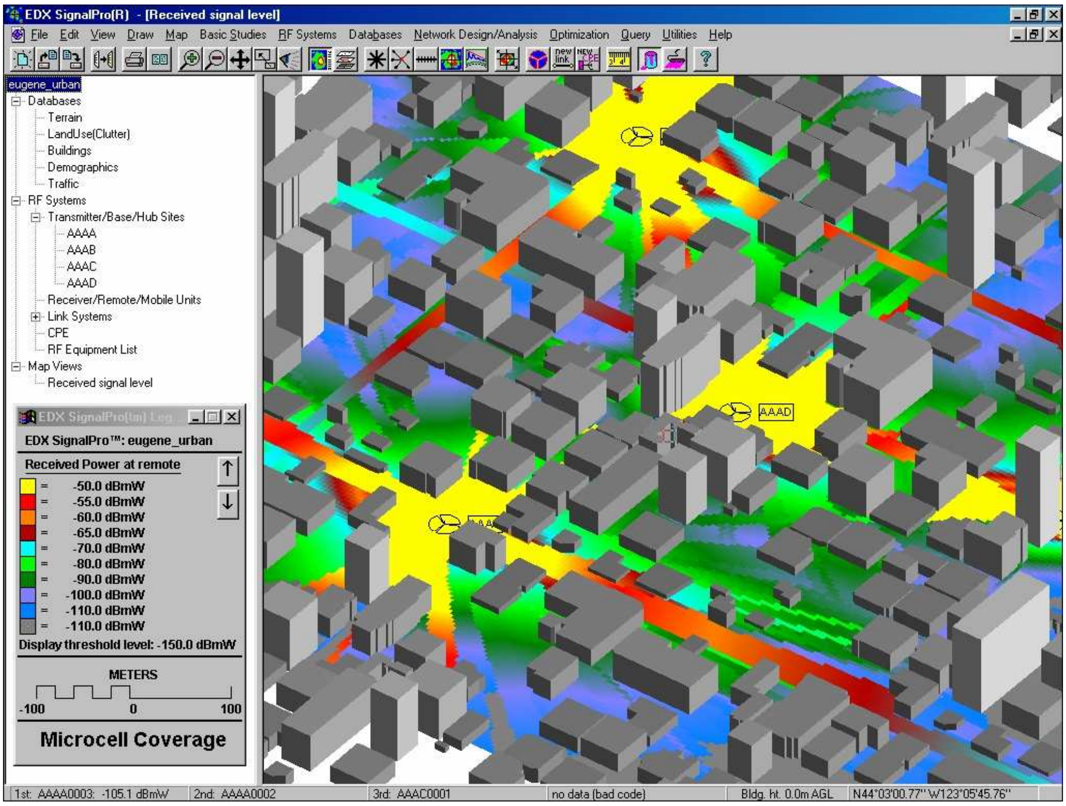

Fig. 16 EDX Signal Pro: Signal prediction in a microcell environment

For the best result EDX Signal Pro uses combination of different modifications of the following propagation models:

- 2D and 3D ray-tracing models for outdoor microcell and indoor wireless LAN/PBX/cell-extender environments

- EDX Simplified Indoor Model for rapid, site-specific indoor signal strength calculations. Takes into account:

    - Line-of-sight rays
    - Wall transmission
    - Corner diffraction
    - Attenuation due to partial Fresnel zone obstruction.

- COST-231 Walfisch-Ikegami propagation model for simplified outdoor microcell studies.

What is more, application is capable of multi-level indoor signal propagation. Calculates attenuation between floors and plots results in 3D, multi-level graphs. In an addition to overall functionality of the system Time Signature Displays are available. Their main aim is to show the waveform of the received data pulse as a function of time. As a result, pulse shape distortions and echoes due to multipath are easily visible. Pseudo-animation mode shows the dynamic nature of the distortions.

2.3 VALVE Source Engine

The Source engine technology, the same used to power Half-Life 2 game, is widely available for any third party users. Source supplies major enhancements in several key areas including character animation, advanced AI, real-world physics, and shader-based rendering.

Engine provides powerful possibilities of characters animations, including animation of sophisticated facial expressions. In addition, these characters possess the industry’s most advanced artificial intelligence, making them extremely capable allies and foes. All those attributes allows developers to create amazing characters and creatures.

Fig. 17 Source Engine in action (game: Half-Life 2: Episode One); source: Valve Software

These characters populate beautifully rendered and physically simulated worlds. Current applications require the use of a physics simulation to provide realistic and responsive environments. That is why it could be easily used for real world physics simulations. All this features allow developers to break from authoring the pre-scripted events featured in previous generations of games, and open the door for the creation of completely new styles of play.

Moreover Source engine contains robust networking code, providing support for 32-player LAN and Internet games, and includes a complete toolset for level design, character animation, demo creation, and more. The main features of the engine that allow creating realistic 3D environments are presented below. [7]

In order to create virtual world as closest to the environment that surround us Source engine provides very sophisticated renderer tools and uses modern techniques of 3D graphic effects. Among others, author shaders with HLSL [1] or dynamic lights, vertex lighting and light maps with High-Dynamic Range system[2].

In relation to indoor and out door environments engine allows to create deformable terrains, dynamically rendered organics (grass, trees etc), real-time radiosity lighting. Effects include but are not limited to: particles, beams, volumetric smoke, sparks, blood, environmental effects like fog and rain.

What is more Valve software developed unique materials system. Instead of traditional textures, Source defines sets of materials that specify what the object is made from and the texture used for that object. A material specifies how an object will fracture when broken, what it will sound like when broken or dragged across another surface, and what that object's mass and buoyancy are. This system is much more flexible than other texture only based systems. Materials can interact with objects or NPCs [3] such as mud or ice for vehicles to slide/lose traction on.

In addition, engine contains components responsible for both LAN based multiplayer and Internet based multiplayer games with prediction analysis for interpolating collision/hit detection and optimizations for high-latency, high-packet loss 56k connections.

---

[1] High Level Shader Language (HLSL): shader language developed by Microsoft for use with Direct3D

[2] High dynamic range rendering (HDRR or HDR Rendering) or sometimes high dynamic range lighting is the rendering of 3D computer graphics scenes by using lighting calculations done in a high dynamic range

[3] A non-player character (NPC) is a character in a game whose actions are previously determined

One of the characteristics that distinguish Source engine from others is advanced physics implemented. They allow for creation of more responsive world with realistic interactions. Sounds and graphics follow from physics. AI characters can interact with physically simulated objects like ropes/cables, machines, constraint systems that could be design with complete freedom. What is more, custom procedural physics controllers can be implemented.

Fig. 18 Source Engine in action (game: Half-Life 2: Lost Coast); source: Valve Software

In relation to programming all code is written in C/C++. Among many different characteristics the most important and useful part is that code structure allows easily and quickly derive new entities from existing base classes. Moreover, modular code design (via DLL's) allows swapping out of core components for easy upgrading or code replacement.

2.3.1 Mod development

Building a "MOD", a game modification which relies on another game's core technology is very often the best way to develop a new game. When you rely on game technology as well-established as the Source engine, time and effort can be spent building creative gameplay and unique content rather than on things like rendering technology or network code or collision detection. The Source Engine and its associated SDK provide the most efficient, complete, and powerful game development package. Thanks to advanced physics implemented in a game the same engine could be used for real world simulation purposes. Moreover, provided tools make the work easier and more effective.

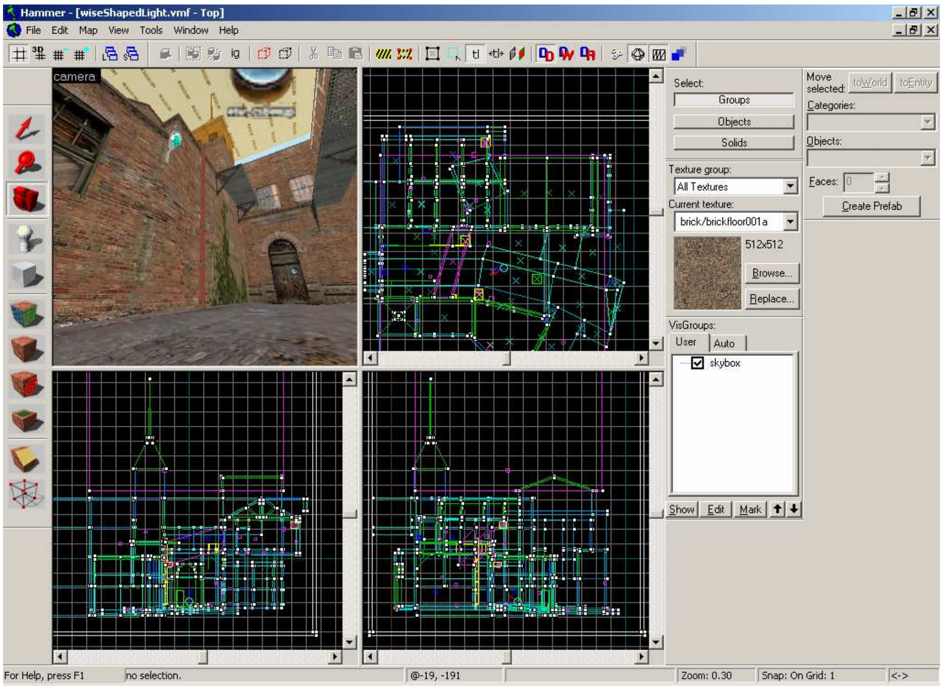

Fig. 19 Hammer World Editor; one of the SDK tools provided by Valve

2.4 Ubiquitous computing simulator

Ubiquitous environments [8] and context aware systems are currently structures that with a combination of surrounding us environment cannot exist without each other. Context aware devices are able to make assumptions about the user's current situation because of strong connections with a physical environment in which the task is being performed. In the paper “A Testbed for Evaluating Human

Interaction with Ubiquitous Computing Environments" [9] authors clearly state that: "The efficacy of such adaptive systems is thus highly dependent on the human perception of the provided system behaviour within the context represented by that particular physical environment and social situation. However, effective evaluation of human interaction with adaptive ubiquitous computing technologies has been hindered by the cost and logistics of accurately controlling such environmental context."

As a result of quoted research a TATUS [10] simulator and its successor PUDECAS were developed. Those ubiquitous computing simulators, based on a 3D games engine, were designed to maximize usability and flexibility in the experimentation of adaptive ubiquitous computing systems. Developed ubicomp simulator:

- Provides a 3D virtual representation of ubiquitous computing environments featuring configurable embedded sensors.
- Easily generates extensive data sourced from large and/or varied sets of embedded sensors
- Avoids the expense and logistical problems involved in the configuration and deployment of real-world embedded sensors required by pervasive computing test-bed.
- Allows experiments to exercise services in a multi-user capacity, drawing on the relationship between user activity and sensor activation to supply the service with environmental context relating to the users physical and social setting
- Allows services to manipulate the environment in response to user activity, based on its electronically sensed view of the pervasive computing environment.

Figure 20 describes overall system architecture, where four main components are visible. Among them a 3D environment based on modification of the game Half-Life 2, which provides the virtual pervasive computing environment.

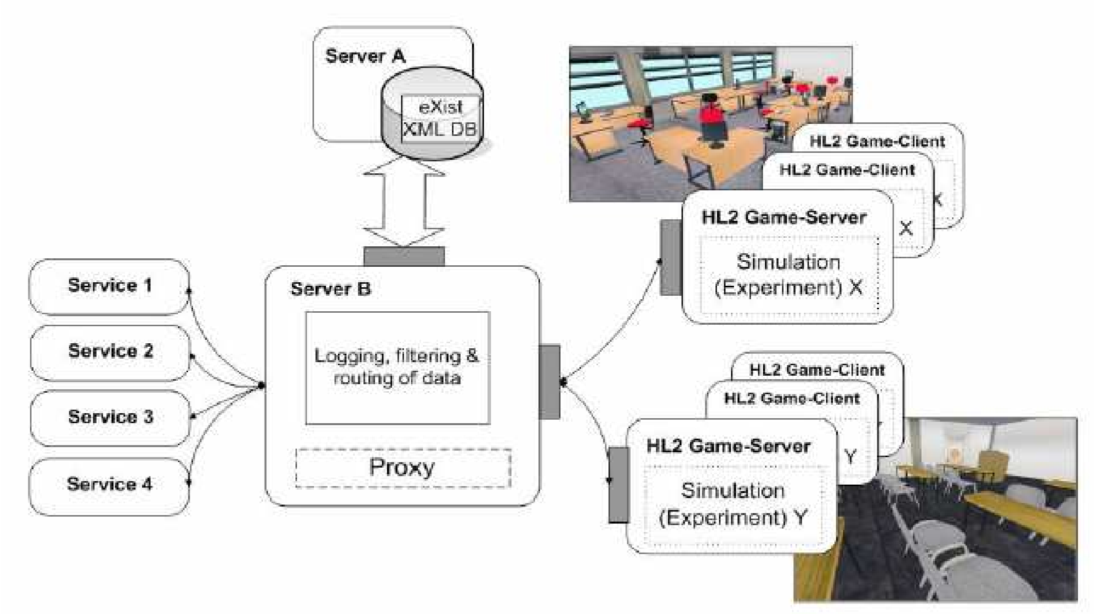

Fig. 20 Ubicomp simulator - platform overview; source: [10]

The system architecture provides a viable solution. Services receive data in a timely manner while users do not suffer perceivable or adverse delays in service response times. Optimised mapping techniques continue to allow experimental environments to grow in size and complexity.

