

SEVENTH FRAMEWORK PROGRAMME
THEME – Energy Efficient Buildings
EeB-ICT 2011.6.4. ICT for energy-efficient building and spaces
of public use



Self learning Energy Efficient building and open Spaces

GA No.285150

**D3.5 Positioning system and Physical Network
Design Framework**

Work Package	WP3 – Wireless intelligent sensors and actuators network (WISAN)		
Task	Task 3.5 – WISAN Deployment Framework		
Revision	0		
Due date	2013/07/31	Submission date	2013/08/02
Dissemination level	PU	Deliverable type	R

Authors	A. Luque (SOFTCRITS), F. Diaz(SOFTCRITS)
Verification	M. Díaz (SOFTCRITS)
Approval	Noemi Jimenez (CEMOSA)



Table of Contents

1	INTRODUCTION.....	7
1.1	PURPOSE.....	7
1.2	SCOPE	7
1.3	REFERENCE DOCUMENTS	7
2	ANALYSIS AND DESIGN	9
2.1	BUILDING INFORMATION FILE	9
2.2	PROPAGATION ALGORITHMS	9
2.3	LANGUAGES AND LIBRARIES	11
2.4	USER INTERFACE DESIGN	12
2.5	DEVELOPMENT MODEL	14
2.5.1	1st iteration	14
2.5.2	2nd iteration	14
2.5.3	3rd iteration.....	14
2.5.4	4th iteration.....	14
2.6	WISAN POSITIONING TOOL DESIGN AND IMPLEMENTATION.....	15
2.6.1	WISAN.Common.Controls.....	15
2.6.2	WISAN.Communications.Common	16
2.6.3	WISAN.Utils.....	17
2.6.4	WISAN.IfManager.....	17
2.6.5	WISAN.PositioningTool.....	18
2.6.5.1	Multi Wall Model.....	20
2.6.5.2	Dominant Path Model (DPM).....	21
2.6.6	Implementation notes.....	25
3	WISAN POSITIONING TOOL VALIDATION	27
3.1	VALIDATION METHODOLOGY	27
3.1.1	Coverage between two devices in line of sight	27
3.1.2	Coverage between two devices in obstructed line of sight	27
3.1.3	Coverage between two devices in no line of sight	28
3.1.4	Repeaters calculations.....	30
4	CONCLUSIONS	33
	ANNEX A - REFERENCES AND BIBLIOGRAPHY	35
	ANNEX B - WISAN POSITIONING TOOL. USER GUIDE	37
1.	CONFIGURATION.....	37
2.	READ IFC MODEL	39
3.	SETUP NODES AND GATEWAYS.....	40
4.	MATERIALS ATTENUATION.....	42
5.	PROCESS	43
6.	RESULTS	44



List of Figures

Figure 1.- a) Empirical Model. b) Ray Tracing Model. c) Dominant Path Model	10
Figure 2.- Position candidate for wireless nodes	10
Figure 3.- Application User Interface	13
Figure 4.- WISAN.PositioningTool components.....	15
Figure 5.- WISAN.Common.Controls packages diagram	15
Figure 6.- WISAN.Communication.Common class diagram	16
Figure 7.- WISAN.IfManager packages diagram	17
Figure 8.- WISAN.Positioning Tool class diagram	18
Figure 9.- Views package diagram	19
Figure 10.- SeedsComponents classes	19
Figure 11.- Classes that implement IPropagationModel.....	20
Figure 12.- Multi Wall Model.....	21
Figure 13.- a) Empirical Model. b) Ray Tracing Model. c) Dominant Path Model	21
Figure 14.- Room numbers (left), coupling walls (centre) and rooms-tree (right)	22
Figure 15.- Convex corners (or wedges) of a room and their influence on the determination of the paths	23
Figure 16.- Relations between convex corners of the room	23
Figure 17.- Devices in line of sight.....	27
Figure 18.- Devices in obstructed line of sight. Path avoids walls intersection	28
Figure 19.- Devices in obstructed line of sight. Path avoids column intersection.....	28
Figure 20.- Devices in no line of sight.....	29
Figure 21.- Devices in no line of sight. Path avoids walls intersection going through a door	29
Figure 22.- Devices in no line of sight. Path avoids stair intersection.....	30
Figure 23.- Candidate position for repeaters (☑).....	30
Figure 24.- Repeaters added in WISAN (☑).....	31
Figure 25.- Application User Interface	37
Figure 26.- Configuration Window.....	38
Figure 27.- Node Window	40
Figure 28.- Room Selector Window	41
Figure 29.- IFC Model with nodes and gateways	41
Figure 30.- Wisan Positioning Tool Database Importer	42
Figure 31.- Materials Attenuation Page	42
Figure 32.- Possible path between two nodes	43
Figure 33.- Possible path between a repeater and a node	43
Figure 34.- Results Page	44
Figure 35.- IFC File with nodes, gateways and repeaters	45

1 Introduction

1.1 Purpose

This document is a result from the project "Self learning Energy Efficient buildings and open Spaces" (SEEDS). It is created as part of the Working Package 3.5 "Positioning systems evaluation and Network Design Framework" and specifies how WISAN Positioning Tool has been developed and the results and conclusions that have been obtained from it. Participant involved in Work Package 3.5 is SOFTCRITS.

The main objective of Work Package 3.5 is the design, implementation and validation of a tool to configure a WISAN inside a building and to study the coverage between the different devices.

This document describes the software developed in the context of the WP 3.5. The nature of the deliverable is a prototype and in the following sections a description of the final application and its validation will be shown.

1.2 Scope

This document analyzes which standards and algorithms have been used during the implementation of this tool, the different iterations, the design of the application, its appearance. It also describes the validation of the tool and its user guide.

An overview of the structure of this document is given below:

- Chapter 2 relates the analysis and design of the application, starting by the standards adopted and finishing by an explanation in detail of the design of the application.
- Chapter 3 describes the validation of the WISAN Positioning Tool.
- Chapter 3 relates the conclusions reached after the development and validation of this new tool.
- Annex A enumerates the references used during the analysis, implementation, validation and documentation of the application.
- Annex B is the user guide of the application, and it describes how to use every single functionality of the tool.

1.3 Reference documents

The documents of importance for this deliverable are the following:

- The SEEDS Description of Work (DOW)
- Annex A of the Project Quality Plan
- SEEDS Deliverable 3.1 ("State of the technology in communications and sensor/actuator nodes for building automation")
- SEEDS Deliverable 3.2 ("Sensor/Actuator Positioning")
- SEEDS Deliverable 3.3 ("Hardware and software design of sensor/actuator nodes")

2 Analysis and Design

An overview of the structure of this section is given below:

- Section 2.1 explains which kind of file used to obtain the information of the building.
- Section 2.2 explains the propagation algorithms used to compute the propagation of the different devices and the communication between them.
- Section 2.3 describes briefly the user interface, i.e. how the user will use the application.
- Section 2.4 specifies the language that will be used to develop the application, as well as the libraries used.
- Section 2.5 describes the process by means of which the application has been implemented.
- Finally, section 2.6 describes the design and implementation of the application and all its components.

2.1 Building information file

In order to implement the application as standard as possible, a standard file type that keeps all the information about the building had to be chosen.

The standard chosen is the file .IFC (Industry Foundation Classes). The application reads the file, and it is able to modify it if it would be necessary to add the different nodes and gateways. IFC is the best option for this kind of application since it contains the kind of materials of each component.

Other type of CAD file to consider was .DWG, used by many CAD applications such as AutoCad®, but it does not contain any information about the material of each component so it was not possible to obtain the propagation properties of components.

2.2 Propagation algorithms

The main goal of the WISAN design tools is to help the designer in a previous stage to the network deployment, suggesting automatically the amount and the position of the wireless devices. The propagation model is an idealized model that predicts communication range between a set of devices.

Currently, the research lines of wireless sensor localization inside buildings indicate that in order to simulate the behaviour of the RF signal is important the use of propagation models. A propagation model is a set of mathematical expressions, diagrams and algorithms used to represent the characteristics of the RF signal propagation in a given environment. Propagation models can be classified into three approaches:

- **Empirical models**, as *MultiWall* [2], [3], [4] and [5], have the advantage of its high computational efficiency. However, the accuracy of their simulations depends on the accuracy and the similarity of the environment where they were taken and the new environment where it is to apply the model. In addition, the calculation is based on the transmitted signal directly, without considering mechanisms such as reflection, diffraction, multipath transmission ... In conclusion, the no inclusion of these mechanisms decreases simulations accuracy. (See Figure 1 a).

- **Deterministic models**, as *Standard Ray Tracing* [6], [7] and [8], are usually complex and computational inefficient, although they take in account indoor propagation mechanisms, such as reflection, diffraction... It leads to a prediction with more accuracy than empirical models. The main disadvantage of these methods is the high computational load needed in order to determine the prediction. (See Figure 1 b) and its dependency on the accuracy of the building structure data.
- **Hybrid models**, as *Dominant Path Model* [9] is concerned in the dominant paths between transmitter and receptor. This model sets the parameters that determine the dominant paths (path longitude, characteristics of the materials that are present in path...). Its computational load is less than the Ray Tracing, and is comparable with the Multi-Wall methods. Its accuracy is upper than statistical models, even some studies [10] considered more accurate than models based on ray tracing. (See Figure 1 c).

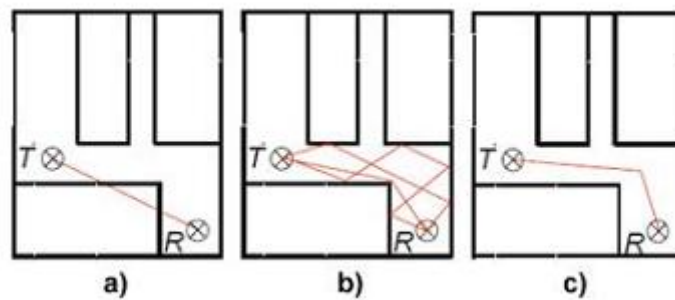


Figure 1.- a) Empirical Model. b) Ray Tracing Model. c) Dominant Path Model

Taking into account the advantages of *Dominant Path Model* (it is concerned just in dominant paths without calculate a huge amount of irrelevant paths, high accuracy and short computational time), the positioning tool to be develop in SEEDS project should at least implement this model.

But the point is that the complexity of *Dominant Path Model* algorithm is extremely high. For this reason, first of all, the *MultiWall* model will be implemented. Secondly, the *Standard Ray Tracing* will be developed, and finally, based on this last algorithm, *Dominant Path Model* will be implemented .

In order to determine the position candidate to the intermediate nodes, an algorithm based on a Growing Neural Gas network (GNG) [11] could be applied. This algorithm will allow distribute uniformly the candidate positions in a determinate environment, taking into account the ideally positions for the wireless devices. (See Figure 2)

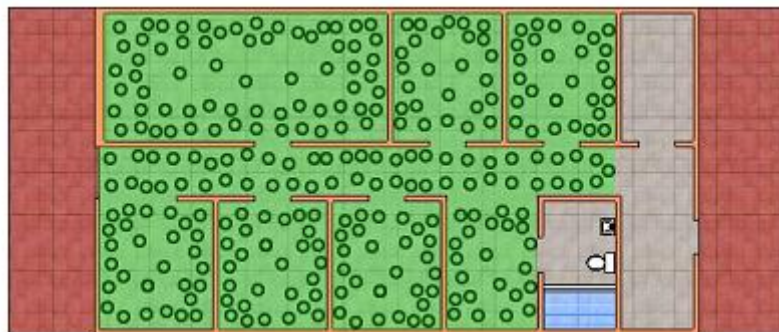


Figure 2.- Position candidate for wireless nodes

Lastly, an optimization mechanism will be needed in order to determine the minimum amount of devices to ensure the connectivity and coverage from the position candidate nodes. Research in this field is based on heuristics methods and metaheuristic algorithms of search [12], [13], [14] and [15].

2.3 Languages and Libraries

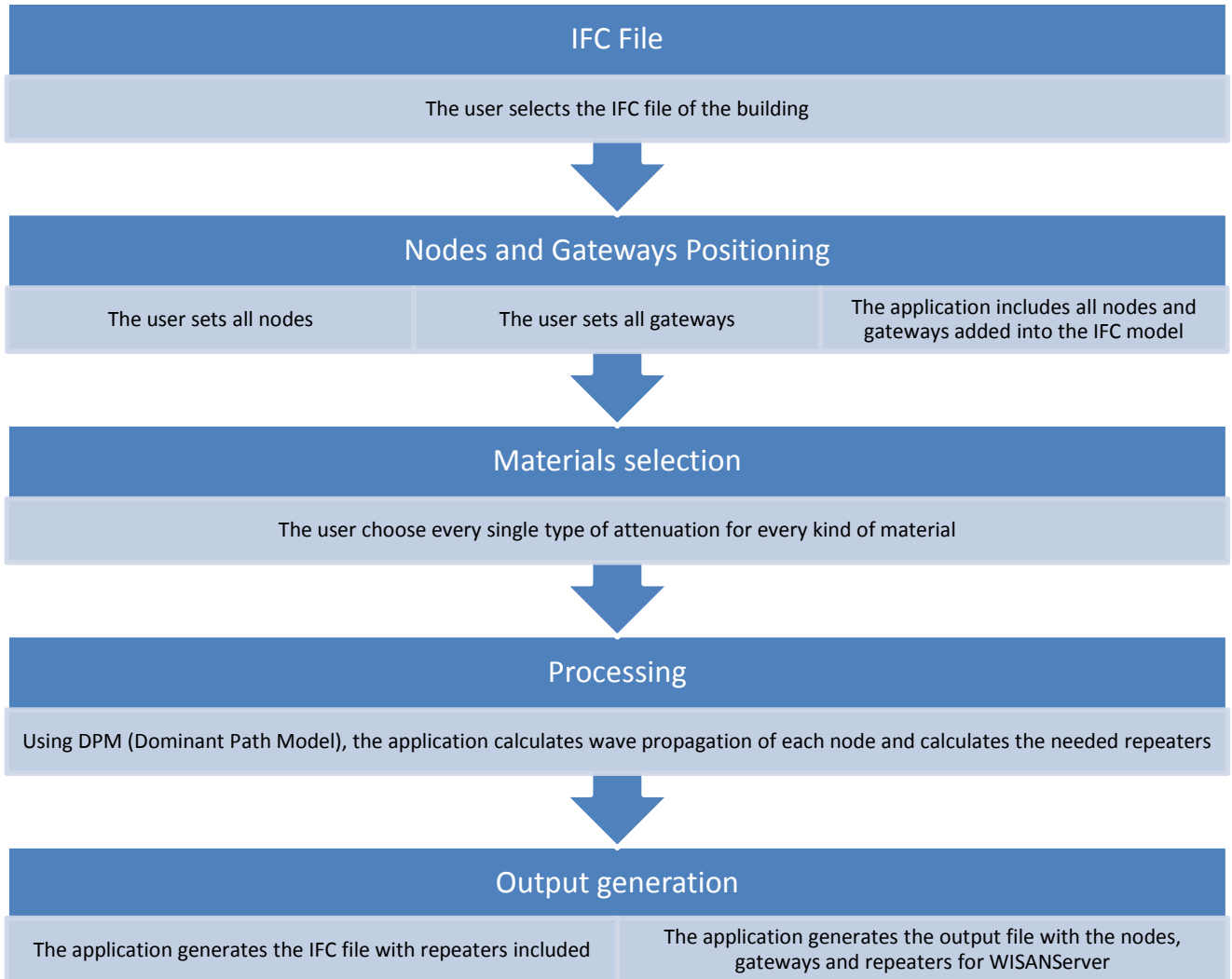
The language chosen to develop the application will be C#, and the GUI will be developed using WPF. The reason to choose this language is its high versatility and the great number of libraries available.

In order to read the different IFC Files, a new library will be implemented. This library will support different functions to read and write IFC files, and enough classes to manage all information needed about a building to process wave's propagation inside it.

With the aim of implementing DPM (Dominant Path Model), implementation of *MultiWall Model* will be done firstly, and once this algorithm works correctly, we will increase the accuracy of the model implementing the DPM.

2.4 User interface design

The process by which the user will use the tool will be the following:



The GUI will be developed with WPF and it will look like the following figure:

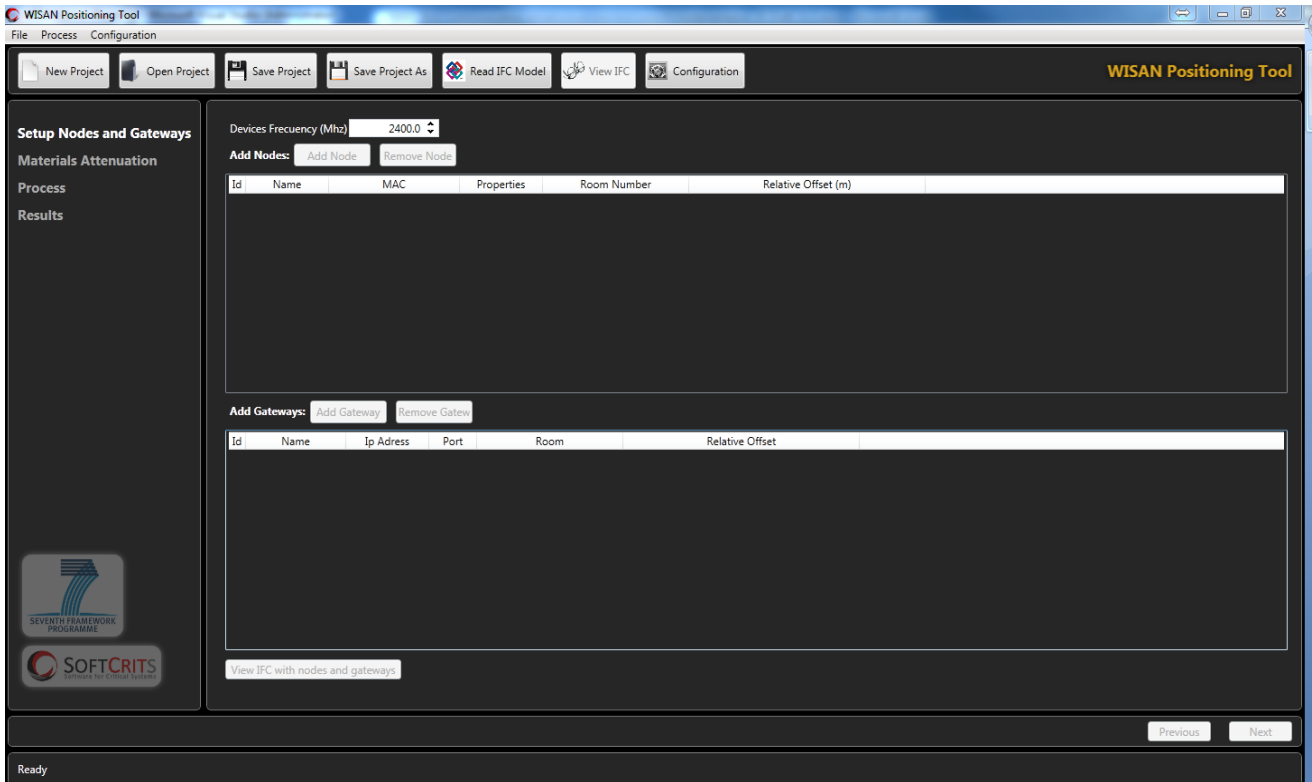


Figure 3.- Application User Interface

The user will follow the different steps in order to get the most efficient solution for the deployment of the WISAN.

2.5 *Development model*

An iterative and incremental development with 3 iterations is proposed. Each iteration will include the following functionality and those of their predecessors.

2.5.1 1st iteration

This version will include:

- IFC reading
- Nodes and gateways location
- Nodes and gateways location inside IFC file
- Configuration and parameters of the application
- *MultiWall* propagation model
- Calculation of repeaters needed
- Output file for *WISAN Server*
- Repeaters location inside IFC file
- WPF user interface
- IFC viewing opening an external IFC Viewer

2.5.2 2nd iteration

This version will include all features developed in 1st iteration and:

- More accurate nodes and gateways location inside the rooms
- Additional restrictions for repeaters location
- Configuration and parameters storage

2.5.3 3rd iteration

This version will include all features developed in 2nd iteration and:

- *Dominant Path Model* propagation model (Some adjust needed)

2.5.4 4th iteration

This version will include all features developed in 3rd iteration and:

- Check room limits for nodes and coordinators setup
- Wide range of Ifc files (no walls between rooms, strange rooms,...)
- More accurate calculation with walls, doors and windows
- Stairs intersection and stairs attenuation calculation
- Columns intersection and column attenuation calculation
- Add material attenuations to configuration and to model
- Add parameters to configuration
- Seeds database integration
- Calibration

2.6 WISAN Positioning Tool design and implementation

WISAN Positioning Tool uses five components to develop its functionalities. The following diagram shows the relation between these components:

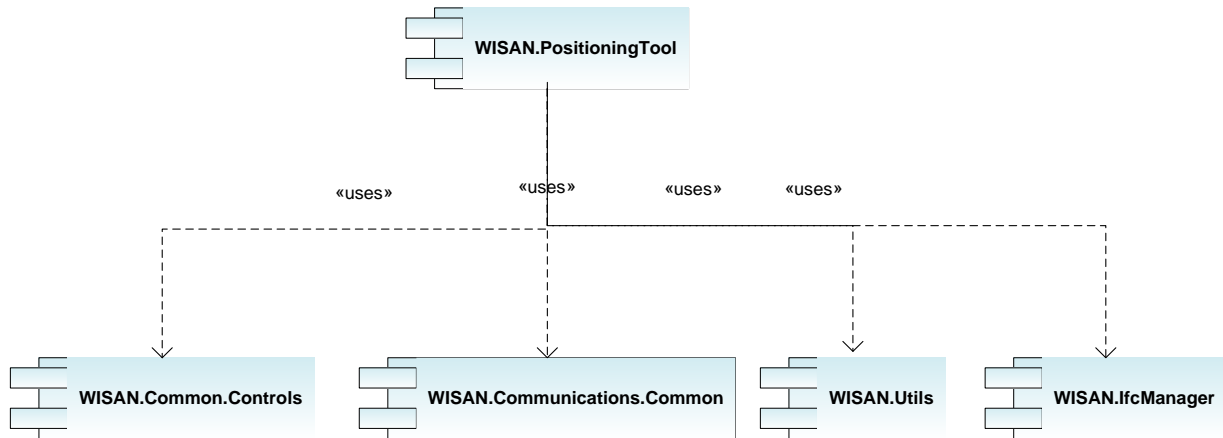


Figure 4.- WISAN.PositioningTool components

During the following sections, each component will be described in detail.

2.6.1 WISAN.Common.Controls

This component contains utilities, common elements and common windows to all the components of the application, for instance: the window for adding a sensor or the converter from sensor type to icon. Within the component, we found six packages shown in the following diagram:

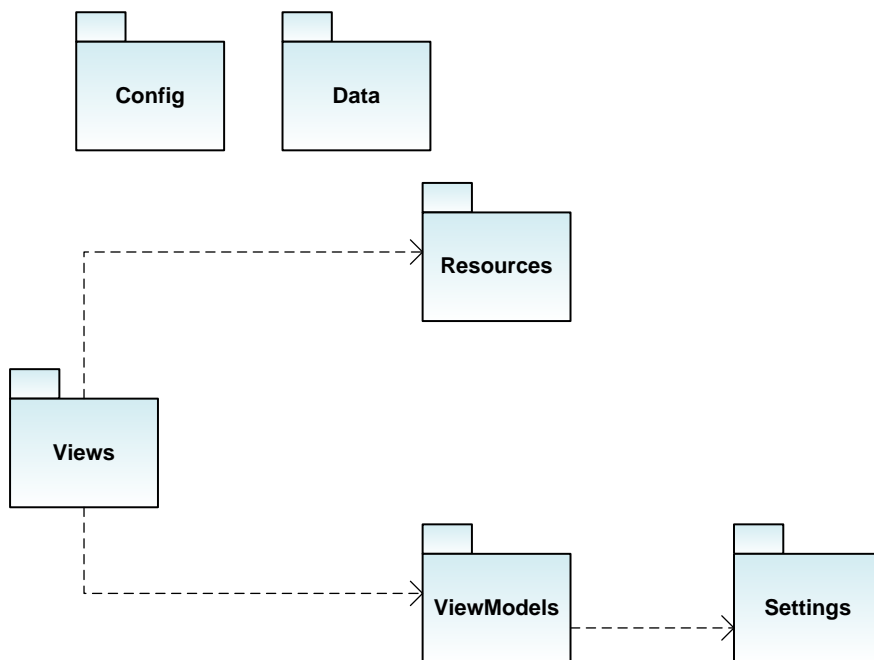


Figure 5.- WISAN.Common.Controls packages diagram

Within the component, we found six packages shown in the following diagram:

- **Config package:** Contains xml files that contain information about sensors and coordinators
- **Data package:** Contains the definition of the enumeration that controls the state of a component.
- **Resources package:** Contains the icons, the converters for WPF classes, the common resources dictionary that defines the common style for the different windows and the loading circle used when some process is running.
- **Settings package:** Contains the classes that control the settings of nodes, coordinators and server.
- **ViewModels package:** Contains a class with the business logic of the windows for each class of the view package.
- **Views package:** Contains a class for some windows of the application that are shared with other applications.