2.5 Conclusions

This chapter has shown that the research area of propagation tools is a quite active one. This is especially the case when considering specific algorithms that are responsible for calculating physics of signal propagation and entire approach of creating predefined databases. Studies on phenomena connected to signal propagation through different materials are continuously updated, as a result databases contain more and more detailed information about particular materials. Work undertaken in the field of signal propagation is mainly focused on improving existing algorithms, as a result producing better and more adequate outcomes, but still with a 'single run' approach. In other words entire problem of dynamic changes in the environment looks like to be completely forgotten or omitted. What is more, there is little evidence of driving progress in the area of visualisation the results. Researchers are more focused on calculating detailed outcomes rather than showing them in accessible way. Lastly, work done in this area does not consider the possibility of interacting with an environment at all. They typically use statistical models of user behaviour, thus not allowing for evaluation that is able to assess the reaction of user test subjects to the adaptive behaviour wireless systems can offer.

Furthermore, ubiquitous simulators like TATUS do not simulate any aspect of the communications networks that must support any operational ubiquitous computing environment. As the vast majority of pervasive environments use wireless network technologies, an adaptation and expansion of functionalities provided by wireless planning tools in such a simulators is desirable.

The situation looks completely different in an area of virtual reality, exactly when it comes to games industry. Modern technology allows creating a very sophisticated virtual environment on a common home PC. And what is more important, results are incredibly close to reality that surrounds us. Game market delivers great number of 3D games engines that can be easily adapted for purposes of this research.

It can thus be concluded that development of entire system for simulating signal propagation in a virtual environment is an original concept. A novel combination of a wireless system planning tool and a popular game will lead to creation of the unique system, which will be able to fully exploit the 3D game engine with a minimum knowledge of rendering technology, and in the same time maximise usability and flexibility of the system. It is hoped that this dissertation may make a significant contribution in realising the development and implementation of such a system.

Chapter 3: Design

This chapter outlines the implementation design. Three phases of implementation took place over the period this dissertation was completed. The first phase was the most important one and involved design and implementation of 3D simulator capable of visualisation of the radio signal propagation. For this part of the work a Valve Source Engine [11]. Second part was connected to signal propagation data transfer from existing external application. Since the main aim was to create fully operational application, the last part of the research was concentrated on creating specific, detailed real world simulation that will be able to show all functionalities of the system.

One of the main factors that affected design process was an incredibly wide potential of the available structures and mechanisms implemented in the Source engine. The best example is the basic modification of the game, which source code consist of over 2500 files (only *.cpp; *.h files). The wide range of settings available through the game engine forced a use of ‘test-and-try’ prototyping approach. Exploitation of all main 3D effects was the most appropriate come up to asses the visual impact of each of the properties tried especially from a moving 3D point of view. The following sections outline the design of three components of the final system in details.

3.1 Overview

Figure below (Fig. 21) illustrates the specific components of proposed implementation. These components are divided into four parts: propagation simulator, ‘JAVA .SVG to .VMF converter’, ‘Source SDK’ tools and main application, 3D simulator. It has already been established that the implementation of core application will be based on a modified version of ‘Half-Life 2: Deathmatch’ game.

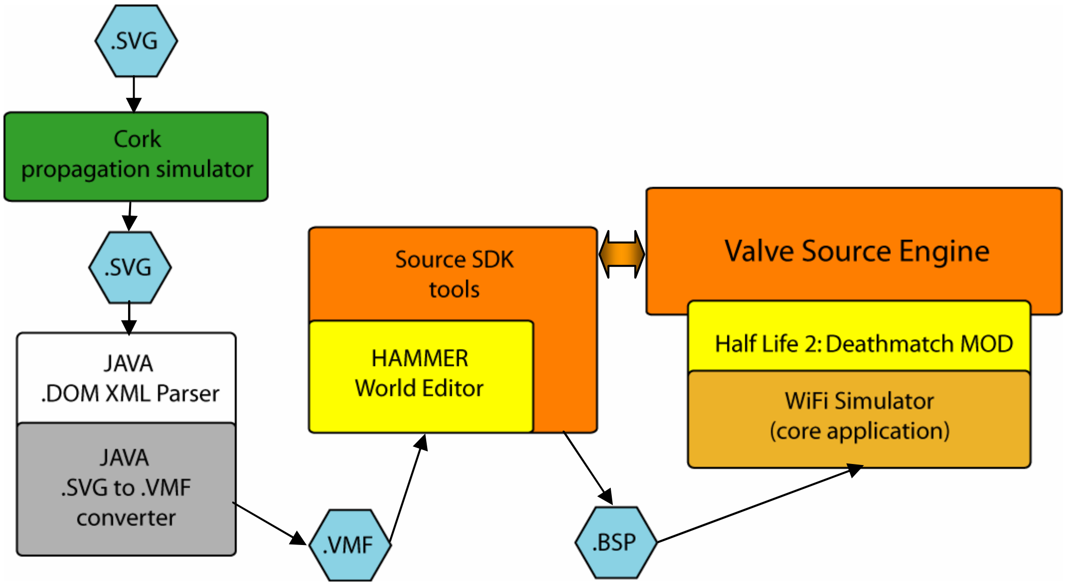

Fig. 21 System overview

The most important component of the entire system is the main application, that is a modification of 'Half-Life 2: Deathmatch' game. 'WiFi Simulator' is an application based on 'Half-Life 2' game environment. With such an approach, when you rely on game technology as well-established as the Source engine, the most of the time and effort can be spent building creative environment and unique content rather than on things like rendering technology, network code or collision detection. It provides rendering, sound, animation, user interface, networking, artificial intelligence, and physics. What is more, tools provided with a 'Source engine' make work easier and more effective.

The core application is responsible for carrying out complete emulation of the real environment in which simulation of RF signal propagation is conducted. In other words, user is able to transfer into virtual world and by moving around check the area covered by the signal of base stations placed in a building.

Because of the area of the research that is limited to visualisation aspects of the simulation it is necessary to use external tool for calculating signal propagation. This task is delegated to simulator developed by Cork Institute of Technology.

To automate the process of creating entire simulation it is necessary to develop an application-in-a-middle that will be responsible for converting simple map files (blueprints of the building) into format supported by simulator.

Moreover, functionality offered by ‘Source SDK tools’ provided by ‘VALVE’ cannot be omitted here. Despite the fact, that their impact is not directly visible and they do not create any part of the system itself, the entire development of the system would not be possible without them.

A detailed technical discussion of these components as implemented is undertaken in the next chapter.

3.2 Simulation process

Illustration of the basic steps for conducting the simulation is done by means of activity diagram.

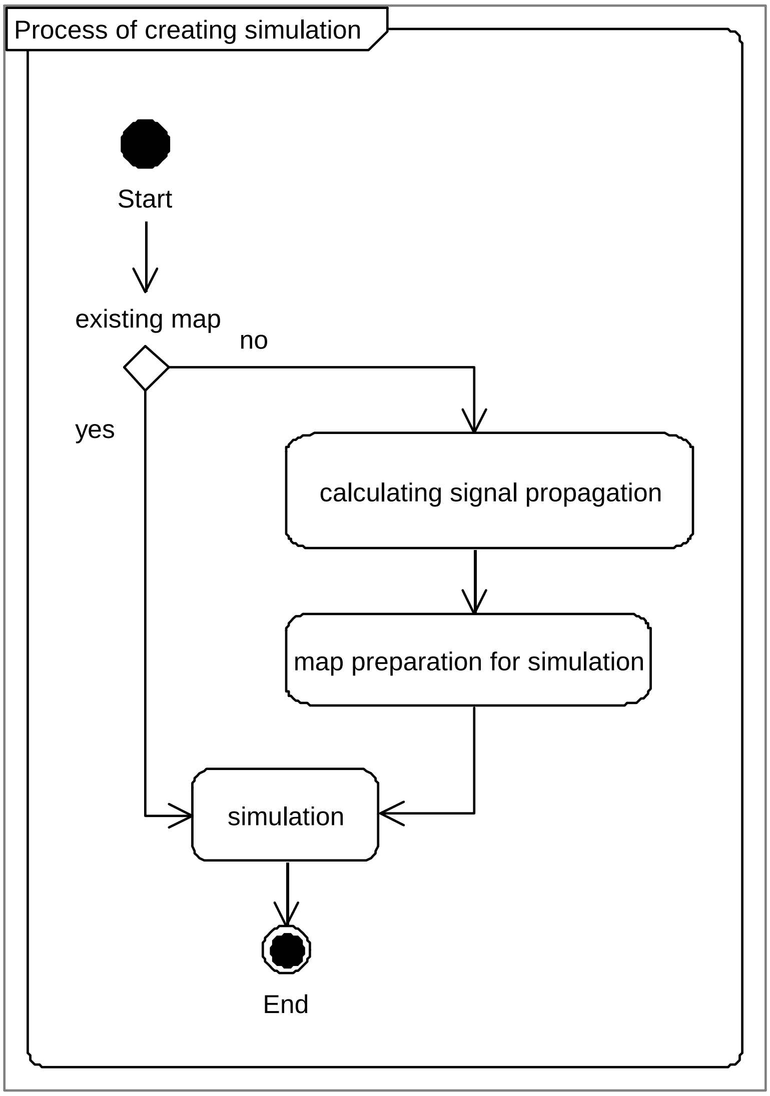

Fig. 22 Activity diagram: Process of creating simulation

In order to conduct a process of simulation user is required to perform following steps. The start of the operation depends if user has only a blueprint of the location with information about base stations and wants to perform entire simulation from the beginning or already has a map of the environment ready for simulation. In a first solution, prediction of the radio wave propagation needs to be conducted in a first place, what is delegated to remote application (in this case it is 'Cork simulator'). Due to the difference in file format of the output of the propagation tool and format accepted by simulator it is necessary to convert it to applicable form – on the diagram this process is called a 'map preparation'. As a result of previous steps the actual simulation could take place. When the user has a map that is 'ready-to-go' (for an instance: map previously used) all of those steps could be omitted and simulation could be directly executed.

The entire process with more information about implementation and activity diagram with more details is described in the *Implementation* chapter.

3.3 Functionality requirements

The top goal for this project is to provide a flexible 3D virtual environment that can be used to test wireless systems. Previous chapters give a broad overview of main key objectives for the simulator which in the initial stages of the project were outlined in the most significant terms.

3.4 Conclusion

In conclusion, a development of entire system will involve coding of two separate components:

- WiFi Simulator, a main application - modification of 'Half-life 2: Deatchmatch' game
- map converter, separate application responsible for map conversion

Due to the nature of the system and components that are used in a development, at this phase of the research is hard to be more specific and detailed about components of the system and its design. Mainly, because of the enormous functionality and possibilities of the source engine that needs more practical investigation. As a result, a process of design overlaps the implementation process and vice-versa. Limitations of the source engine could affect overall assumptions made at the beginning or on the contrary could open the area of the new possibilities which existence were unknown at this stage.

Chapter 4: Implementation

This chapter details how the design ideas and principles outlined in the previous chapter were implemented. The first section reviews the supporting tools used for implementation purposes. The remaining sections introduce how the actual system and its subcomponents were built.

4.1 Technology review

Of the tools examined in the state of the art chapter several were used for implementation. Due to the completely different nature and character of two main applications developed a division into two separate parts is essential.

4.1.1 WiFi Simulator – a core application

It has already been established that the implementation of core application will be based on a modified version of 'Half-Life 2: Deathmatch' game. In order to develop this application Source SDK tools were used. The most important feature provided by these tools is a Half-life 2 source code. In an automated process an entire file structure of the modification is created with a basic functionality provided, in other words, a perfect skeleton for a future modification. Complete application was written in a C++ programming language, that is why for introducing essential changes Microsoft Visual C++ .NET 2003 [4] environment was used.

Because of entire technology that lies beneath the engine like rendering technology, network code or collision detection the main aim of developing simulator was to create visual effect that represents wave signal propagation in 3D

---

[4] Microsoft Visual Studio is an advanced integrated development environment by Microsoft

world. Remained elements were users interface and alternatively overall enhancements of the system.