2.6.2 WISAN.Communications.Common

The WISAN.Communications.Common component defines the data types, messages and interfaces used by the rest of the components in the communications. Within the component, we found three packages and two interfaces shown in the following diagram:

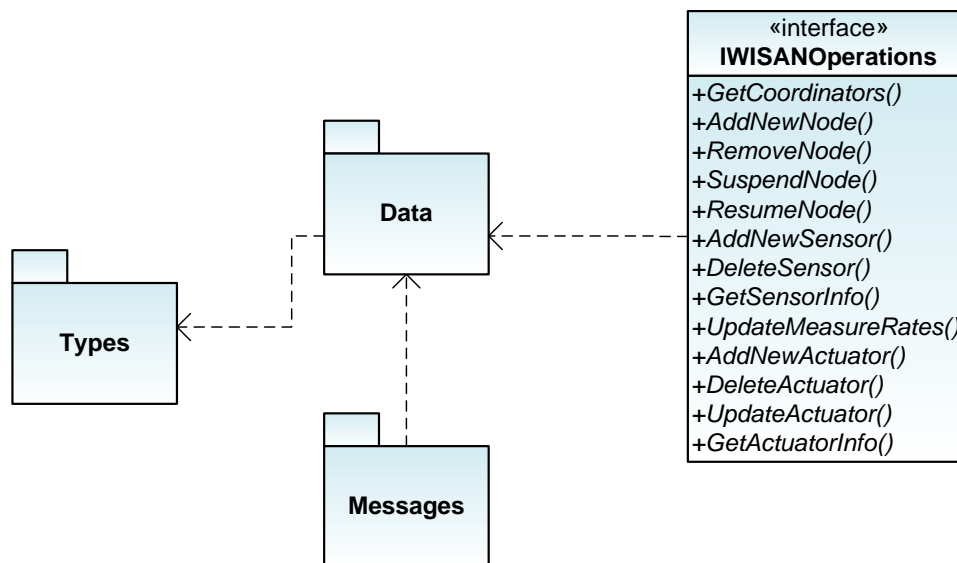


Figure 6.- WISAN.Communication.Common class diagram

- **Data package:** Contains the definition of the entities used in the communications such as nodes, actuators, sensors and coordinators.
- **Message package:** Contains the definition of the messages interchanged between the server and clients.
- **Types package:** Defines the types used in the communications and in messages.
- **IWISANOperations interface:** Defines the operations offered by the WISAN through the WISAN Communication server.

2.6.3 WISAN.Utils

This component contains utilities or common elements to all the components of the application, for example: the application log or the WISANDataUnit class that represents a measure of a sensor.

2.6.4 WISAN.IfManager

WISAN.IfManager defines all interfaces, classes... needed to manage IFC files whose version is IFC2x3. Inside of this component, there are functions that allow reading and writing IFC files. It also contains a window that shows the structure of an IFC file and some classes to analyse the information contained in the different classes of an IFC file. Within the component, we found eight packages shown in the following diagram:

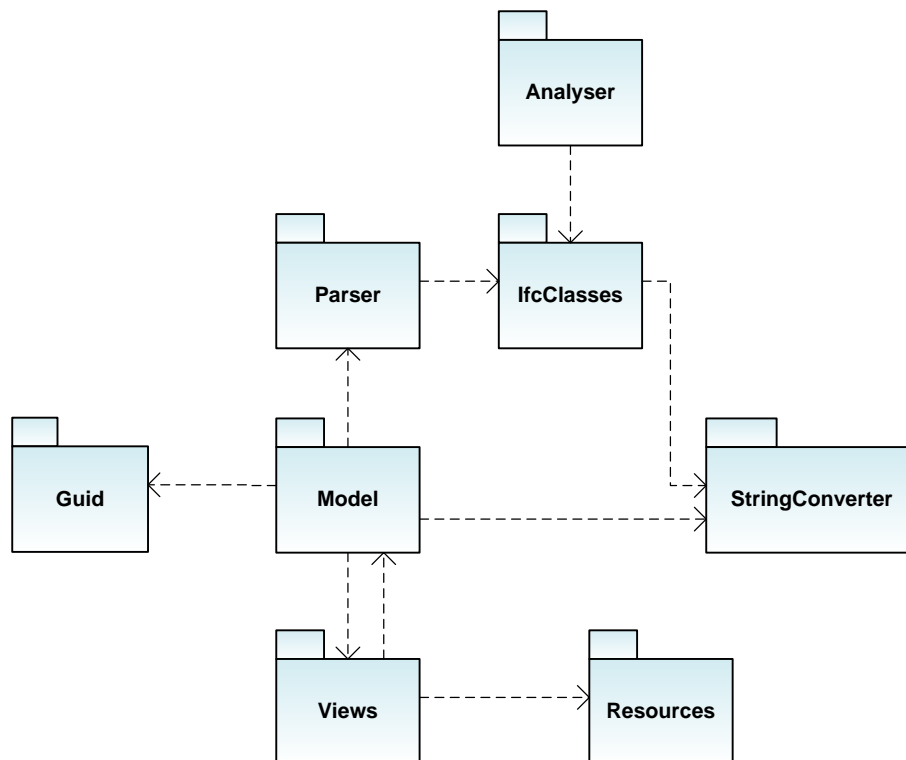


Figure 7.- WISAN.IfManager packages diagram

- **StringConverter package:** Contains the class that encode and decode string into different standards.
- **Resources package:** Contains the icons used in the views package.
- **Guid package:** Contains the classes that manage the GUID (Globally Unique Identifier) objects.
- **IfcClasses package:** Contains the most of the classes of the WISAN.IfManager. It defines a class for every single object defined by Ifc2x3 standard. It also contains some interfaces and enumerations defined by Ifc2x3 standard.
- **Model package:** This package defines the classes that manage IFC files.

- **Parser package:** Contains the definition of classes that parse IFC files, creating its correspondent IFC classes.
- **Analyser package:** This package contains some classes that calculate some information such as the exact position of an element in the space, using the information of all the classes of an IFC file.
- **Views package:** Contains some classes to represent IFC files data and the definition of the windows that show them.

2.6.5 WISAN.PositioningTool

WISAN.PositioningTool component is a .NET desktop application. This application is a GUI implemented using WPF and the MVVM (Model-View-ViewModel) design pattern. The following diagram shows the relation between the main packages of the component:

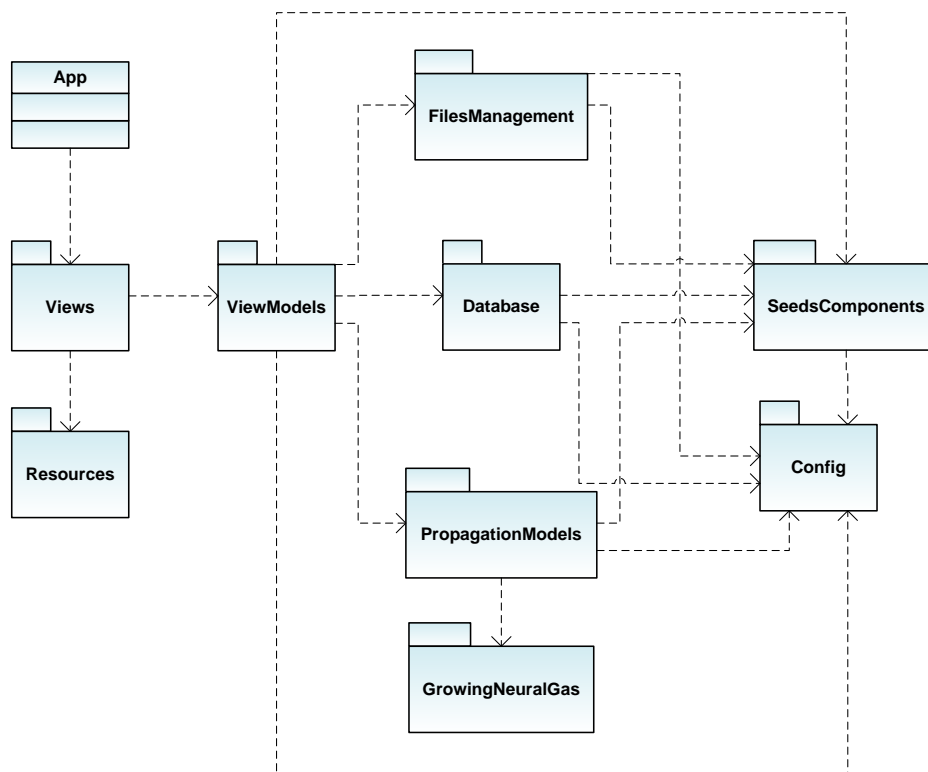


Figure 8.- WISAN.Positioning Tool class diagram

- **App:** The App class is the entry point of the program. This class initializes the MainWindow of the application.
- **Views package:** This package contains a class for each window of the application. MainWindow is the main window of the application and it contains a WizardMainControl. WizardMainControl is a user control that contains others user controls but only shows the one that is active. These user controls are SetupPage, MaterialsPage, ProcessPage and ResultsPage.

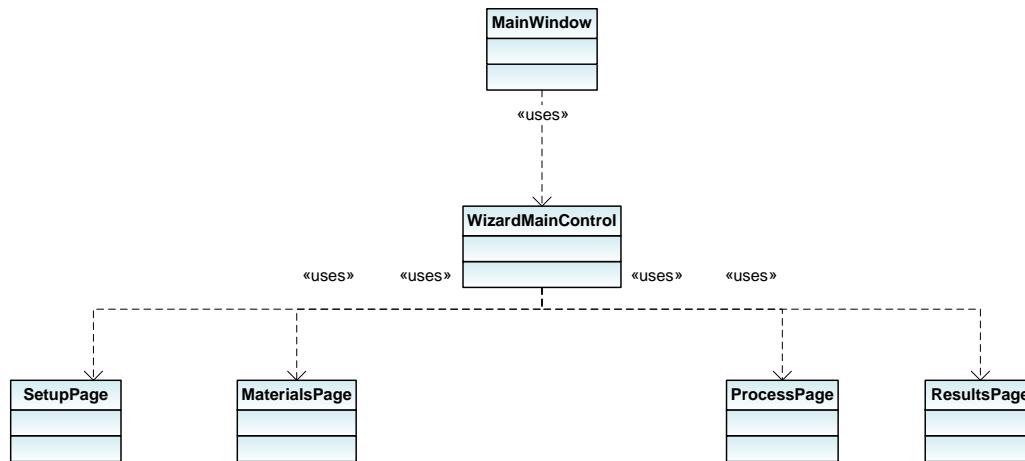


Figure 9.- Views package diagram

- **ViewsModels package:** This package contains a class with the business logic of the windows for each class of the view package.
- **Resources package:** This package contains the resources dictionary that defines the common style for the different windows of the application.
- **FilesManagment package:** This package manages different kind of files. It controls WISAN positioning tool project files, IFC files, ... It is responsible for reading and writing IFC models inside a project.
- **Database package:** This package contains the necessary classes to import the information of sensors, actuators, nodes and coordinators from a database. The database may be an Access database or a MySQL one. The class DataImporter is the class able to create both kinds of connections.
- **SeedsComponents package:** This package contains some classes that use classes SeedsNode and SeedsCoordinator and extend their functionality such as its location inside the building. It also contains the class WisanComponentOperations that adds the nodes, gateways and repeaters to the Ifc model.

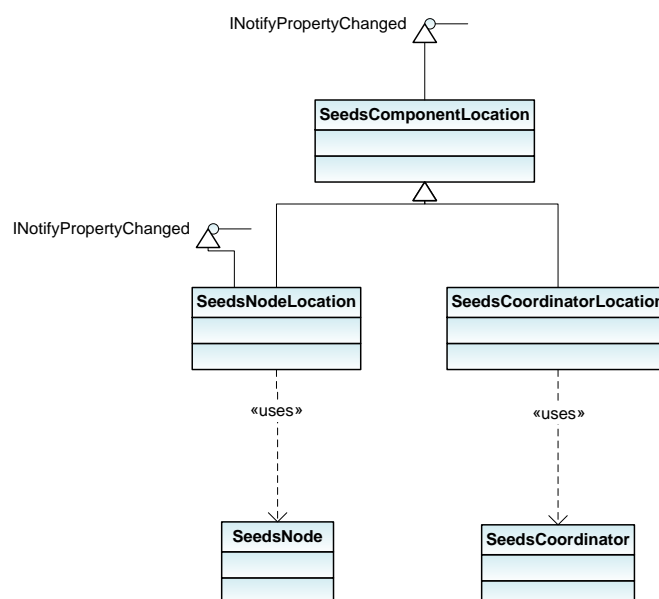


Figure 10.- SeedsComponents classes

- **Config package:** This package contains the constants and the settings of the application. Settings are saved in an xml file when are modified so they are reloaded when the application starts again.
- **GrowingNeuralGas package:** This package contains the necessary classes to create candidate positions for repeaters using a growing neural gas network. The main class of this package is ComputeGNG, and it is the one that make most of the calculations necessary.
- **PropagationModels package:** This is the package that implements all propagation calculations. It contains the interface IPropagationModel that is implemented by the class GeneralPropagationModel. Every propagation model must extend the class GeneralPropagationModel. At present, two different propagations models are implemented:
 - Multi Wall Model
 - Dominant Path Model

This package also contains the class WisanPropagationAnalyser that implements the algorithm that calculates the connections between the components of each storey, and calculates the power of the wave that every device receives from its correspondent transmitter.

A multi-objective algorithm to decide which of the candidate repeaters available is the most appropriate is also implemented in this package.

This package also implements the necessary classes to read from file the different materials attenuations, and calculate the attenuation for every material given, taking also into account thickness and frequency. Materials attenuation information is saved in 'Config\Materials.xml' file.

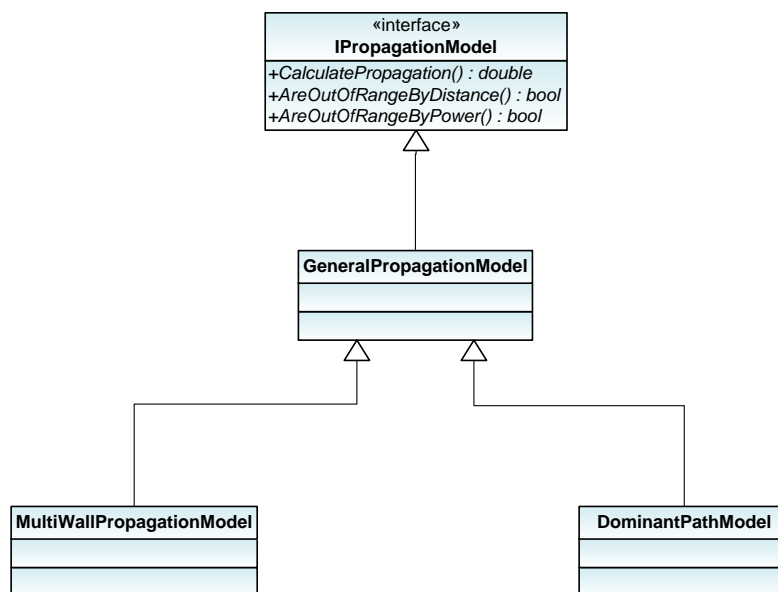


Figure 11.- Classes that implement IPropagationModel

In the following sections, both models are explained in detail.

2.6.5.1 Multi Wall Model

Multi Wall Model is an empirical propagation model that calculates the propagation of the wave tracing a ray between the transmitter and the receiver. It considers the distance between both devices and the objects that are in the path of the ray traced. It takes into account the different type of objects that the wave may cross, either walls, stairs, doors, windows or similar objects.

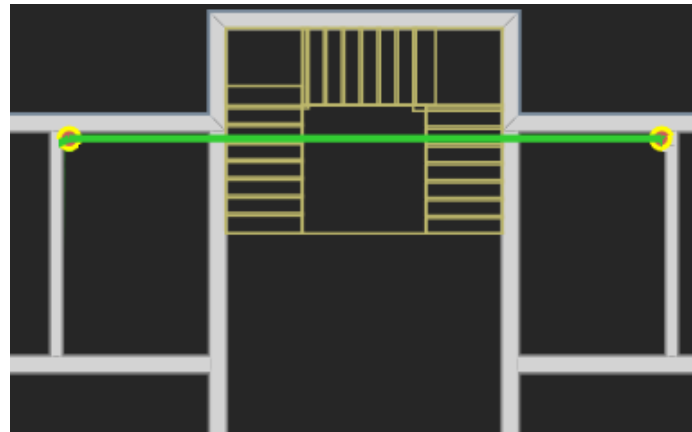


Figure 12.- Multi Wall Model

For each one of the objects that it intersects, it calculate the distance of intersection, and with it, the attenuations of the materials that compose the object and the frequency, this model calculates the losses produced by these objects. Subtracting materials attenuation (L_T) and distance attenuation(L_{FS}) to the power emitted by the transmitter(P_T), the algorithm calculate the power of the wave received by receiver:

$$P_R = P_T - L_{FS} - L_T$$

2.6.5.2 Dominant Path Model (DPM)

Empirical models (like Multi Wall Model) consider only the direct path between a transmitter and a receiver, and although they are very fast, its accuracy is very low. On the other hand, Ray Tracing models determine numerous paths and are very accurate but they are very time-consuming and dependable on the accuracy of building structure data. As shown in the figure on the right, Dominant Path Model determines only the most relevant path, reducing its considerably processing and still being very accurate.

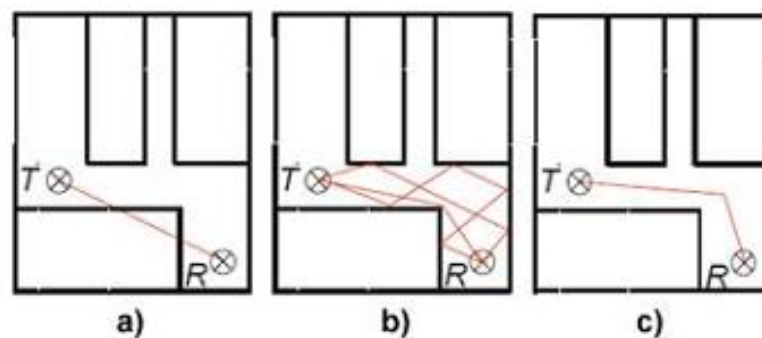


Figure 13.- a) Empirical Model. b) Ray Tracing Model. c) Dominant Path Model

In order to calculate the dominant path model between a transmitter and a receiver the model follows the next algorithm:

i. Rooms dictionary

First of all, this algorithm gets all the rooms from the IFC model that are located in the storey where both devices are situated. With these rooms' data, it may get all the information about the shape of the room and the bordering rooms.

ii. Tree of rooms relationship

Once all rooms of floor have been obtained, the information about the neighboring rooms is used to compute a tree with the structure of the rooms in the storey. In the following figure, reader may find a storey example, where rooms and walls are numerated. The root of the tree corresponds to the room in which the transmitter is located. The first layer contains all neighboring rooms of the root room and if there is more than one coupling wall between the room of the transmitter and the neighboring rooms, the neighboring room is placed in the first layer as many times as there are coupling walls between the two rooms. The others layers are determined by the same algorithm. All coupling walls to the neighboring rooms lead to a new branch in the tree, except if the wall is already used in this path in a layer above, in which case is not included.

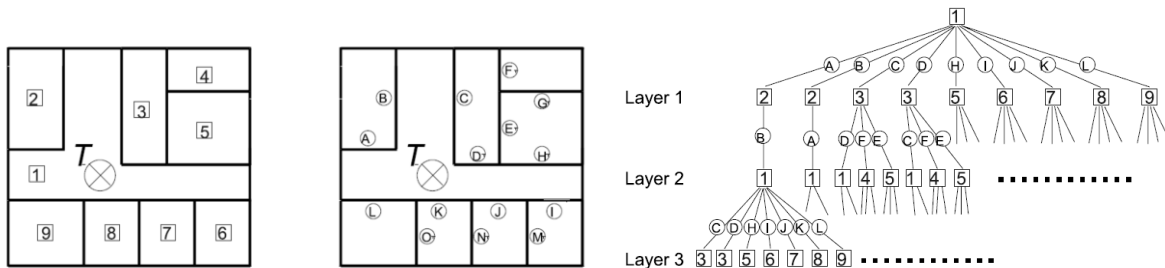


Figure 14.- Room numbers (left), coupling walls (centre) and rooms-tree (right)

Using IFC classes, the walls that are connected to a room are calculated using classes IfcRelSpaceBoundary that relate spaces and walls. Classes IfcRelSpaceBoundary are also used to divide some walls that form a single piece. For example, walls such as L, K, K and I usually forms a single wall, but this may be divided by spaces boundaries saved in the IFC file.

iii. Possible paths search

Once the determination of the tree has been processed, the dominant paths between the transmitter and the receiver can easily be calculated, because the tree represents in its branches all possible dominant paths and the sequence of rooms and walls passed. If the receiver is located in room_i, the tree must only be examined for room_i. The corresponding dominant path is determined by following all branches from room_i to the root of the tree, saving the sequence of rooms and walls passed.

iv. Rooms combination

After possible paths are calculated, room's combination is calculated for each path. If a path passes through a sequence of rooms, all rooms considered are combined to form a new single room by erasing all coupling walls in the branch representing the path in the room tree. If transmitter and receiver are located in the same room, no processing needs to be done. The combination of different rooms to a new room is shown in the following figure, for transmitters located in room 1 and receivers located in rooms 6 and 2 respectively.

v. Determination of the path inside room combination

Now the path inside the new room can be computed by using the following algorithm.

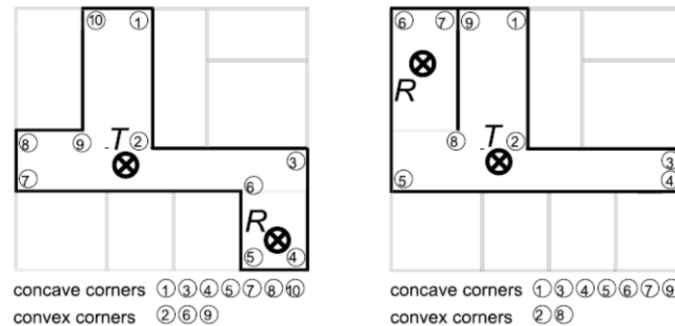


Figure 15.- Convex corners (or wedges) of a room and their influence on the determination of the paths

The dominant paths inside this room are determined with the information about the convex corners as shown in the previous figure. There are two different scenarios for the determination of the path between the transmitter and the receiver: Line of sight and obstructed line of sight. The first case is trivial because the dominant path is the direct ray between transmitter and receiver. In obstructed line of sight the dominant path must lead via convex corners to the receiver. For the determination of the path a tree with the convex corners must be computed. In order to generate the tree of convex corners, an analysis of convex corners of the combined room must be done.

If inside the combination of rooms there is some object such as stairs, columns ... the convex corners that have these objects must also be added to the list of corners in order to be computed and permit to avoid these objects.

After that, to generate the tree, the algorithm must check the visibility between the corners, omitting the removed walls. All corners visible from the examined corner are new branches in the tree. As shown in figure 7, the tree of corners starts with the corners visible from the transmitter. The receiver is also included in the tree. Each time the receiver is found in the tree, the corners along the path can be determined by following the branches back to the transmitter.

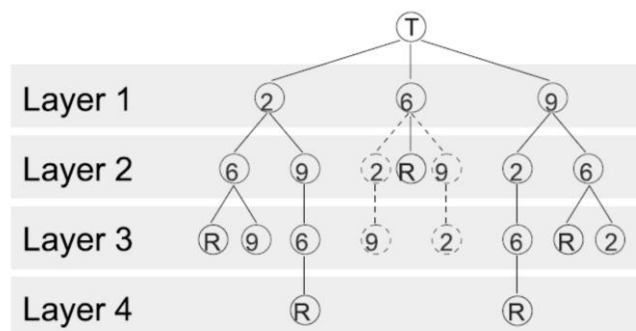


Figure 16.- Relations between convex corners of the room

The last step is to move some distance the path from the corners in order to make the bend slighter.

vi. Selection of the minimum-loss dominant path (MLDP)

The algorithm for the determination of the dominant paths leads to more than one solution. But in most cases only one solution with the smallest path loss is necessary for an accurate

prediction. This most important path is called minimum-loss dominant path (MLDP). The MLDP is chosen by utilizing the total loss of the path:

$$L = W_F \cdot L_{FS} + W_T \cdot L_T + W_I \cdot L_I$$

L represents the empirical determination of the total attenuation along the path and consists of three different parts:

- **Free space attenuation L_{FS}**

The free space loss L_{FS} depends on path *length* and *frequency*:

$$\begin{aligned} \frac{L_{FS}}{dB} = & 10 \cdot PropagationFactor \cdot \log_{10} \left(\frac{Frequency}{MHz} \right) \\ & + 10 \cdot PropagationFactor \cdot \log_{10} \left(\frac{Distance}{m} \right) \\ & - (30 \cdot PropagationFactor + 32.44) dB \end{aligned}$$

where *PropagationFactor* is the factor of attenuation in a determined space. Its value is usually 2.0 but it may vary slightly.

- **Transmission loss L_T**

The accumulated transmission loss of all walls passed is represented by L_T . If the path intersects n objects with their individual transmission losses L_i , the total transmission loss L_T is computed from

$$L_T = \sum_{i=1}^{N_T} L_i$$

- **Interaction loss L_I**

Changes in the direction of the path as given by angle α_i at point *i* represent an additional loss. All angles α_i are accumulated and normalized with a factor α_L to give the interaction loss L_I .

$$L_I = \frac{1}{\alpha_L} \cdot \sum_{i=1}^{N_I} \alpha_i$$

Best results are gained with $\alpha_L = [5^\circ, 8^\circ]$.

Different paths are determined for each receiver point and the path with the smallest total attenuation L is chosen and called minimum-loss dominant path (MLDP). It is possible to adjust the results of dominant paths by adjusting the three weighting factors W_{FS} , W_T and W_I .

Finally, the power of the wave received by the receiver is calculated by the following formula:

$$P_R = P_T - L_{FS} - L_T - L_I$$

2.6.6 Implementation notes

This tool is a *.NET* application that has been developed under *Microsoft Visual Studio 2010* using *C#* language. WPF Technology has been used for the design of the windows, which have been implemented using the *MVVM* (Model-View-ViewModel) design pattern.

In order to facilitate the calculation of intersections between objects and rays, *Microsoft XNA* tools have been used. And the graphic representation of the components of the buildings and nodes, gateways and repeaters has been done using *Canvas*.

3 WISAN Positioning Tool validation

3.1 Validation methodology

In order to validate WISAN Positioning Tool, a validation process is proposed. This validation consists on the deployment of a WISAN in a real building using this tool, and comparing the results of signal power calculated by the application with the real ones. Results considered are those calculated with the Dominant Path Model, which offers more accurate results than *MultiWall Model*.

In order to make the validation as precise as possible, we will divide the validation in four main parts:

- Coverage between two devices in line of sight
- Coverage between two devices in obstructed line of sight
- Coverage between two devices in no line of sight
- Repeaters calculation

For each part, different cases may be taken into account. In the following sections, each one of these cases will be explained.