4.1.1.1 Visual effect for signal propagation

The main goal of the entire system is the visualisation of radio waves signal propagation through the indoor environment in a virtual 3D world. That is why part of the research and development of the actual visual effect that is able to carry within itself all of the required information was the most essential. In order to develop this kind of visual facet following aspects need to be taken into consideration: first of all, the effect needs to represent visually all information about propagated signal and in the same time cannot be obtrusive for general world representation, and what is more important cannot disturb in a general users movement in a virtual world. Moreover, its image should be unconsciously associated in users mind with its functionality. As a result it is established that this effect should be represented by some kind of mist or fog. Mainly because this kind of visual effects meet requirements previously listed. Even with a humans nature we are saying about getting lost in a fog what can be easily associated with loosing the power of the signal from base stations. Furthermore, on the programming level the effect should be easily applicable in different situations and flexible in employ.

4.1.1.1.1 Overview of existing Source engine effects

Among many graphic effects that are implemented in a Source engine following were taken into consideration:

- Lights

The `light, light_dynamic, light_spot` entities[5] are responsible for definition of light object within the virtual world. Static lights objects defines a sources and type of produced light with all required properties like colour,

[5] A object defined within the Source Engine as having characteristics which differentiate it from "the world"

brightness and additional effects (i.e. fluorescent flicker, candle, or strobe). In contradiction, dynamic light is an invisible light source that changes over time. Can be turned on and off through inputs, and can aim at any object, including moving ones. Dynamic lights are calculated on the fly in the game, which means they have a higher processing cost but are much more flexible than static lighting. The third type of light is a cone-shaped, invisible light source. Can be turned on and off through inputs. This is a static spotlight that can be pointed at another entity.

- Fog

As the name states, this effect is responsible for rendering an effect of the fog in the scene. The `env_fog_controller` entity controls fogging in a level. Functionally, by adding fog to a level, a far scene plane can be hidden, the amount of rendered geometry reduced, improving game performance.

- Smoke

An entity that spits out a constant stream of smoke. This effect it is represented in code by class `CSmokeStack`. Its main advantages are great movement features like the wind direction and its strength.

- Cloud of dust

An effect of dust cloud that is represented by `func_dustcloud` entity looks as a best backbone for the visual effect of the radio wave propagation. This object spawns a translucent dust cloud within its volume and has many predefined properties that are useful in required data representation. Among others, the most important are: colour of the cloud, transparency level, size of the dust particles, number o particles produced within specified time, definition of movement within area of the cloud of dust.

In order to develop visual effect that is suitable for system purposes usability tests were required. It was crucial to test their potential under all requirements that were previously specified. The general examination of the effects consists of tests that inspected their graphical and visual possibilities as well as code analysis. Detailed evaluation of the existing effects present in a Source engine is described in the next chapter. As a result of this evaluation a final effect was chosen, which needed additional enhancements to meet all requirements for a visualisation effect previously defined.

4.1.1.1.2 Enhancement of the 'func_dustcloud' entity

The main improvements of the dust cloud entity were connected to control of the once created object in the simulation. First functionality added was responsible for turning on and off the object from every single moment of the simulation. In other words, at every stage of the conducted prediction of the signal propagation user is able to disable the effect what makes it not present. Second improvement allows changing the colour of the dust cloud. Both options are essential when the particular colour or density of the entity interference with other parts of the virtual world. Simply, by pressing a keyboard key user is able to toggle the state of the effect from every moment of the simulation. Changing a colour is a little bit more sophisticated action: user is required to manually type the RGB parameters of the expected colour (for an instance: "`255 0 0`" for a red)

4.1.1.1.3 Final version of the dust cloud effect

Final version of the modified `func_dustcloud` entity in action is shown on the following screenshots from the actual simulator.

Fig. 23 Dense modified dust cloud in action

Fig. 24 Modified dust cloud entity in action

Fig. 25 Modified dust cloud entity in action

4.1.1.2 User interface

In order to provide additional feedback about radio waves signal propagation the 2D map of signal coverage was implemented. In the top right corner, in a small window a map with all necessary information about signal propagation and building layout can be displayed. The map could present previously prepared plan of the virtual environment or display information about current signal coverage in the building.

Moreover, map feature contains information about location of the user in the environment and it is represented by a red dot. It can be used for locating purposes or as an additional feedback about current strength of the signal. What is more, when more than a one user is taking part in the simulation, all of them are visible on the map.

Fig. 26 2D coverage map in a top right corner

Furthermore, there is a possibility to change the type of the displayed information into full screen mode as well. On the full screen version entire map is displayed, in contrast to windowed version where only small area of the map is visible. This functionality could be assigned to one of the keyboard keys, as result user is able to switch between three different modes of displaying required information by pressing specific key.

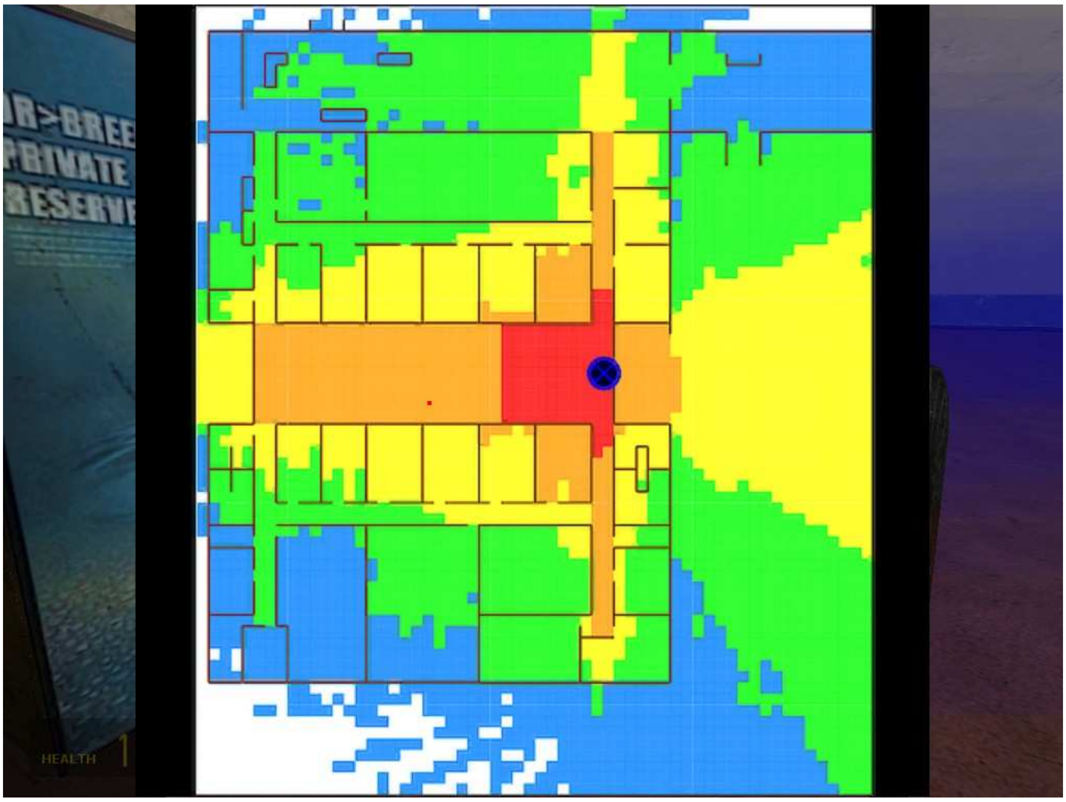

Fig. 27 Entire coverage map of the virtual environment

In addition to windowed map a small panel with additional information was developed. It could be displayed on demand by pressing specific keyboard key during the simulation. The panel contains description of the symbols used in virtual environment. Example content of the window is shown on the figure below.

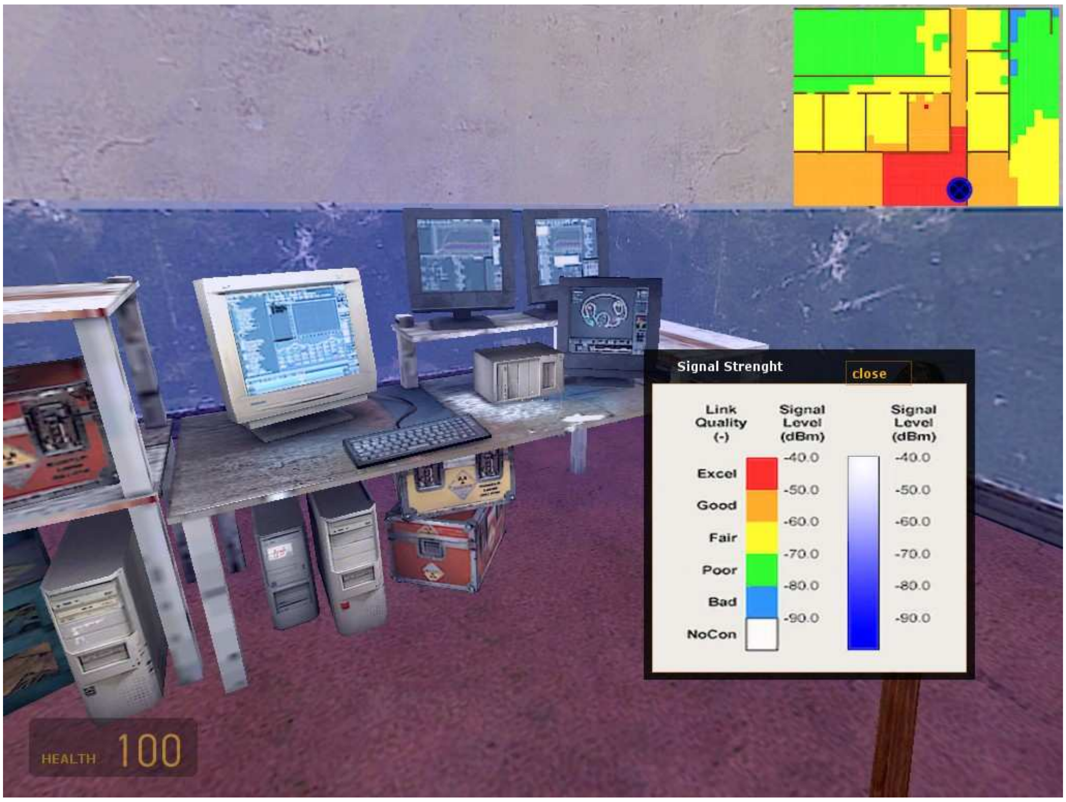

Fig. 28 Hint panel with a description of the symbols used

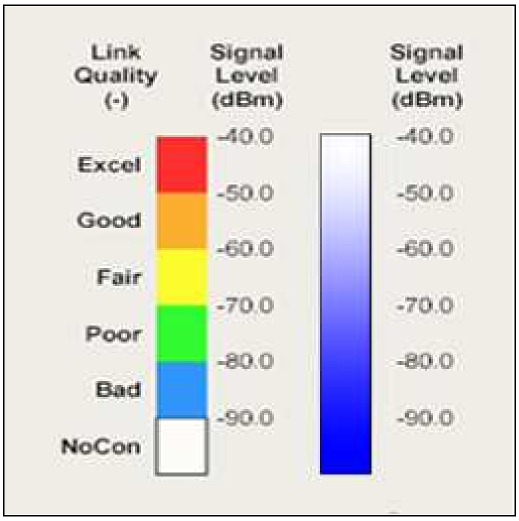

Fig. 29 Content of the information window

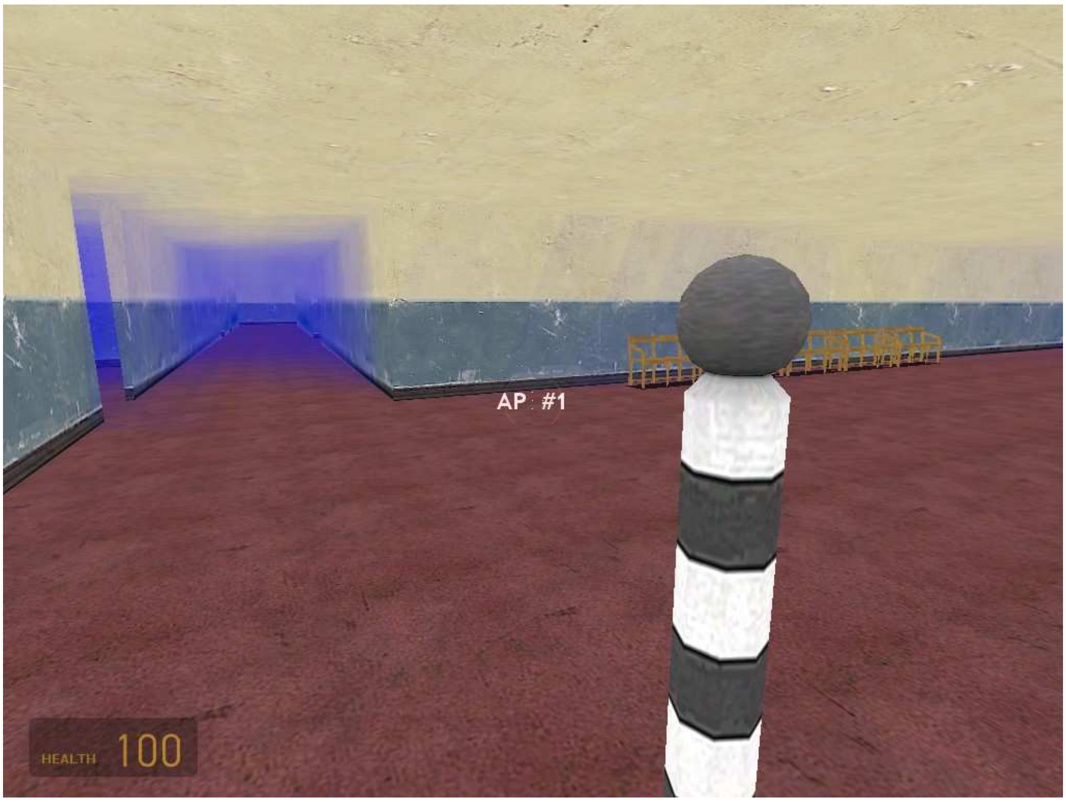

Fig. 30 Name of the access point diplayed in a centre of the screen

In order to simplify recognition of specific base stations a simple functionality of displacing the name of the access point was implemented. During the simulation by looking at the specific access point in a fraction of a second the name of the specific access point is displayed in the centre of the screen. Screenshot from the simulation presented above perfectly describes the problem. In this case a black and white pole represents access point with a following description: “AP #1”.