3.1.1 Coverage between two devices in line of sight

This is the simplest case, when two elements are in line of sight, not having any element between them. Only distance attenuation is taken into account.

In this case, differences between the simulation and real measure should be very small.

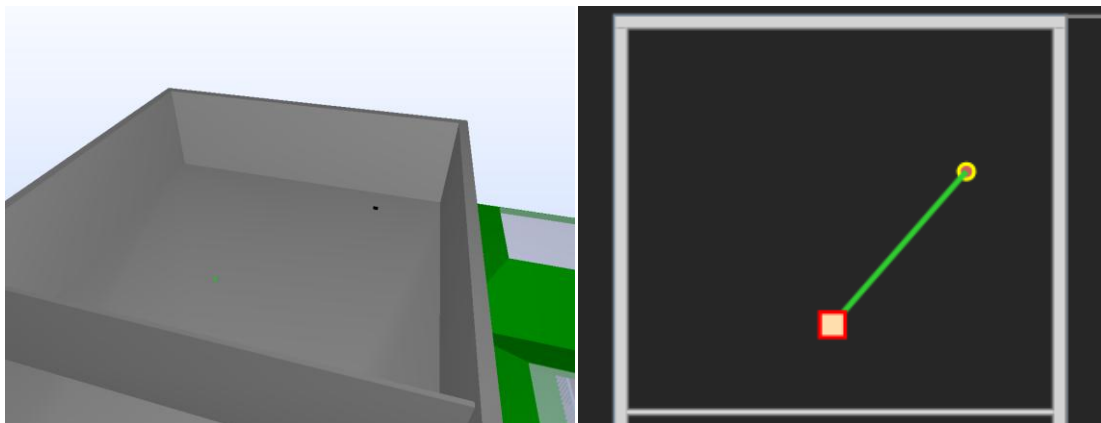


Figure 17.- Devices in line of sight

3.1.2 Coverage between two devices in obstructed line of sight

In this case, the devices are not in line of sight, but there is a path that joins both points and does not need to cross any object, for example, going through a corridor. In some of these cases, wave power is higher than crossing a wall or another kind of object, so the dominant path on these cases is similar to the following figure.

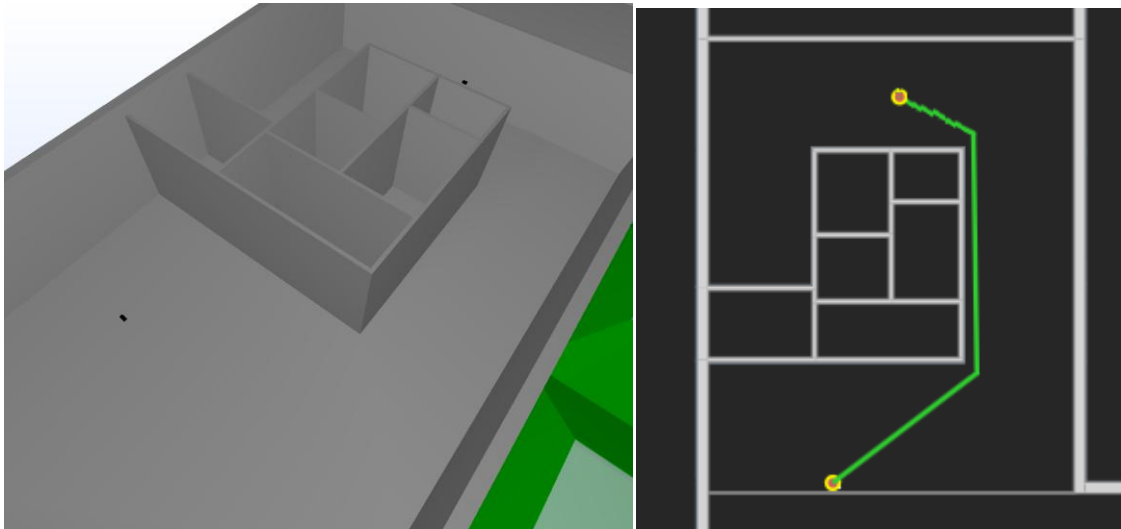


Figure 18.- Devices in obstructed line of sight. Path avoids walls intersection

For this case, the model analyses many different options and choose the solution with the lowest loss. To analyse the different options, it creates a map of connections between rooms and decide which walls, doors or windows should cross. If the lowest loss is acquired without crossing any object, simply travelling through the corners, we obtain this case. This case is possible because of the effect of reflection, diffraction and scattering.

This case not only considers walls but also stairs, columns, windows, doors or similar objects. In the following figure, reader may find how the algorithm has calculated that avoiding transmission through the column obtains the path of the wave with minimum loss.

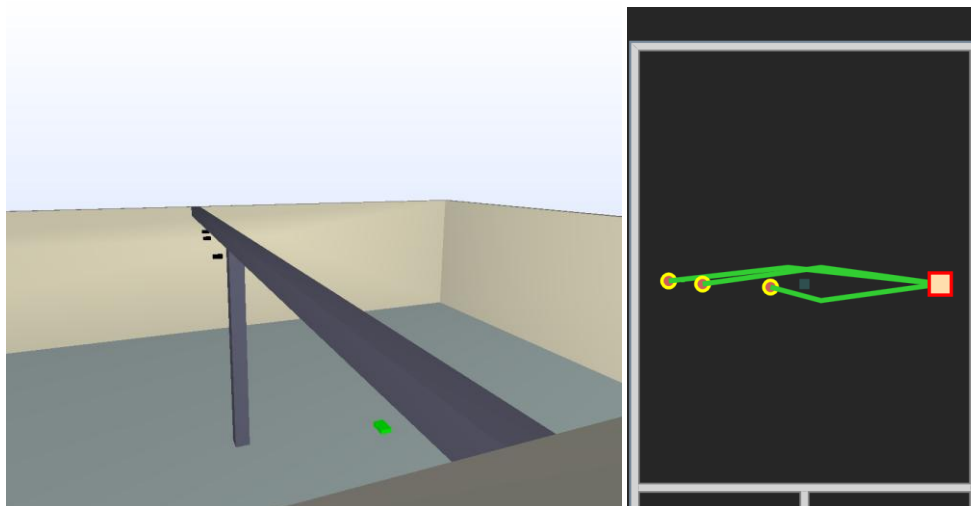


Figure 19.- Devices in obstructed line of sight. Path avoids column intersection

3.1.3 Coverage between two devices in no line of sight

This last case is the more complex one, because of its complexity and diversity. It happens in most cases, when the two devices are in no line of sight because there are one or more objects

between the line that joins both gadgets. Objects considered between the devices include walls, stairs, columns, doors, windows and similar objects.

For this case, the model analyses many different options and choose the solution with the lowest loss. To analyse the different options, it creates a map of connections between rooms and decide which walls, doors or windows should cross. Usually, model tries to pass between rooms through doors or windows because usually, these objects have attenuations factors lower than walls. Although this fact not always happens since attenuations factors may be similar or distance to cross from one room to another through a door usually is higher than crossing the wall directly. It depends on every single situation.

The simplest casuistry of this case is the following one. The wave dominant path crosses the wall and reaches the receiver.

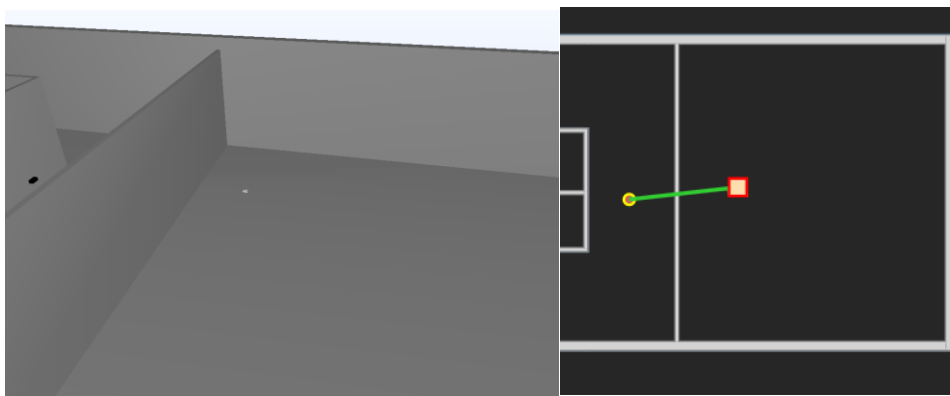


Figure 20.- Devices in no line of sight

But usually, cases are more complex. For example, in the following situation, the path avoids crossing many walls and simply crosses a wall and a door travelling though the corridor. By this way, it reduces the losses that would be produced by crossing 5 walls. This case is possible because of the effect of reflection, diffraction and scattering.

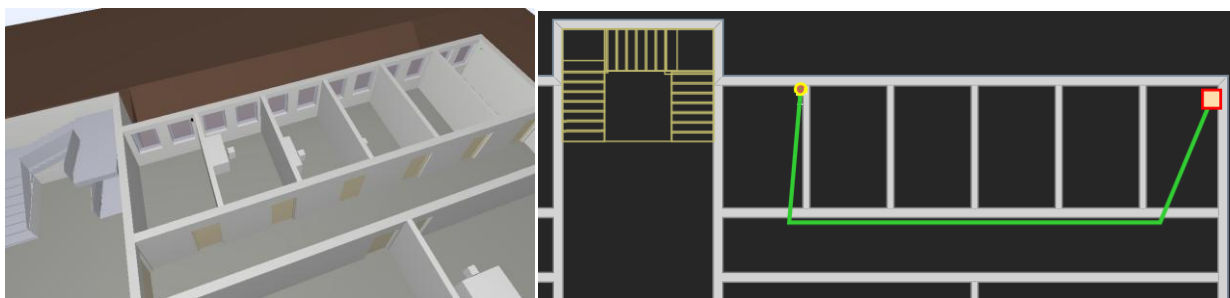


Figure 21.- Devices in no line of sight. Path avoids walls intersection going through a door

In the following case, the path avoids crossing a stair (that has a high attenuation factor because of its thickness and material) and simply crosses two walls. By this way, it reduces the high loss that would be produced by crossing the stair. This case is possible because of the effect of reflection, diffraction and scattering.

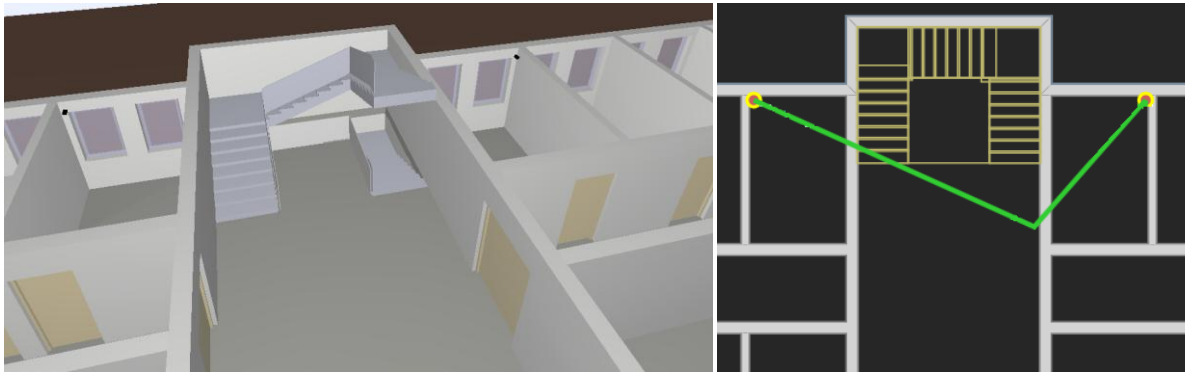


Figure 22.- Devices in no line of sight. Path avoids stair intersection

3.1.4 Repeaters calculations

When there are some devices incommunicado, repeaters are needed. In order to calculate the minimum amount of repeaters that guarantee communication between all gadgets in WISAN, an algorithm that calculates the number and position of repeaters that are needed has been included in the tool. This algorithm creates candidate positions for repeaters using a *Growing Neural Gas Network* and then calculates repeaters needed using a multi-objective algorithm. In the following figure reader may find all positions generated by the *Growing Neural Gas Network*.

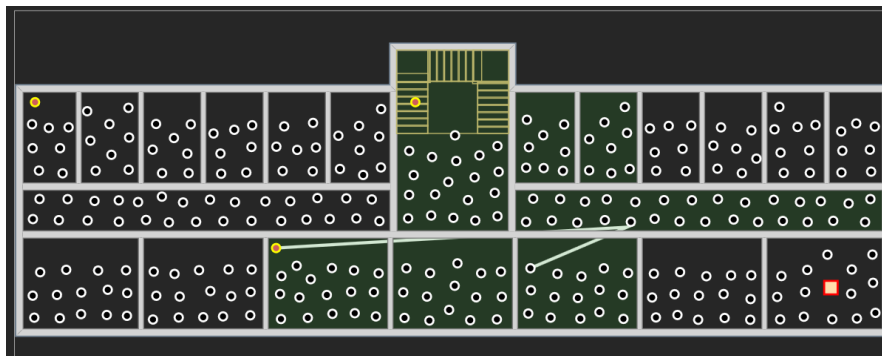


Figure 23.- Candidate position for repeaters (□)

The multi-objective algorithm must guarantee communication between every single node with at least one gateway, without adding more repeaters than necessary. For this reason, during the validation, we must check that every single node is communicated after adding all repeaters proposed. We also must check that if we remove one of these repeaters, at least one of the nodes cannot receive data from any coordinator or the communication between them is not stable.