Additionally the custom main panel of the application was developed (Fig. 31). From this stage user is able to change all necessary graphic and audio options for the simulator as well as create the server for the simulations or basically setup complete simulation.

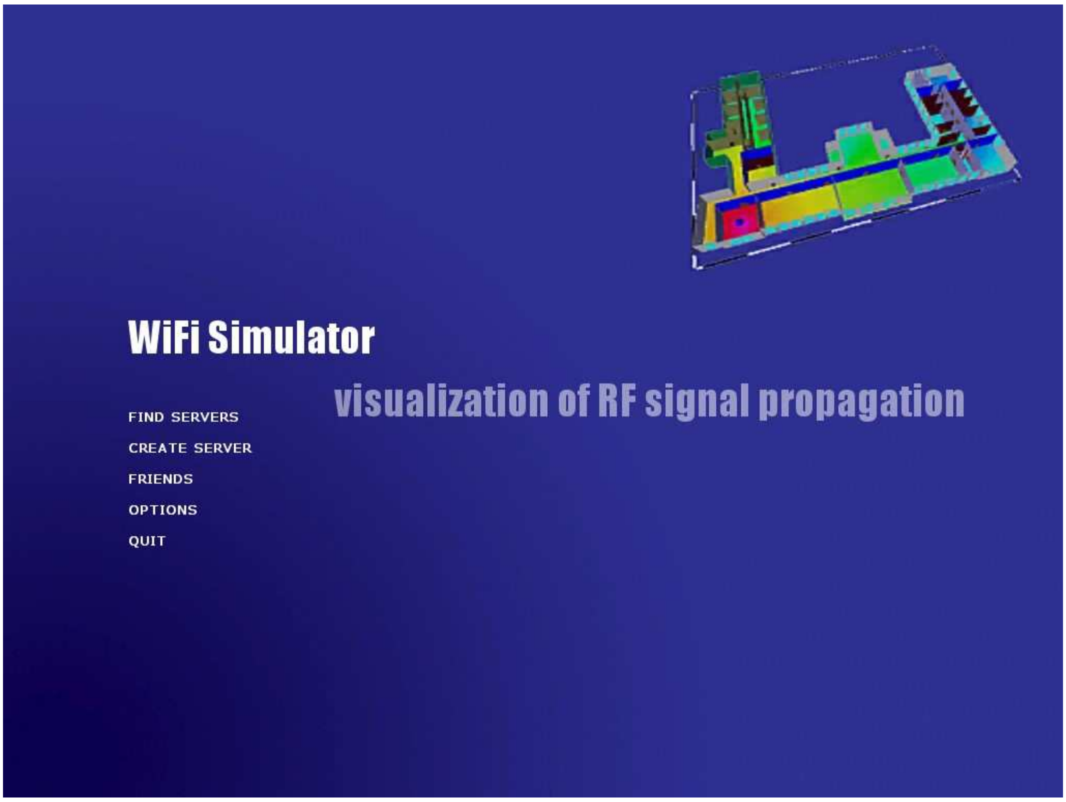

Fig. 31 Main panel of the simulator

4.1.1.3 Additional feedback

Source engine provides incredibly sophisticated mechanism of virtual world simulation when it comes to 3D graphic and visualisation as well as complex sound environment. Not taking an advantage of this would be a big mistake. With the aim of providing additional information about signal propagation sound feedback was used. Throughout the simulation user is equipped with a pseudo Geiger counter that increases intensity of its ‘buzz’ sound when user is approaching closer and closer to the nearest access point.

4.1.2 .SVG to .VMF converter

File format supported by the propagation tool developed by Cork Institute of Technology, which is used in this research, is a .SVG file. Scalable Vector Graphics (SVG) is an XML markup language for describing two-dimensional vector graphics, both static and animated (either declarative or scripted). All input and output files must be compatible with this standard. Building blueprints must be translated into XML files, where the specific tags contains all necessary information about layout of the particular parts of the construction. The only difference after all propagation calculations between the input and output are additional information about signal strength in a particular part of the building, which are indicated by specific XML tags as well. On the following figures difference between input and output files.

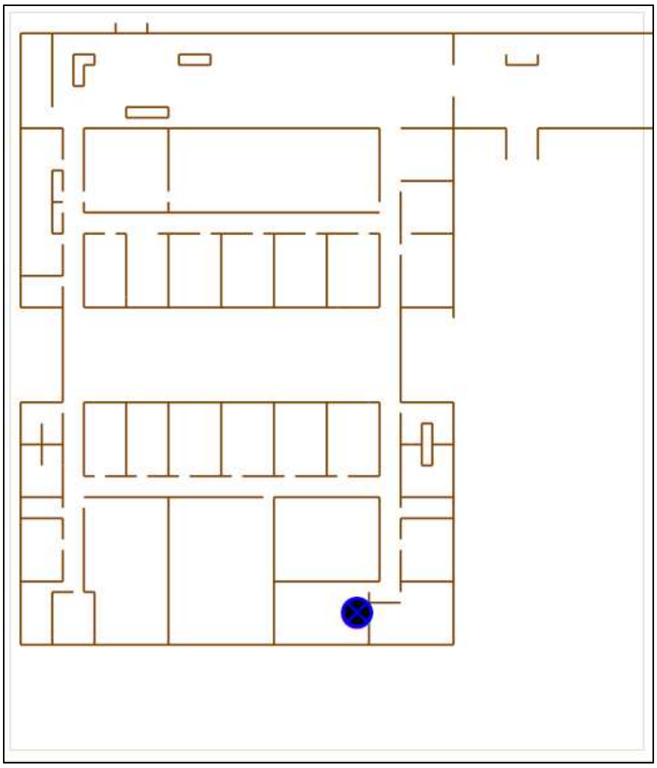

Fig. 32 Input - blueprint of the building with one access point indicated as a circle

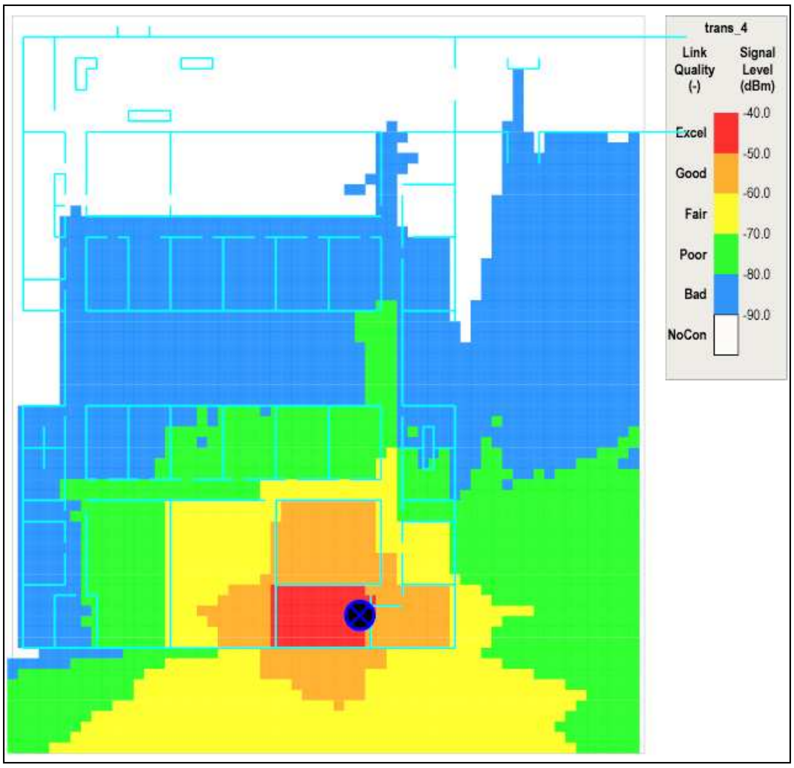

Fig. 33 Output (with prediction of the signal propagation) - blueprint of the building with one access point indicated as a circle

In example, the following piece of code from the .SVG file is responsible for definition of the wall:

```
<line id="wall_130" type="0" x1="6.00" y1="7.00" x2="7.00" y2="7.00" stroke="#804000" />
```

Tag ‘line’ indicates that following line is responsible for drawing a line in the .SVG viewer and in the same time with parameter ‘id’ in the propagation tool stands for a wall of the building. Attributes ‘x1’, ‘x2’ and ‘y1’, ‘y2’ define a coordinates of the wall.

In contrast to Cork simulator, the representation of the world (source code of a map) for Source engine is defined by VMF format, which stands for Valve Map Format. Like any source file it must be compiled to be used and is thus designed to be easy to read and work with rather than optimized for execution. Because of this a .VMF file is written an easy to understand form of coding language common in dealings with the source engine. This is to be the documentation of everything defined within a .VMF file detailing what is what and how it is shown. Later on, with a use of Source SDK tools it needs to be compiled to BSP (Binary Space Partition) format. .BSP is the file extension of the final format that defines maps/levels used by developed simulator.

The following part of the .VMF file is responsible for 3D definition of the plain wall in the simulated environment shown on the Figure 34:

```
solid
      {
            "id" "4"
            side
            {
                  "id" "7"
                  "plane" "(224 -608 160) (256 -608 160) (256 -768
                  160)"
                  "material" "PLASTER/PLASTERWALL005C"
                  "uaxis" "[1 0 0 7] 0.25"
                  "vaxis" "[0 -1 0 0] 0.25"
                  "rotation" "0"
                  "lightmapscale" "16"
                  "smoothing_groups" "0"
            }
            side
            {
                  "id" "8"
                  "plane" "(224 -768 64) (256 -768 64) (256 -608
                  64)"
                  "material" "PLASTER/PLASTERWALL005C"
                  "uaxis" "[1 0 0 7] 0.25"
                  "vaxis" "[0 -1 0 0] 0.25"
                  "rotation" "0"
                  "lightmapscale" "16"
                  "smoothing_groups" "0"
            }
            side
            {
                  "id" "9"
                  "plane" "(224 -608 160) (224 -768 160) (224 -768
                  64)"
                  "material" "PLASTER/PLASTERWALL005C"
                  "uaxis" "[0 1 0 7] 0.25"
                  "vaxis" "[0 0 -1 256] 0.25"
                  "rotation" "0"
                  "lightmapscale" "16"
                  "smoothing_groups" "0"
            }
            side
            {
                  "id" "10"
                  "plane" "(256 -608 64) (256 -768 64) (256 -768
                  160)"
                  "material" "PLASTER/PLASTERWALL005C"
                  "uaxis" "[0 1 0 7] 0.25"
                  "vaxis" "[0 0 -1 256] 0.25"
```

```
                "rotation" "0"
                "lightmapscale" "16"
                "smoothing_groups" "0"
        }
        side
        {
                "id" "11"
                "plane" "(256 -608 160) (224 -608 160) (224 -608
                64)"
                "material" "PLASTER/PLASTERWALL005C"
                "uaxis" "[1 0 0 7] 0.25"
                "vaxis" "[0 0 -1 256] 0.25"
                "rotation" "0"
                "lightmapscale" "16"
                "smoothing_groups" "0"
        }
        side
        {
                "id" "12"
                "plane" "(256 -768 64) (224 -768 64) (224 -768
                160)"
                "material" "PLASTER/PLASTERWALL005C"
                "uaxis" "[1 0 0 7] 0.25"
                "vaxis" "[0 0 -1 256] 0.25"
                "rotation" "0"
                "lightmapscale" "16"
                "smoothing_groups" "0"
        }
        editor
        {
                "color" "0 139 232"
                "visgroupshown" "1"
                "visgroupautoshown" "1"
        }
}
```

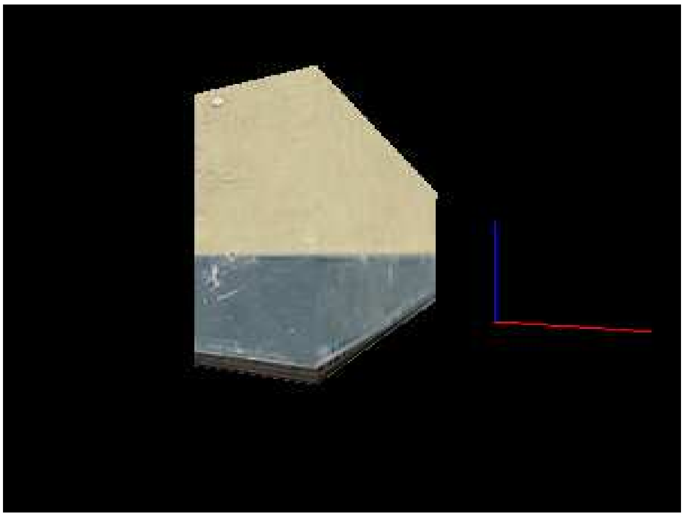

Fig. 34 Wall object in a simulated environment

The main task of this separate application is to convert information encoded in a .SVG file into file supported by the simulator, what in this case is a .BSP, a result of compiling a .VMF file.

Programming language used to develop this part of work was a JAVA language, mainly because of its built in support for parsing XML format. Eclipse SDK Version: 3.1.2 environment was used as a programming platform for JAVA. In relation to XML parsing a DOM[6] approach was used.

Fig. 35 Result of the .SVG to .VMF converter - screenshot from the simulator

In order to produce fully functional environment for the simulation application examines output of the Cork simulator and searches for the specific patterns by parsing XML information encoded in a .SVG file. Next, with a use of predefined objects creates theirs equivalents in a .VMF file. Finally, to create a .BSP files an output of this application is compiled with use of Source SDK tools.

---

[6] Document Object Model (DOM) is a description of how an XML document is represented in an object-oriented fashion.

4.1.2.1 Class diagram

The following diagram presents entire structure of the application.

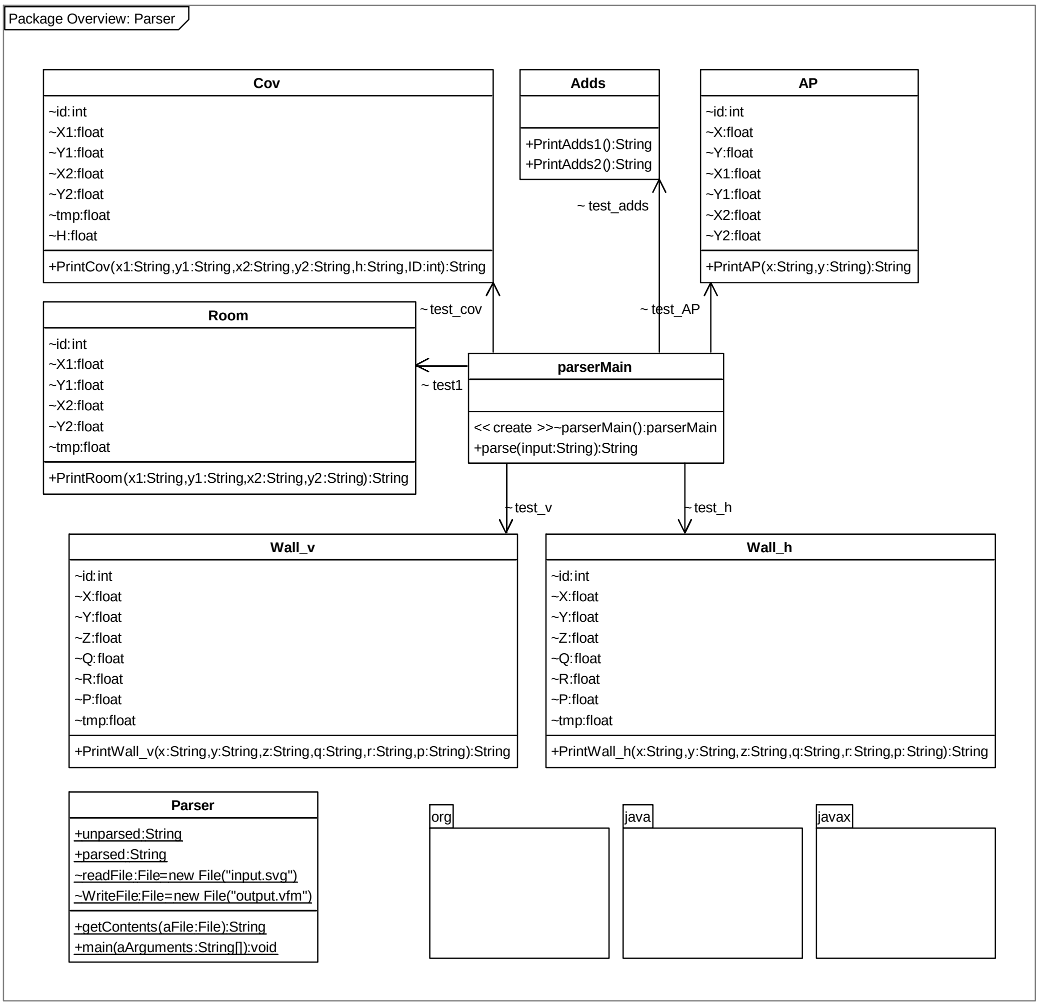

Fig. 36 Class diagram of the .SVG to .VMF converter application

4.1.3 Test map

For testing purposes it was necessary to manually create an independent map of the world for the simulation. In order to build representation of the real world a HAMMER editor was used. Editor produced by Valve, provided as a component of the Source SDK, is the official Source mapping tool. Apart from the construction of level architecture, it is also heavily involved with creating level events and scripting. As a backbone of the map output generated by .SVG to .VMF converter was used.

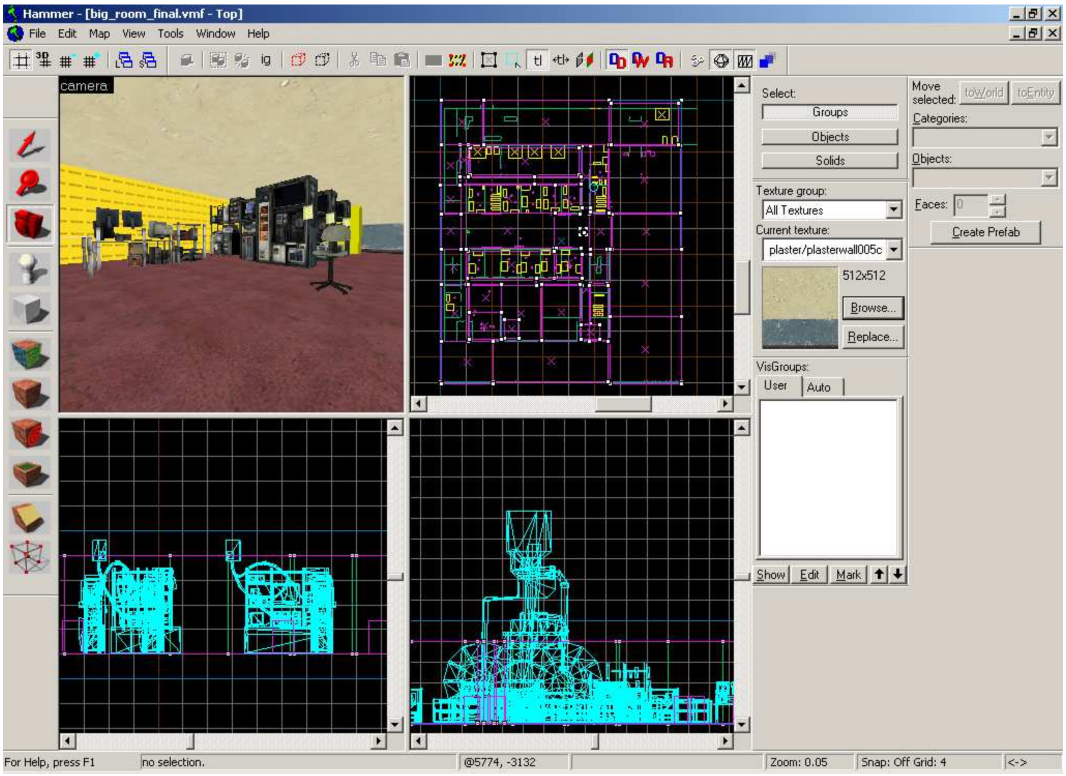

Fig. 37 Creating a testing map in HAMMER editor

4.2 Final solution

Previously described components, a WiFi simulator based on a Source engine and .SVG to .VMF converter, as well as manually produced map create entire system. Figure 38 describes the sequence of steps and activates that are required to conduct complete simulation of the signal propagation.

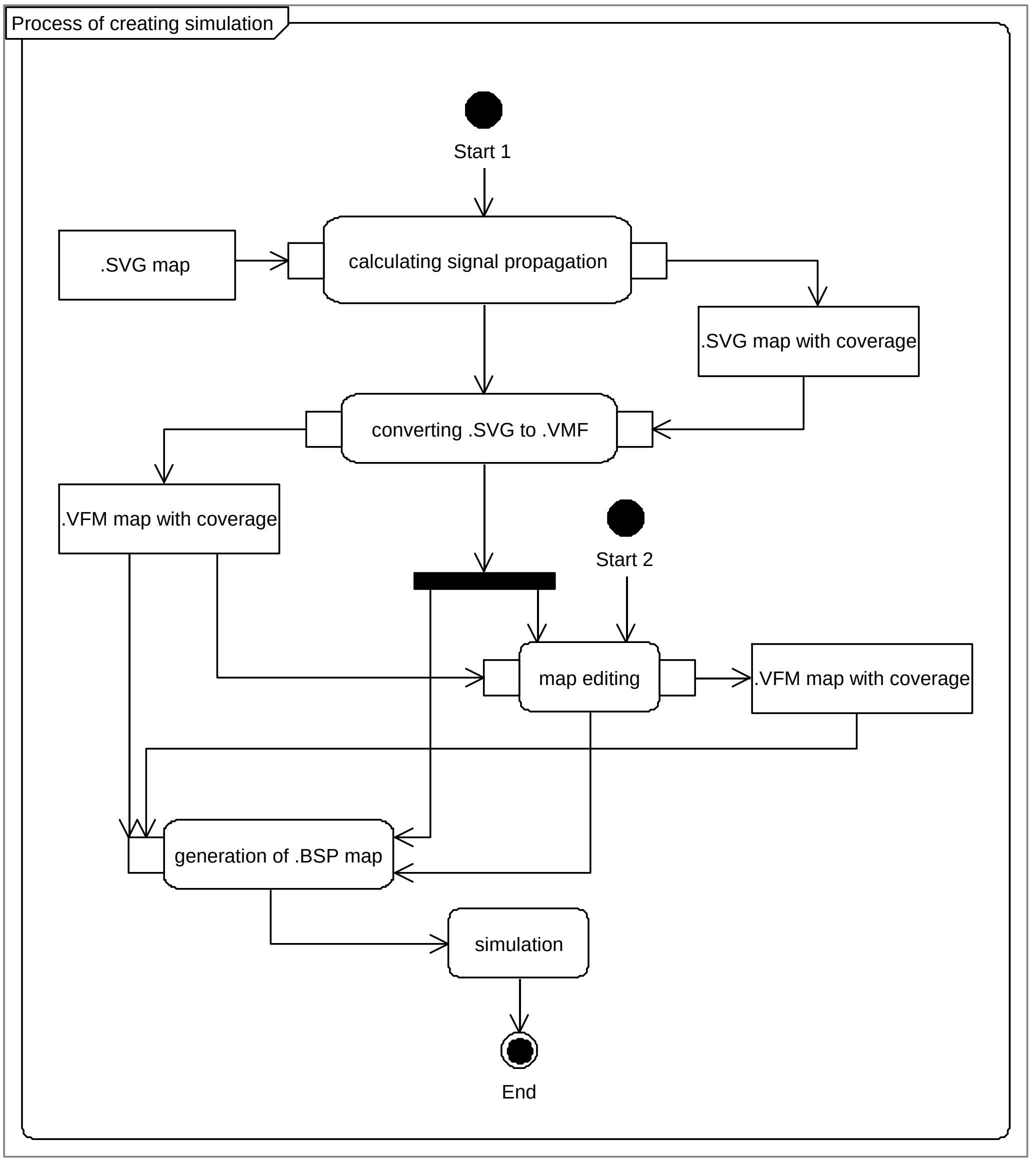

Fig. 38 Activity diagram of creating the simulation for final version of the simulator

In order to begin a blueprint of the building with layout of the base stations in the .SVG format is required. Next, the prediction of signal propagation with all necessary calculations is handed over to the external application, in this case a propagation tool developed by Cork Institute of Technology. The .SVG file with encoded information about signal strength is produced as a result of this operation. After that, it is necessary to convert received file to a file format supported by developed application. This action involves two steps. The first one uses a .SVG to .VMF converter, where as a result a map supported by Source SDK tools is produced. Second step could involve a HAMMER world editor when introducing any changes to the final version of the map are required, or simply with a use of SDK tools it could be directly compiled into final version. As a result of all steps previously described a file in a .BSP format should be produced. Finally, probably the most important step could be carried out: the simulation. By opening the generated file with a building layout and information about signal propagation user is able to move into virtual world. With the final step the entire freedom of action is in the users hands. From this stage the user is allowed to reuse previously created maps.