In the following figure, reader may see where 2 repeaters have been added to WISAN. The repeater in the middle of the storey, join 3 devices with the coordinator. With only one repeater, the coordinator now connects with two nodes and it is closer to the node that it is still incommunicado.

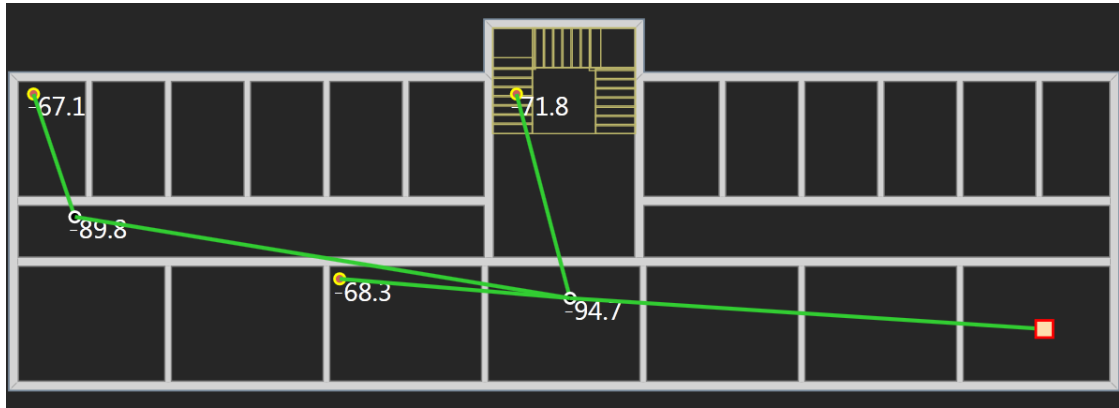


Figure 24.- Repeaters added in WISAN (□)

4 Conclusions

Deliverable 3.5 ("*WISAN Deployment Framework*") describes the analysis, design and implementation of an application that assists engineers to deploy wireless intelligent sensor and actuator network (*WISAN*) in a Building Automation Control Systems. With this application, people in charge of *WISAN* deployment will be able to study the coverage of the different nodes across the building, improving the distribution of the nodes throughout the building, and calculating the necessary repeaters to guarantee network communication.

The application described in deliverable 3.5 implements an *IFC* reader/writer in order to get all necessary data of the building. *IFC* is a standard that describes building and construction data. Besides, it implements an extensible library of materials attenuations to calculate the attenuations through the different materials with a wide range of frequencies.

In addition, this new tool implements the *Dominant Path Model* in order to simulate the propagation of the waves through the building model. *Dominant Path Model* offers much more accurate results than *MultiWall Model*, also implemented in this application, and it is not excessively time-consuming.

To guarantee the communication between nodes, the deployment of necessary repeaters is also implemented. This implementation includes the generation of multiple candidate positions using a *Growing Neural Gas Network* and study of the coverage of the different candidate repeaters by means of a multi-objective algorithm. In order to validate this tool, a validation methodology has been proposed.

With all these features, this new application ease the deployment of a wireless sensor and actuator network making easier to improve its coverage.

ANNEX A - References and Bibliography

1. **K. Akkaya, M. Younis.** *A survey on routing protocols for wireless sensor networks.* s.l. : Elsevier Journal of Ad Hoc Networks 3 (3) 325–349., 2005.
2. **Chang, J.J., Hsiu, P.C., Tei-Wei:** *Search-oriented deployment strategies for wireless sensor networks.* In: *Object and Component-Oriented Real-Time Distributed Computing*, 2007. ISORC '07. 10th IEEE International Symposium on. (2007) 164 - 171.
3. **Pinto, A., D Angelo, M., Fischione, C., Scholte, E., Sangiovanni-Vincentelli, A.:** *Synthesis of embedded networks for building automation and control.* In: *American Control Conference*, 2008. (2008) 920 - 925
4. **Kohtamaki, T., Pohjola, M., Brand, J., Eriksson, L.:** *Piccsim toolchain - design, simulation and automatic implementation of wireless networked control systems.* In: *Networking, Sensing and Control*, 2009. ICNSC'09. International Conference on. (2009) 49 - 54
5. **Ma, Y.S., Song, J., Kwak, J.Y., Yu, M., Woo, D.K., Mah, P.:** *Realssim: a simulator for indoor sensor network systems.* In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems.* SenSys '10, New York, NY, USA, ACM (2010) 369{370
6. **Guinard, A., McGibney, A., Pesch, D.:** *A wireless sensor network design tool to support building energy management.* In: *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-E ciency in Buildings.* BuildSys '09, New York, NY, USA, ACM (2009) 25 - 30
7. **Mc Gibney, A., Klepal, M., ODonnell, J.T.:** *Design of underlying network infrastructure of smart buildings.* In: *Intelligent Environments*, 2008 IET 4th International Conference on. (2008) 1 - 4
8. **Ma, Y.S., Song, J., Kwak, J.Y., Yu, M., Woo, D.K., Mah, P.:** *Realssim: a simulator for indoor sensor network systems.* In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems.* SenSys '10, New York, NY, USA, ACM (2010) 369 - 370
9. *Indoor Dominant Path Prediction Model.* <http://www.awe-communications.com/Propagation/Indoor/DPM/index.htm>
10. *Indoor Prediction Models.* <http://www.awe-communications.com/Propagation/Indoor/Comparison/index.htm>
11. **Fritzke, B.:** *A growing neural gas network learns topologies.* In: *Advances in Neural Information Processing Systems 7*, MIT Press (1995) 625 - 632
12. **Zhang, H., Hou, J.:** *Maintaining sensing coverage and connectivity in large sensor networks.* NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks - (2004)
13. **Dhillon, S., Chakrabarty, K., Iyengar, S.:** *Sensor placement for grid coverage under imprecise detections.* In: *Information Fusion*, 2002. Proceedings of the Fifth International Conference on. Volume 2. (2002) 1581 - 1587 vol.2
14. **Chang, J.J., Hsiu, P.C., Tei-Wei:** *Search-oriented deployment strategies for wireless sensor networks.* In: *Object and Component-Oriented Real-Time Distributed Computing*, 2007. ISORC '07. 10th IEEE International Symposium on. (2007) 164 - 171

15. **Huang, Y.K., Hsiu, P.C., Chu, W.N., Hung, K.C., Pang, A.C., Kuo, T.W., Di, M., Fang, H.W.:** *An integrated deployment tool for zigbee-based wireless sensor networks*. In: Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on. Volume 1. (2008) 309 - 315
18. *Material Database*. http://www.awe-communications.com/Download/DemoData/Databases_Material.zip
19. **G. Wölfle, F. M. Landstorfer:** *Dominant Paths for the Field Strength Prediction*. Institut für Hochfrequenztechnik, Universität Stuttgart
20. **G. Wölfle, F. M. Landstorfer, R. Gahleitner, E. Bonek:** *Extensions to the Field Strength Prediction Technique Based on Dominant Paths Between Transmitter and Receiver in Indoor Wireless Communications*. Institut für Hochfrequenztechnik, Universität Stuttgart
21. **G. Wölfle, F. M. Landstorfer:** *Field Strength Prediction in Indoor Environments with Neural Networks*. Institut für Hochfrequenztechnik, Universität Stuttgart
22. **G. Wölfle, F. M. Landstorfer:** *Field Strength Prediction with Dominant Paths and Neural Networks for Indoor Mobile Communication*. Institut für Hochfrequenztechnik, Universität Stuttgart
23. **Mohamed Bechir Khrouf, Mohamed Ayadi, Sodki Ben Romdhane, Najib Saghrouni, Sami Tabbane, Ziad Belhadj:** *Indoor Prediction of Propagation Using Dominant Path: Study and Calibration*. Unite de recherche en Reseaux Radio mobiles Multimedia (MEDIATRON); Ecole Superieure des Communications de Tunis (Tunisia)

ANNEX B - WISAN Positioning Tool. User guide

WISAN Positioning Tool consists on an application that helps to deploy the different nodes and gateways in a building, analyzing their coverage and setting repeaters where necessary. The main window of the tool looks like:

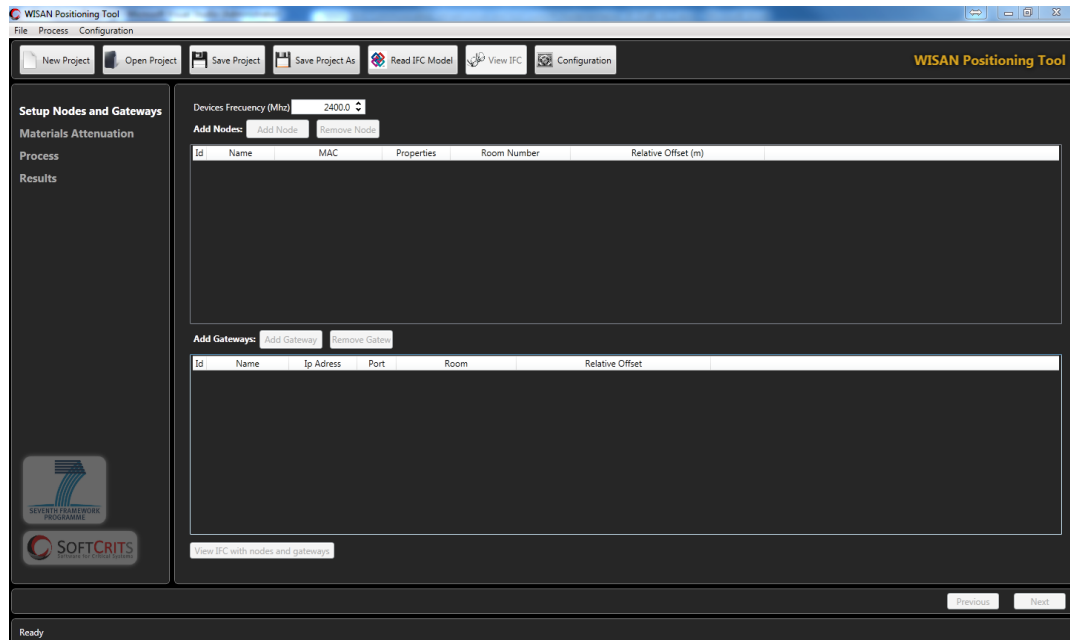


Figure 25.- Application User Interface

In order to analyze the coverage of the wireless components inside a building, user must follow the next steps:

1. Configuration

First of all, user should configure the most important parameters of the application. These parameters may be set clicking on the button 'Configuration'. From this window, user may set the following parameters:

- **Transmit Power (dBm):** The power of the wave transmitted by every wireless component. The higher the power, the farther the distance will reach the waves.
- **Receiver Sensitivity (dBm):** Measured in dBm, it is the minimum power that a wireless component may receive and process. If a component receives a signal with a lower power than this parameter, the component will not be able to receive any data from this signal.
- **Max Hops of Packages:** A data package may be sent from one device to another through other devices if they are not able to communicate between them directly. That package may pass a maximum quantity of devices in the middle, defined by this parameter.
- **Default Component Elevation (m):** Default elevation inside a floor where the nodes and gateways are set when they are added to WISAN. Besides, this will be the elevation of the repeaters added if necessary.
- **Propagation Model:** There are two different models implemented in this application:

- Multi Wall: This is the simplest model. It calculates the propagation of the wave tracing a ray between the transmitter and the receiver. It considers the distance between both devices and the objects that are in the path of the ray traced.
- Dominant Path Model(DPM): This is the recommendable model. This model is a hybrid model that considers all rooms and walls in the storey and traces a path between the different rooms and walls that minimize the losses. Then it calculate the power of the wave received considering the distance of the path, the interaction of every change in direction and the losses produced by the transmission through objects that cannot be avoided.
- **Processing Priority:** User may choose between a quick solution and a more accurate one. The more accurate, the longer it will take. A trade-off between performance and accuracy is recommendable. If the solution does not take very long, user may select a more accurate configuration.