4.3 Conclusion

In this chapter the implementation of WiFi Simulator has been described. A high-level view of the relationships between system components was presented. Following this each of the three components were discussed in isolation, starting with the core 3D simulator, followed by the .SVG to .VMF converter and finally the creation of custom, independent map. Chapter 5 presents results, evaluation and a discussion of the developed simulator.

Chapter 5: Results, Evaluation & Discussion

This chapter presents results, evaluation and a discussion of developed simulator. As a system developed consist of two main components the evaluation process was divided in the same manner. The first part of this chapter is connected to the evaluation of the core simulator, where the second contains an evaluation of the separate application responsible for converting different formats of the maps used in simulations. This part contains short considerations about the security of the system as well, to provide overall idea of this important problem. The chapter finishes with a discussion relating the design and implementation of the simulator.

5.1 Hardware used

All tests and entire development and implementation of the system and its components were conducted on a personal computer with following configuration:

General information	
Operating System	Windows XP Professional (5.1, Build 2600) with Service Pack 2
Processor	AMD Athlon XP 2500+, MMX, 3DNow
Memory	512MB RAM
DirectX Version	DirectX 9.0c (4.09.0000.0904)
Display Devices	
Card name	NVIDIA GeForce4 Ti 4200 with AGP8X
Display Memory	128.0 MB
Current Mode	1024 x 768 (32 bit) (60Hz)
Driver Version	ForceWare Version: 84.21
Others:	
Microsoft Visual Studio .NET 2003 Version 7.1.3088	
Eclipse SDK Version: 3.1.2 with Java 2 Runtime Environment, SE (build 1.5.0_04-b05)	

Table 1 System information

5.2 Evaluation approach

Due to the nature of the system and constant requirement of evaluating freshly developed components of the simulator and its visualisation effects the entire process of evaluation was conducted simultaneously during the implementation process. Each of the newly developed features required to be checked against overall system's functionality and compatibility with existing components.

However, after finalising a development of the simulator, in order to evaluate and check the overall functionality and usability of the system a series of tests were conducted. There were mainly focused on a carrying out a regular simulation based on plans of the environment with an initial layout of the wireless access point as well as previously developed maps.

5.3 Evaluation of the signal propagation effect

Source engine contains a great number of graphic effects that could be used to illustrate radio wave signal propagation. Most of them are described in section 4.1.1.1 of the previous chapter. In order to choose the final one, which was used in a developed system, an evaluation needed to be carried out. All effects were tested against following criteria:

- visualisation impact: if effect could be used in terms of representing radio wave propagation
- flexibility: how easy the effect could be implemented; does it require changes in a source code and if yes how advanced are they
- performance load: if the effect cause a visible overhead in terms of system performance

5.3.1 Existing Source engine effects

5.3.1.1 Lights

Following screenshots show different lights effects that are built in Source engine.

Fig. 39 In-game effect of light

Fig. 40 Beam of light effect

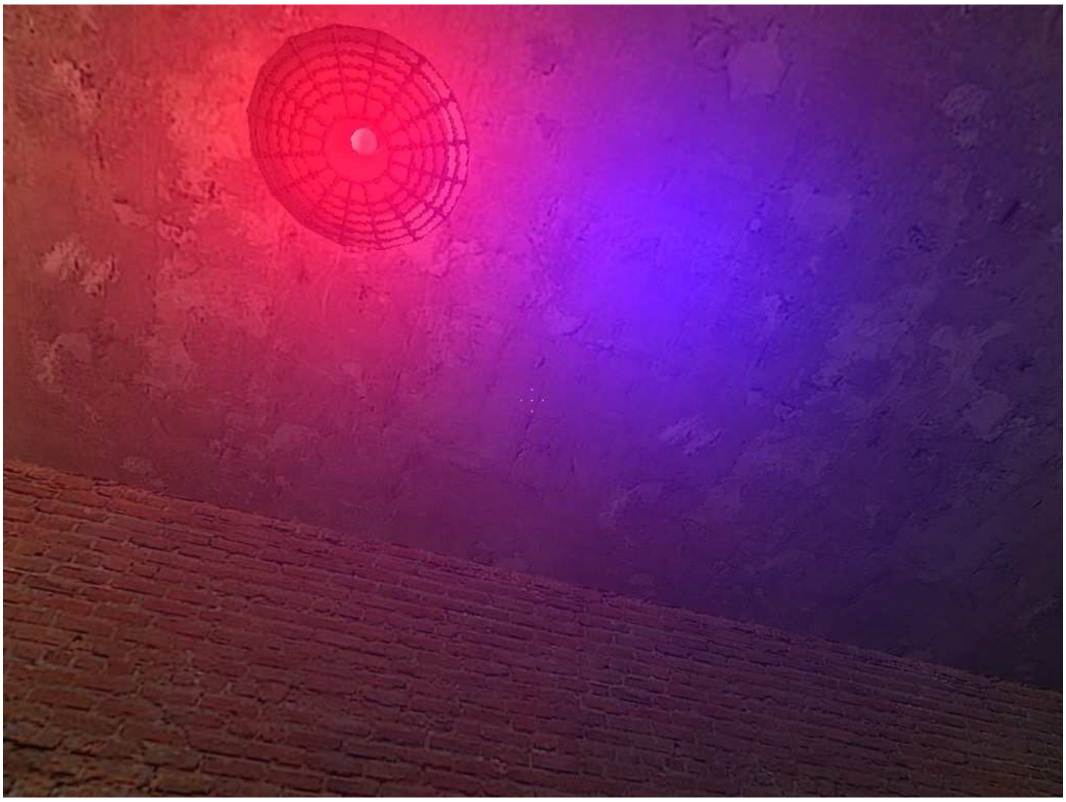

Fig. 41 Two overlapping sources of blue and red light

Fig. 42 Combined beam of white and blue light with random 'dust' particles

In relation to simulation of wave propagation using light entity as a target effect could cause a lot of ambiguities, mainly due to required use of those types of entities as a standard source of lights in the simulated environment. What is more, the control over the shape of those objects is very limited. Extension of the existing effect to fulfil all requirements would require advanced and deep changes in a source code. Moreover, rendering of the large number of lights entities is costly and could affect overall performance. In conclusion, this type of visual effects cannot be used.

5.3.1.2 Fog

Despite the fact that the graphic effect of the fog implemented in a Source engine meet the visualisation requirements (Figure 43 & 44) it cannot be used due to the lack of the possibility to define the area of operation. What is more, only the first instance of this entity in a level is used. As a result, this effect looks exactly the same for the every single position in the simulated environment. This facet disqualifies this entity from use while various types of the effect are required in a scene, even when this entity cause average performance load of the system.

Fig. 43 Fog effect in an indoor environment

Fig. 44 Fog effect with changed colour to red

5.3.1.3 Smoke

Smoke effect is responsible for spiting out a constant stream of smoke. It is represented in code by class `CSmokeStack`. Its main advantages are great movement features like the wind direction and its strength. Unfortunately, this attribute makes it sometimes hard to control and predict. Moreover, control over the shape of the produced smoke is limited. Another drawback of this effect are huge processor resources needed to render. All presented arguments are strongly against using of this effect as a final solution.

Fig. 45 Two sources of smoke: yellow and red

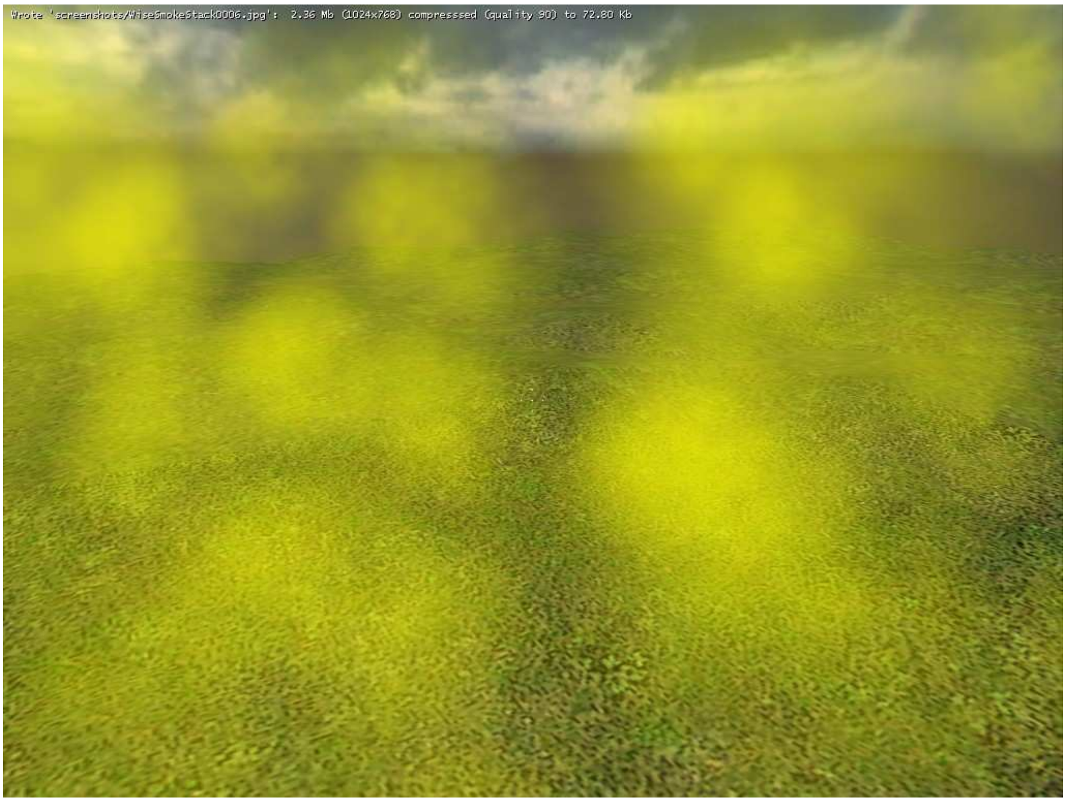

Fig. 46 In the cloud of yellow smoke

5.3.1.4 Dust

The last evaluated effect, cloud of dust, looked to be a silver bullet solution for the problem. This graphic effect responsible for representation of the dust in the game gives a biggest freedom of action. Entity representing dust allows easily controlling created object in a ground of shape, colour and density. What is more, it can be easily used in a multiple form in the same environment and every single item could be controlled separately. Even areas occupied by entities can overlap each other. Moreover, code of this effect required the fewest number of changes. In relation to system resources, this effect can be treated as a medium one. Even when it requires the same processor power as a fog effect, it wins fulfilling better other requirements. Definitely it is an overall winner.

Fig. 47 Cloud of dust with modified colour: blue

Fig. 48 Large overlapping clouds of dust

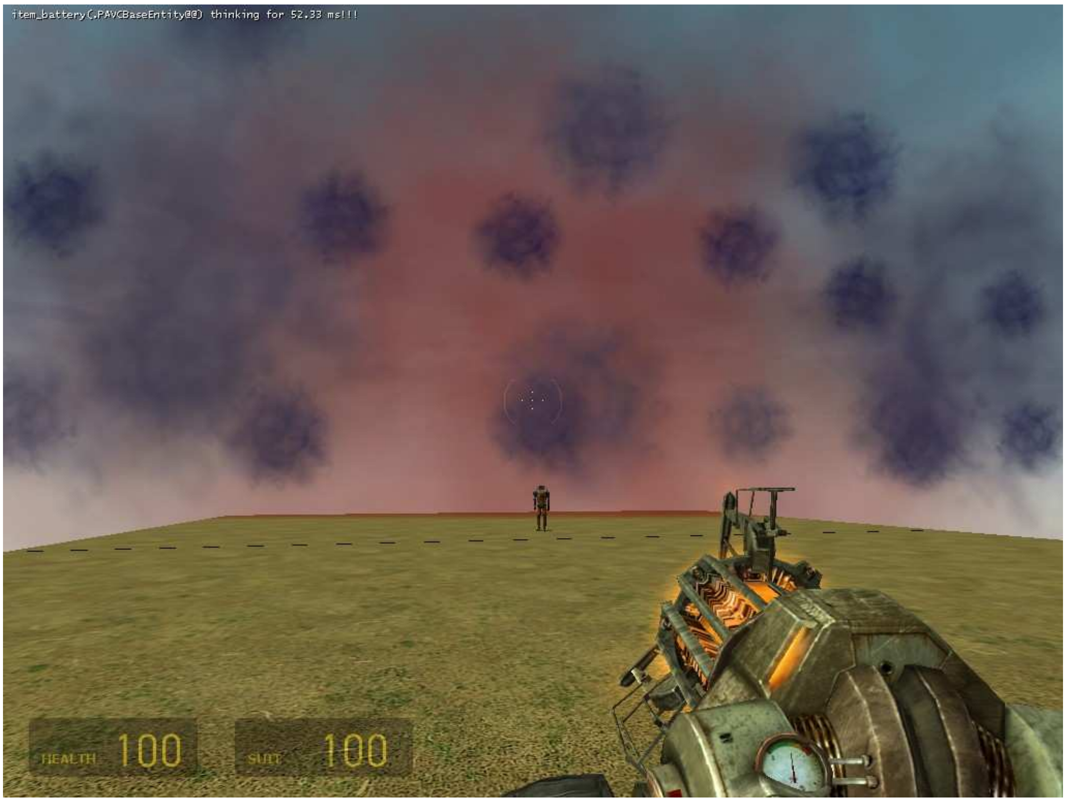

Fig. 49 Clouds of dust in different sizes

5.3.2 Summary

General overview of the results on evaluated effects is presented in a table below. On such a breakdown it is clearly visible that the effect of dust is a best possible solution.

requirements / effects	visualisation impact	flexibility	performance load
lights	good	medium	medium/high
fog	good	low	medium
smoke	medium	low/medium	high
dust	very good	high	medium

Table 2 Existing effects evaluation overview

5.4 Evaluation of the modified dustcloud effect

In result of previously described evaluation as well as development and implementation steps described in a section 4.1.1.1 of the Chapter 4 a completely new effect was developed.

After a running a different simulations it can be clearly stated that implemented effect fully meet all requirements defined at the beginning of this dissertation. The most important factor is that the effect represents visually all information about propagated signal and in the same time it is not obtrusive in any way. What is more important its presence does not disturb in a general users movement in a virtual world. In addition, its representation is unconsciously associated in users mind with its functionality.

After all, its enhancement in a source code do not affect performance load in contrast to its predecessor being in the same time flexible in use.

5.5 Evaluation of the .SVG to .VMF converter

Second important component of the system is responsible for conversion of the .SVG maps into .VMF file format. In order to evaluate this application a several of tests were conducted. There were simply based on converting set of .SVG maps into .VMF, which were revived in a HAMMER editor at the next stage, and finally run in the final version of the actual simulator.

Generally, all evaluated maps were correctly translated and there were no problems with opening them with HAMMER editor. Converter was able to parse the entire XML structure of the .SVG files without any errors and as a result produce an output file. Time of the entire process of the conversion depended mostly of the size and level of sophistication of the .SVG file. On the machine with a configuration described at the beginning of this chapter conversion last from a period of a few seconds with a simplest maps to a few minutes with a plans covering very large areas.

The main characteristic that was clearly visible at the first sight was an artificial look of the generated environments. Simplicity of the .SVG files, which is limited to encoded information about layout of the walls in the building and coverage of the signal leads to ascetic look of the simulated environment. That is why a next conclusion is easy to draw: when we want to conduct the simulation of the detailed environment that reflects surrounding world we need to edit maps manually with a use of HAMMER world editor. However, to make that process easier the backbone generated by evaluated application can be used.

Furthermore, lack of information about scale of the blueprints provided in a .SVG files is a small disadvantage. As a result, assumptions about scale and height of the walls need to be made at the beginning of the conversion. This problem could be simply solved by specifying a simple parameter responsible for a scale factor at the beginning of each conversion process.

Finally, at the end of the tests an important problem was revealed. Due to a structure of the .SVG file the signal coverage is described by covering an entire map with small squares with particular signal strength and coordinates. As a result, every single square is translated into entity in the .VMF file that is responsible for visualisation of the signal propagation.

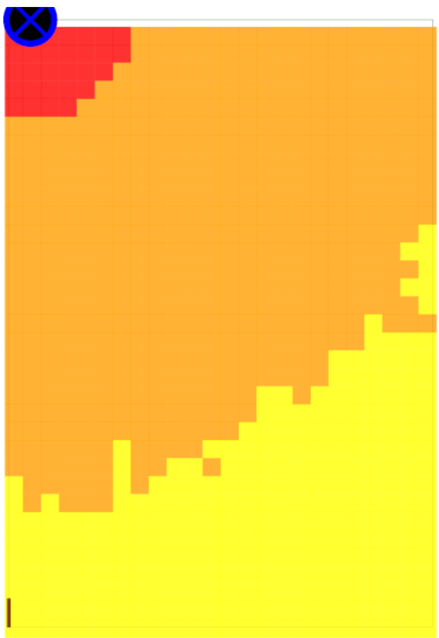

Fig. 50.SVG file with coverage information encoded

Conversion of the file shown on the Figure 51 is visible on the following screenshot form the HAMMER editor. Map is covered with a large number of small entities responsible for representing strength of the signal.

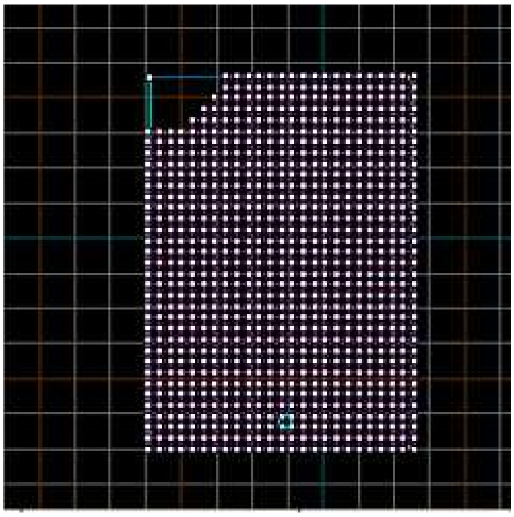

Fig. 51 VFM file: result of the conversion; view from a HAMMER editor

The problem occurs in an actual simulation process when converted .SVG file contains enormous number of entities (Fig. 52), what actually happens with maps that enclose information about very large indoor environments. As a result, during the simulation process errors in a display of effects are visible (Fig. 53). In other words, a Source engine is not capable of displaying correctly such a number of similar entities of a dustcloud within small area in the same time.

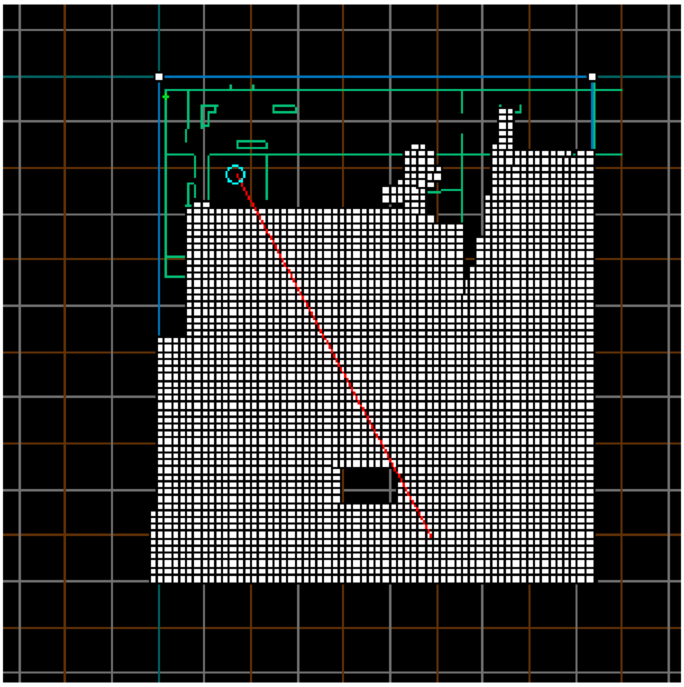

Fig. 52 Enormous number of dustcloud entities visible on the map; view from a HAMMER editor

In order to solve described problem an optimisation algorithm should be implemented. It should be responsible for merging objects with a similar signal strength, what in a result should lead to creation of a few entities that will cover larger areas of the map. Proposed output of such a algorithm is outlined on the Figure 54.

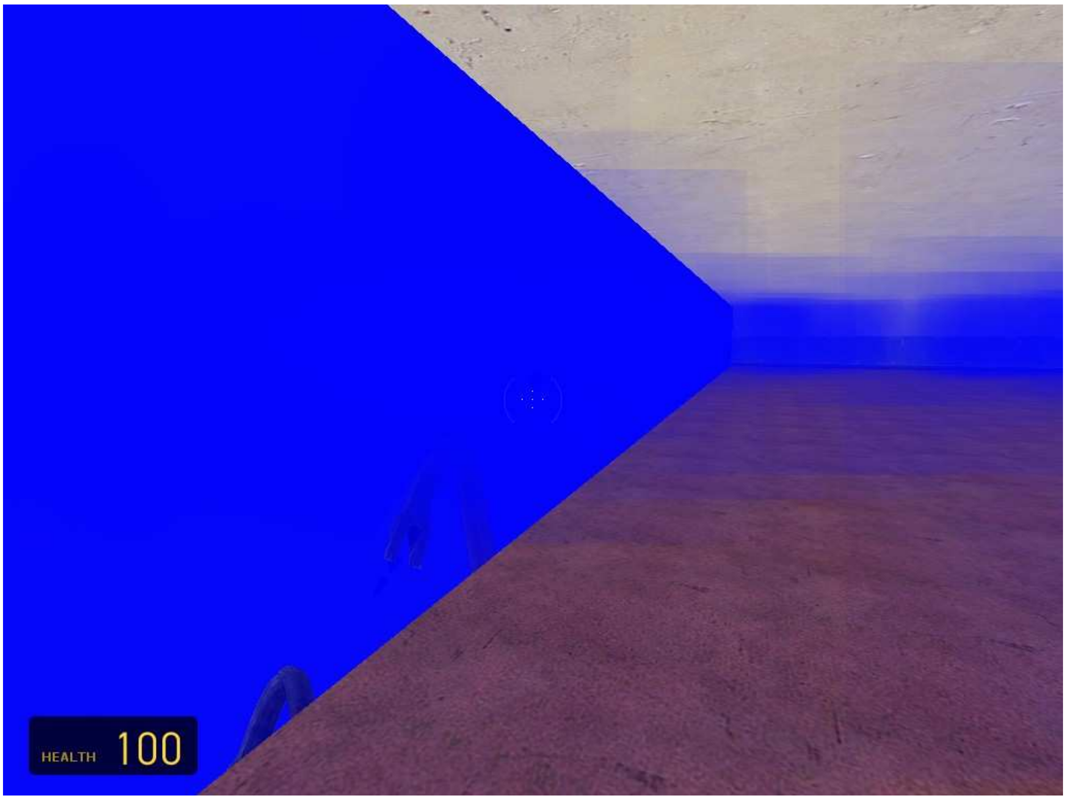

Fig. 53 Errors in a rendering of big numbers of entities in final version of the simulator

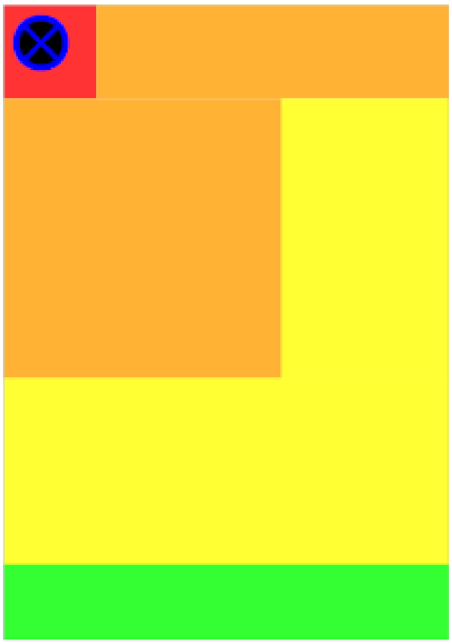

Fig. 54 Vision of proposed solution of the optimisation algorithm

5.6 Evaluation of remaining components of the system

Apart of main purpose of the core application, which is visualisation of the signal propagation, and second important component: the .SVG to .VFM converter there are a few other aspects of the system that should not be omitted in an evaluation process.