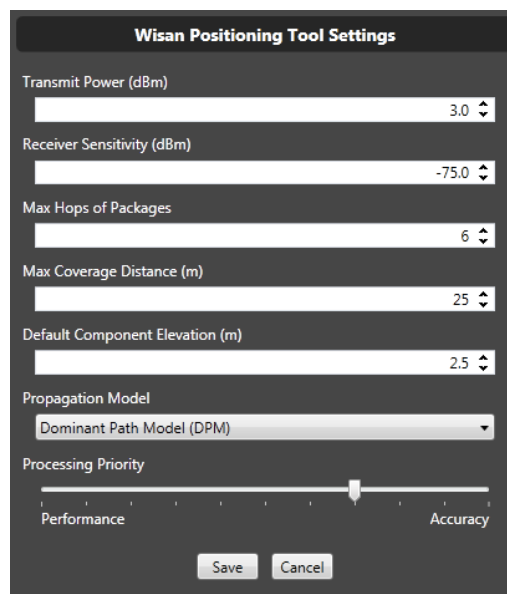


Figure 26.- Configuration Window

There are others advanced parameters that are editable from an .xml file, but they just must be configured by an expert of the tool. The file 'Config\ PositioningToolSettings.xml' has the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<WisanPositioningToolSettings>
  <Settings ReceiverSensitivity="-102" TransmitPower="3" MaxLatency="500" MaxPackageHops="6"
  MaxCoverageDistance="25" Frecuency="868" DefaultComponentElevation="2.5" PropagationModel="DominantPathModel"
  ProcessingPriority="5">
    <DetailedLog>false</DetailedLog>
    <StandardSeparationOfComponentsFromWall>0.15</StandardSeparationOfComponentsFromWall>
    <LineOfSightPropagationFactor>2.1</LineOfSightPropagationFactor>
    <NoLineOfSightPropagationFactor>2.5</NoLineOfSightPropagationFactor>
    <DistanceToMoveFromConvexCornerToPathCross>0.55</DistanceToMoveFromConvexCornerToPathCross>
    <LineOfSightPropagationFactorBeforeBreakpoint>2</LineOfSightPropagationFactorBeforeBreakpoint>
    <ObstructedLineOfSightPropagationFactorBeforeBreakpoint>2.1
      </ObstructedLineOfSightPropagationFactorBeforeBreakpoint>
    <NoLineOfSightPropagationFactorBeforeBreakpoint>2.2</NoLineOfSightPropagationFactorBeforeBreakpoint>
    <LineOfSightPropagationFactorAfterBreakpoint>4</LineOfSightPropagationFactorAfterBreakpoint>
    <ObstructedLineOfSightPropagationFactorAfterBreakpoint>4
      </ObstructedLineOfSightPropagationFactorAfterBreakpoint>
    <NoLineOfSightPropagationFactorAfterBreakpoint>4</NoLineOfSightPropagationFactorAfterBreakpoint>
    <AngleLoss>6</AngleLoss>
    <FreeSpaceAttenuationWeight>1</FreeSpaceAttenuationWeight>
    <TransmissionLossWeight>1</TransmissionLossWeight>
    <InteractionLossWeight>1</InteractionLossWeight>
    <MaximumInteractionLoss>13</MaximumInteractionLoss>
  </Settings>
</WisanPositioningToolSettings>
```