The first one is an additional feedback provided during the conducting the actual simulation. Sound system, that works in a similar way to standard Geiger counter found to be very helpful. On the level other that graphic descriptions it visualises the problem with the use of other important sense which is hearing. Even the simple text description of the access point that was displayed after looking at the object allowed for an additional orientation in the environment.

Moreover, simple user interface allowed for a conducting all of the necessary activities within virtual world. Especially useful here occurred to be a small 2D map of the layout of the building with an embedded signal coverage. When the primary effect of the visualisation caused an ambiguities in the interpretation of the fact there was always a traditional form of information about signal strength.

5.7 Vision vs. Final Version

The following section compares the original objectives as set out in Chapter 1, with the final deliverable presented in chapters 3 and 4.

- **Allow to conduct simulations in a realistic 3D environment.** The developed simulator completely fulfil this requirement by fully exploiting the possibilities of the Source game engine that allows for development of realistic environments limited only to our imagination

- **Allow to conduct simulations of wave propagation within multi-floor indoor environments.**

  Developed core of the final system allows for carrying out all necessary simulations within multi-floor environments. The only restriction in this area is combination of the Cork simulator used for calculations of the signal propagation and converter of .SVG files into .VFM format, where they simplicity allows only for conducting predictions within one-level indoor environments. As a result, a multi-level simulations are only limited in a process of caring out of the simulation from the early phase. There is always a possibility to manually develop a multi floor environment.

- **Visualise RF signal propagation with use of special 3D graphic effects (e.g. fog or mist)**

  This requirement is fully realised by the existence of the modified version of the one of the basic graphic effects implemented in a Source engine. Detailed evaluation of this aspect was conducted in a beginning of this chapter.

- **Allow for interaction with the surrounding environment and be capable of dynamic changes within it.**

  Due to the limitations of the Source engine this goal was only partially achieved. Systems allows for an user interaction with a surrounding world (by opening doors, moving objects, switching lights on and off, etc.), however all those actions does not have a direct impact on a visualisation of the signal propagation. Architecture of the used game engines is hardly flexible in relation to dynamic changes. All elements of the system need to be previously prepared in order to use in a simulation. The best example here are textures: in order to use the texture in an application they need to be converted into standards supported by the engine.

- **Must be usable, in particular, this means a straightforward initial setup procedure and an easy mechanism to configure and run simulations and produce results on the spot.**

  WiFi Simulator easily fulfil of those requirements. Regular personal computer with a hardware allowing for running a Half-Life 2 game could be used as a platform for developed system. The only limitation so far is a delegation of the prediction procedures to remote application, currently a Cork simulator. The silver bullet solution for this problem would be a complete integration of the propagation tool with a developed system by including all delegated procedures in the source code.

- **Allow for a 'multi-client' approach (client-serve architecture).**

  Embedded client-server functionality in the Source engine delivers all necessary architecture for the multi-user approach.

- **Allow for a possible integration with other applications, like context aware systems.**

  Technology used to develop this system widely opens a possibilities of integration with other environments. As a result, integration of such a system could extend their functionality.

5.8 Security considerations

Regarding security concerns of developed application we can divide the problem into three different facets:

- Support of the local network security analysis provided by the application
- Security concerns of the core application
- Security concerns of the system (including use of 'Cork simulator')

The nature of the signal propagation simulator is strictly connected to the security analysis of the wireless networks. There are several security issues with protocols used in indoor environments. Most importantly, WEP and Shared Key are optional, and turned off by default in access points. If these protocols are not turned on in even one access point, it is trivial for hackers to connect to the network, using standard wireless cards and drivers. The 802.11 signal can travel surprisingly large distances from the access point, often a thousand feet or more, allowing the hackers to connect from outside the building, such as from a parking lot, or from the street, (leading to the term "drive-by hacking".) If, as is often the case, the wireless network is connected directly to a corporate intranet, this gives the hackers direct access to the intranet, bypassing any internet boundary firewalls.

The problem of "open" access points is made more difficult due to the low cost and easy availability of access points, and the difficulty of detecting them. It is not uncommon to find individuals or groups within a company who have installed "rogue" access points without the knowledge of the normal networking group, and without properly configuring the access point. These rogue access points are often difficult to detect with normal network monitoring tools, as access points are normally configured as invisible bridges.

Solution for this problem is proactive analysis of the future network layout before actually building the network. At this stage a simulator can help. With a prediction of the signal propagation we can avoid unfortunate setup where signal is leaving our premises, badly placed access point, etc.

When it comes to the application itself we have to distinguish its two parts: core application and entire system. Here the small explanation is needed: a perfect target version should be an independent standalone application which will consist of two parts: mathematical, responsible for computing signal propagation and visual, responsible for data representation. Current version differs from the target one by delegating mathematical calculations to the remote service.

In relation to application security, it is completely delegated to Valve Software with a believe that their environment is a secure one. As a result, an output of automated process of mode creation is believed to be a secure application, resistant to hackers that are using new techniques to gain access to sensitive data and administer other malicious activities aimed at the software application.

Of course, we cannot forget about attacks on a released, final version of the application. There is always a possibility that the system could be hacked or cracked later on. In a process of reverse engineering rivals are able to get a source code of an application. At this stage they are able to change source code and release an evil version. Simply, algorithms of calculating signal propagation could be changed. As a result, our application will produce outputs full of errors what will lead to cease using product by customers – they will simply think that the application is badly written. What is more, when rivals have a source code they are able to do everything with an application. They can release it as their own version, add malicious sub-applications (i.e. spyware) and collect fragile data.

Moreover, very simple types of attacks, that do not involve any sophisticated changes in a code, are possible as well. They are only limited by ‘hackers’ imagination. For example input files (maps, blueprints, etc.) could be swapped before entire process of computing signal propagation. As a result user will be sure that this is an applications fault to produce strange outputs. Unaware of changes on input files he will never think that the calculated output is correct for an input provided.

All problems described before are common for both type of application mentioned at the beginning. The version that uses remote server to invoke procedures to calculate signal propagation what introduces vulnerabilities. All mathematical operations connected to predicting RF signal propagation are delegated to remote services. At this stage all vulnerabilities are inheritance of web services problems.

- Privacy: For many services it is important that messages are not visible to anyone except the two parties involved. This means traffic will need to be encrypted so that machines in the middle cannot read the messages.

- Message Integrity: This provides assurance that the message received has not been tampered with during transit.

- Authentication: This provides assurance that the message actually originated at the source from which it claims to have originated. You may need to not only authenticate a message, but also prove the message origin to others.

- Authorization: Clients should only be allowed to access services they are authorized to access. Note that authorization requires authentication, because without authentication, hostile parties can masquerade as users with the desired access.

Moreover, there is always a possibility to perform a DoS attack so the server will not be able to response at all. To limit a possibility of such accidents an encrypted communication (SSL/HTTPS) instead of plain HTTP should be used. What is more, authentication could be another step to make an attack harder to conduct. In addition digital certificates from a well-known trusted authority could be implemented as well.

As the world increases its dependency on computers for critical information, the chances of applications being attacked are also increasing. Network security is no longer enough to secure an application. Security needs to be a part of the application design process. Implementing security during the design phase ensures that security is being designed into the application, thus decreasing the risk of an attack. Moreover, meeting the challenges of security needs requires knowing what the security needs and priorities are, what technologies can be used to achieve them, and above all, thinking clearly about system's weaknesses.

5.9 Conclusion

In conclusion, the results presented in this chapter, supported by conducted tests, show that a usable and flexible tool has been developed. The project objectives as laid out in chapter 1 have been satisfied. The final chapter outlines some possible paths for future work concerning this finding.

Chapter 6: Conclusions

This chapter describes ideas for future development of the system followed by overall conclusions about the design and success of the project.

6.1 Further Work

While some of the questions posed by the research objective have been answered, as work on this dissertation has progressed further avenues for research and evaluation have emerged. In this section, a further work is presented by the components of the system and their future extensions.

6.1.1 Core application

- Development of the new user interfaces features.

In order to extend existing functionality of the system and make the process of the simulation easier additional user interface feature should be developed. It could involve new panels/windows containing additional information as well as shortcuts or other widgets that could shorter current users operations.

- Development of the new world models

In order to make simulation process more realistic additional models of the surrounding environment should be designed and implemented. Models of elements of the existing networks or simply objects of everyday use are among them.

- Extend sound feedback

Currently implemented sound feedback is only a foretaste of the possibilities that lies beneath the powerful sound environment implemented in a Source engine. As a result it should be fully exploited to use its entire potential.

- More detailed support for multi-floor environments

Unfortunately, as it is stated in a evaluation chapter a support for multi-level buildings need to be extended, as in a current form the system provides only a core technology for simulations. As a result, an integration with more sophisticated planning tools is necessary. Contrary, instead of delegating all calculations to separate application an entire prediction component should be a part of a core application.

- Support for dynamic changes

As it is described in an evaluation chapter systems partially suffer for a lack of support for dynamic changes. More research in a addition of that support into a Source engine is a must in this situation.

- Integration with a context-aware adaptive system

It was already established within this work that integration with a ubiquitous simulators is essential. Currently, existing simulators of the pervasive environments does not simulate any aspect of the communications networks that must support any operational ubiquitous computing environment. Especially, where the vast majority of these types of environments are based on wireless technologies.

6.1.2 .SVG to .VMF converter

- recognition of different types of objects

In order to extend a possibilities of this application it is necessary to extend a number of objects that are recognised by the converter. By such a enhancement, produced output will be much more detailed, and in result more attractive for a user.

- optimisation algorithm

Problems described in a section 5.5 of the evaluation chapter an optimisation algorithm is vital extension of this component of the system. Developing this type of algorithm obviously requires more research in this area.

6.2 Conclusions

At the final stage of this research it could be easily stated that this project has realised a WiFi Simulator, a unique combination of a wireless system planning tool and a popular game, that has been developed to support engineers, network administrators, researches and regular wireless system users testing and analysing radio wave signal propagation in a virtual 3D environment.

In conclusion, WiFi Simulator fulfils the objectives presented in chapter 1 as follows:

- Allow to conduct simulations in a realistic 3D environment.
- Allow to conduct simulations within multi-floor indoor environments.
- Visualise radio wave signal propagation with use of developed 3D graphic effect
- Allow for interaction with the surrounding environment
- It is simple and intuitive in use and allows running simulations and producing results on the spot.
- Provides client-server architecture
- Allow for a possible integration with other systems

Abbreviations

WiFi - Wireless Fidelity

3D - Three dimensional

FPS - First Person Shooter

RF - Radio frequency

LOS - Line of sight

dB - Decibel

ITU - International Telecommunication Union

GSM - Global System for Mobile Communications

GPRS - General Packet Radio Service

EDGE - Enhanced Data rates for GSM Evolution

IEEE - Institute of Electrical and Electronics Engineers

SCDMA - Synchronous Code Division Multiple Access

GUI - graphical user interface

WLAN - wireless local area network

HLSL - High Level Shader Language

NPC - non-player character

DoS - Denial-of-service attack,

Bibliography

[1] *Id Software, Inc* http://www.idsoftware.com

[2] *Unreal Engine 3, Epic Games*
http://www.unrealtechnology.com/html/technology/ue30.shtml

[3] *Introduction to RF Propagation*, by John S. Seybold,
2005 by John Wiley & Sons, Inc

[4] *Wireless Communications - Principles & Practice*, by Rappaport, Theodore S
IEEE Press, 1996

[5] ITU-R Recommendations, *Propagation Data and Prediction Methods for the Planning of Indoor Radiocommunication Systems and Radio Local Area Networks in the Frequency range 900MHz to 100GHz*, ITU-R P.1238-2, Geneva, 2001

[6] *Wireless Communications Principles and Practice*, 2nd ed., by T. S. Rappaport
Prentice-Hall, Upper Saddle River, NJ, 2002,

[7] Engine Licensing, information Sheet; 2004 Valve Corporation

[8] The Computer for the Twenty-First Century Mark Weiser Scientific American, 1991,Vol 265, No. 3, pp. 94-104

[9] *A Testbed for Evaluating Human Interaction with Ubiquitous Computing Environments*, E. O'Neill, M. Klepal, D. Lewis, T. O'Donnell, D.O'Sullivan, D. Pesch, in Proc 1st International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (TRIDENCOM'05), Trento (Italy), February 23 - 25, 2005, pp 60-69

[10] *A Platform for User-Centred Evaluation of Context-Aware Adaptive Services*.
by O'Neill, E., Lewis, D., (2005)
Proceedings of 5th Annual Information Technology & Telecommunications Conference (IT&T 2005)

[11] *Valve Software SDK*, http://developer.valvesoftware.com/wiki/Main_Page