```
<BreakpointDistance>500</BreakpointDistance>  
</Settings>  
</WisanningPositioningToolSettings>
```

The meaning of these parameters is the following:

- **DetailedLog:** If true, the log that appears in the process page will be more detailed.
- **StandardSeparationOfComponentsFromWall:** Standard separation in meters from a wall to every new device (node or gateway) added to WISAN. The exact position may be changed for every single device through setup page.
- **LineOfSightPropagationFactor, NoLineOfSightPropagationFactor:** Propagations factors for multiwall model.
- **DistanceToMoveFromConvexCornerToPathCross:** Separation in meters from a convex corner to the path by which the wave is calculated.
- **LineOfSightPropagationFactorBeforeBreakpoint, ObstructedLineOfSightPropagationFactorBeforeBreakpoint, NoLineOfSightPropagationFactorBeforeBreakpoint:** Propagations factors for DPM when the distance is lower than BreakpointDistance.
- **LineOfSightPropagationFactorAfterBreakpoint, ObstructedLineOfSightPropagationFactorAfterBreakpoint, NoLineOfSightPropagationFactorAfterBreakpoint:** Propagations factors for DPM when the distance is higher than BreakpointDistance.
- **AngleLoss:** Factor of normalization for interaction loss. Best results are gained with values between 5 to 8
- **FreeSpaceAttenuationWeight:** Weight of free space attenuation. Standard value is 1. Set it higher than 1 to increase the effect of free space attenuation, or lower to decrease it.
- **TransmissionLossWeight:** Weight of transmission loss. Standard value is 1. Set it higher than 1 to increase the effect of transmission loss, or lower to decrease it.
- **InteractionLossWeight:** Weight of free interaction loss. Standard value is 1. Set it higher than 1 to increase the effect of interaction loss, or lower to decrease it.
- **MaximumInteractionLoss:** Maximum loss produced by interactions through the path.
- **BreakpointDistance:** The breakpoint distance describes the distance from the transmitter when destructive superposition of the direct ray and the ground reflected ray occurs. Because of that the attenuation is much higher and thus larger path loss exponents are often used. The default distance for the breakpoint distance in indoor environments is 500m.

2. Read IFC Model

Once the configuration is set, user must read the IFC Model of the building. It must be an IFC file of the version IFC2x3. Older or newer versions may not work properly.

When the IFC file has been read, user may save the project using buttons 'Save Project' or 'Save Project'. The project will include the IFC file and a project file with the configuration of all nodes and gateways added. The project must be loaded in any moment using the button 'Open Project'.

3. Setup Nodes and Gateways

After previous step, user may begin adding nodes and gateways. First of all, user must introduce the 'Devices frequency' in MHz. The frequency of the most common standard protocols used inside buildings are:

- ZigBee (IEEE 802.15): 868 MHz (Europe), 915 MHz (USA and Australia) and 2400 MHz in most jurisdictions worldwide.
- Wi-Fi (IEEE 802.11): 2400MHz and 5000MHz.
- Bluetooth: 2400MHz.

Once that frequency is introduced, user should start adding nodes and gateways. To do this task, simply press buttons 'Add Node' and 'Add Gateway'. For nodes, a window will appear. With this window, sensors and actuators may be added to a node. Of course, both nodes and coordinators may be deleted selecting them and pressing the correspondent 'Delete' button.

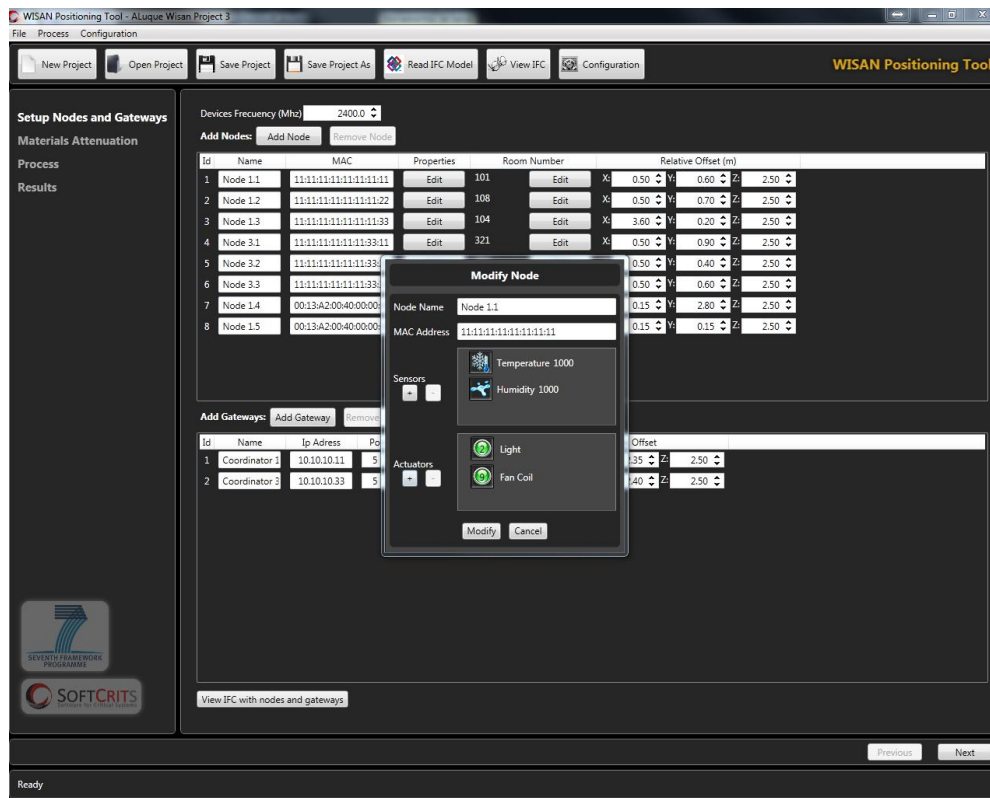


Figure 27.- Node Window

For both, nodes and gateways, user has to select the room where they are placed. For this purpose, press the button 'Edit' that is situated in the column 'Room Number'. A window with all building structure will appear. Select a valid room to set the node, and then, set it in the specific required position using the relative offset parameters of the component. If user selects a position out of the bounds of the room, an error message must appear.

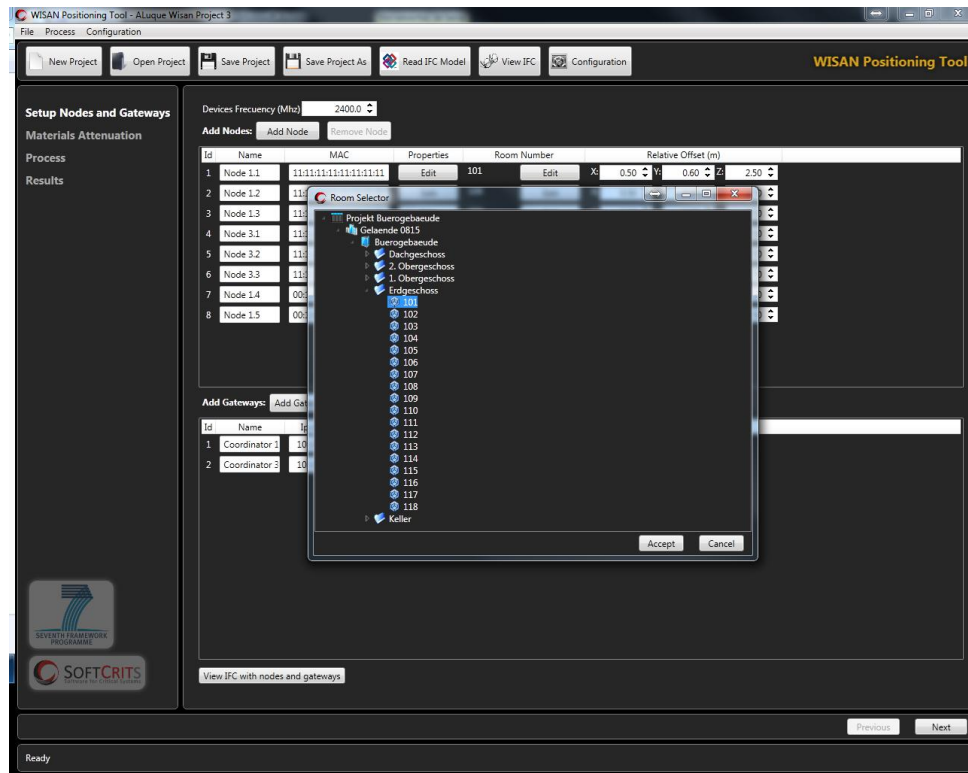


Figure 28.- Room Selector Window

User may use the button 'View IFC with nodes and gateways' to check that the devices are set in the desired position. In order to open the IFC file, an IFC Viewer is needed. If there is no IFC Viewer installed in the computer, user may find many free version on internet.

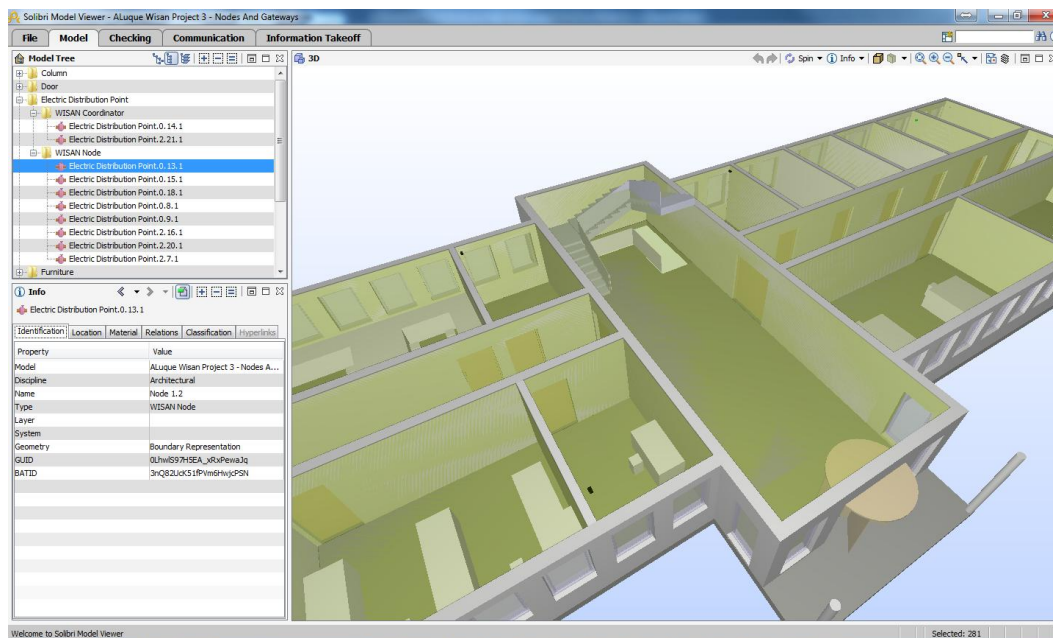


Figure 29.- IFC Model with nodes and gateways

User may notice that he can import components from SEEDS database, available in Access and MySQL. This function is available in the menu **File --> Import Components from SEEDS Database**.

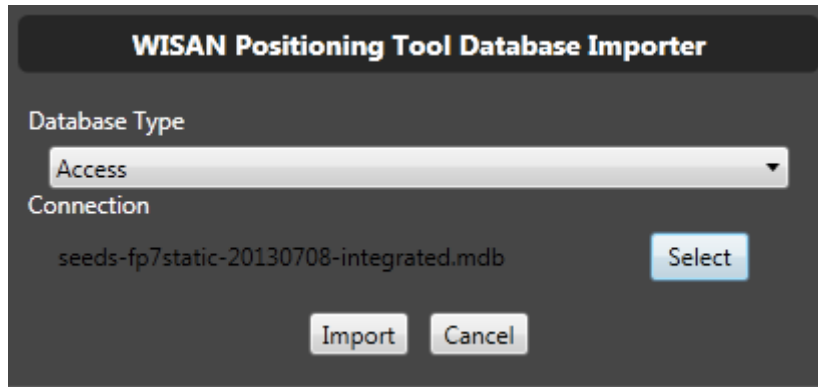


Figure 30.- Wisan Positioning Tool Database Importer

After selecting the database type and the connection, importation will begin. When it is finished, the tool asks user if he would like to view the xml output file that WISAN Server needs to be configured. User should realise if some of the parameters was not imported correctly, since some parameters like location are not always imported correctly because may not exist in the IFC file.

Once this first step is complete, user can pass to the next step, 'Materials Attenuations'.

4. Materials Attenuation

In this page, the application shows all materials that exist in the IFC model selected, and assign directly attenuation depending on the frequency, thickness and material. The tool assign the most similar material available in its materials library, but user may select another kind of material if he considers that the material chosen by application is not the proper one. User may also introduce a specific attenuation if he knows it selecting 'User Defined' in 'Material Type' list.

Materials library is read from the file 'Config\Materials.xml' and must not be modified by anyone but an expert in materials attenuations.

It is recommendable to save the project again at this point since this process may be bothersome.

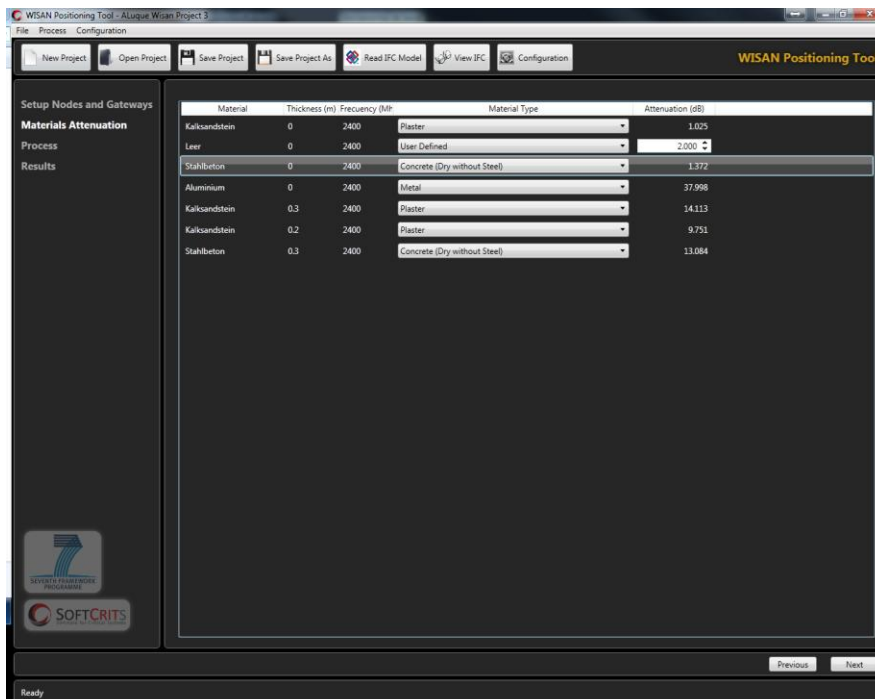


Figure 31.- Materials Attenuation Page

5. Process

After selecting materials attenuation, user can now press 'Start Coverage Processing'. For each storey, the analysis of the devices will be done and displayed by the log and graphic sections.

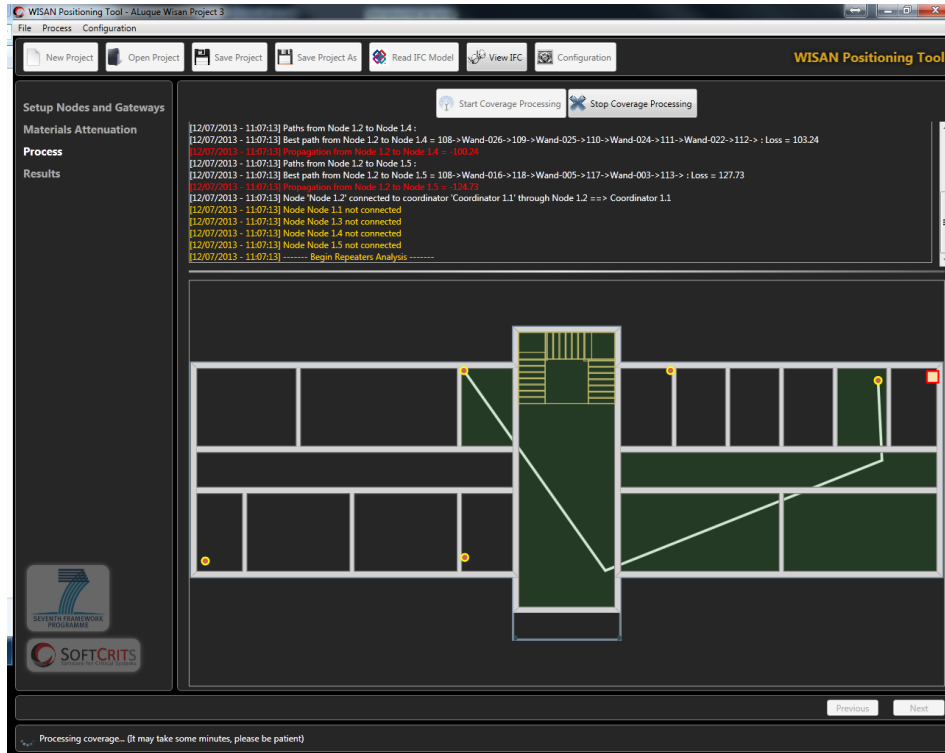


Figure 32.- Possible path between two nodes

In the graphic section are shown the gateways (■) and the nodes (●). If some repeaters are needed, the application generates a mesh of candidate repeaters using a Growing Neural Gas Network. Repeaters are represented by ⊙. In the previous and following figures, the different paths that the tool creates before to choose the dominant path are displayed.

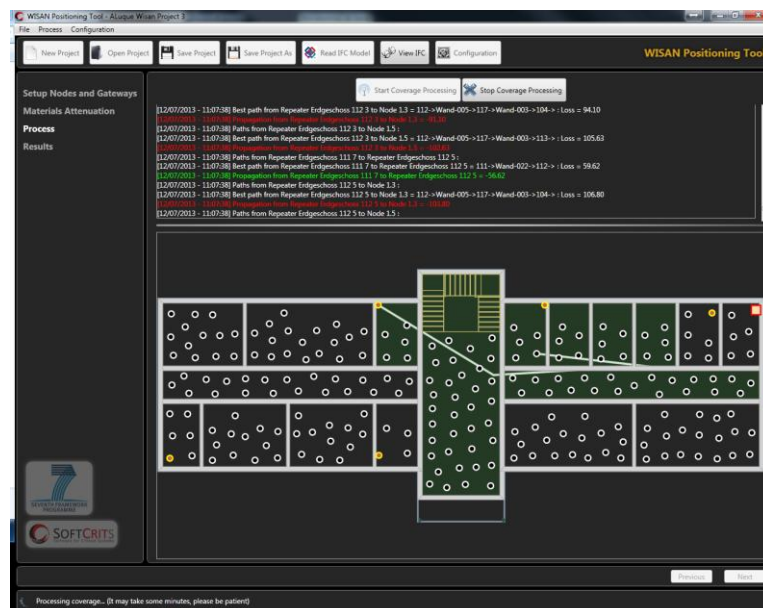


Figure 33.- Possible path between a repeater and a node

Processing may be stopped in any moment by user, for example, to reduce the accuracy of the calculation if it takes too long. Once the process has finished, user may pass to the results page.

6. Results

In this page, repeaters (if needed) in WISAN are shown, in its exact room and position. Besides, wave power in every link is displayed. User can change the name of every repeater and its MAC, in order to create the Xml Output File that WISAN Server needs to be configured. User may view this file pressing the button 'View Xml Output File' or in the path 'Config\Coordinators.xml'.

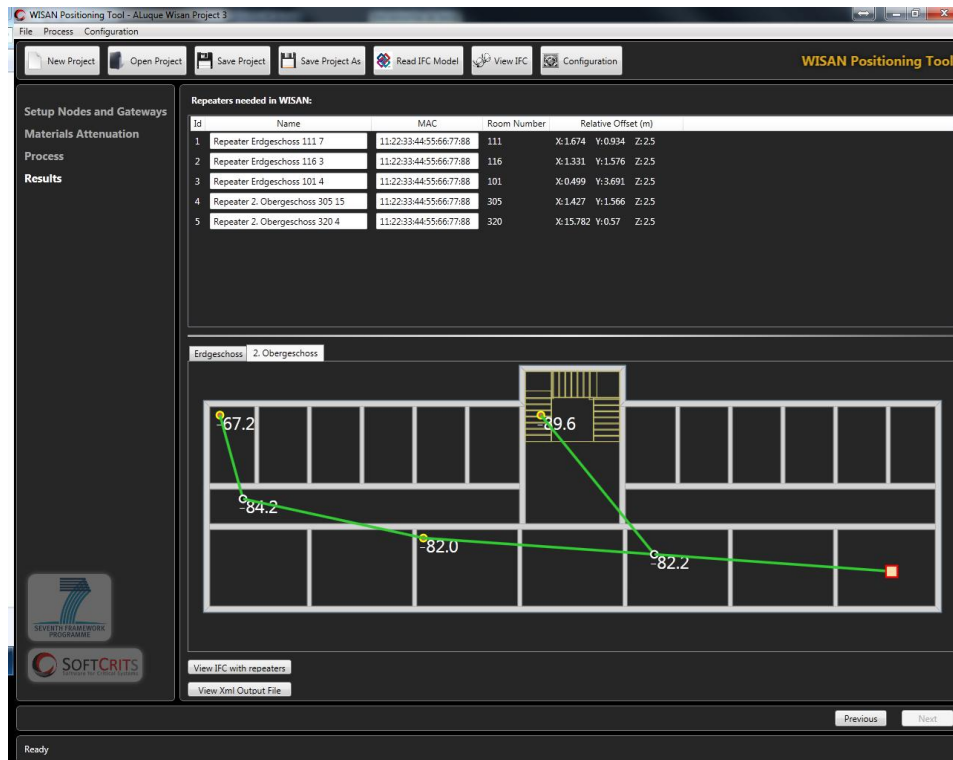


Figure 34.- Results Page

It is also possible to view the final IFC file with all nodes, gateways and repeaters included in the building structure by pressing the button 'View IFC with repeaters'.

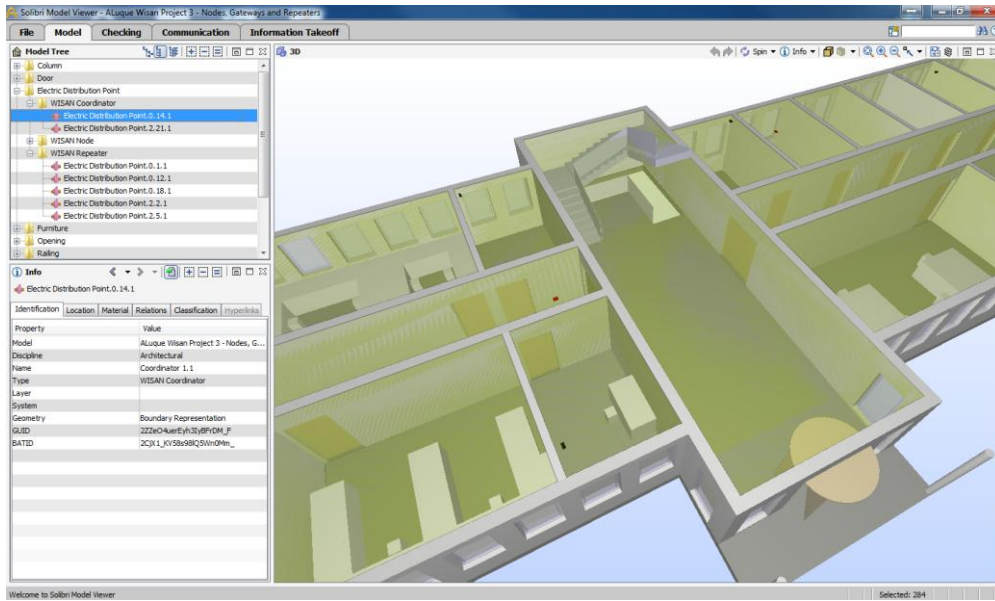


Figure 35.- IFC File with nodes, gateways and repeaters