INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

A 3D Interactive Environment for Automated Building Control

Dissertation for the achievement of the degree

Master in Information Systems and Computer Engineering

Jury

Chairman:	Prof. Doutor José Manuel Nunes Salvador Tribolet
Advisor:	Prof. Doutor Paulo Jorge Fernandes Carreira
Co-Advisor:	Prof. Doutor Alfredo Manuel dos Santos Ferreira Júnior
Members:	Prof. Doutor Renato Jorge Caleira Nunes

November 2012

Acknowledgements

First of all, I would like to thank my advisors Professor Paulo Carreira and Alfredo Ferreira, who gave excellent ideas, resources and connected me to the right persons, making this work more interesting and important than what I had imagined. Thank you for your wisdom and patience on writing this thesis.

A special thank you to my friend Pedro Cruz, who helped me with my english grammar issues making me laugh a lot with its funny comments regarding my text. Thank you for your patience with my repetitive and probably annoying habit of using "where" instead of "were" and "trough" instead of "through".

To all my friends, who gave me lots of support, feedback and help me through this work. Without you this thesis would not be possible, and for that I am extremely grateful. Thank you for your friendship. Academic life without you guys would not have been the same! I hope I made you all proud.

Last but not least, i would like to thank, with all my heart to my parents, who supported me through all my life and gave up on a lot of things to get me where I am. For my grandparents, who were just as instrumental in my education. This is the result of all my hard years of study. You are incredible support and loving family throughout. This thesis is dedicated to you.

Abstract

Building Automations Systems (BAS) coordinates and controls electric and mechanical devices installed in buildings to improve occupant comfort and safety, while at the same time reducing energy consumption. An important aspect for the effective usage of these systems is efficient handling the representation of the location of devices along with their corresponding status. However, the state of the art, in terms of graphical interaction regarding these types of user interfaces, is still quite primitive by comparison with modern interactive applications.

This thesis explores the use of a 3D interactive environment for maintenance activities by augmenting a tridimensional virtual facility with information regarding the status of systems and the space itself while providing simple and intuitive ways to monitor and control them. To do so we implemented a BAS prototype regarding a satisfactory interface for monitoring and control in the building automation domain, making use of game engine technologies. By satisfactory we mean an interface that adequately conveys the perception of the status of the building automation systems. Our prototype makes use of 3D building representations to ameliorate the perception of space along with the representations of each device, its corresponding status and spatial information.

To evaluate our ideas, we report a comparative study that contrasts our interface applied to the centralized control of a building automation system with a corresponding legacy application, verifying the reliability and possible benefits of 3D interactive environments on BAS. Conclusions of the evaluation of our prototype, indicate that these kind of interfaces have the potential to significantly increase the productivity in maintenance tasks. We think that the potential drop in user time and increase in engagement with a 3D interface will eventually translate into lower cost and to an increase in quality, potentially turning 3D based interfaces the option of choice in future IT tools for building automation systems.

Keywords: Interfaces, 3D Interactive Environments, Building Automation, Game Engines

Contents

List of Tables

List of Figures

Acronyms

API Application Programming Interface

BAC Building Automation Control

BAS Building Automation System

BIM Building Information Modeling

CAD Computer-aided Design

CAN Controller Area network

CAFM Computer-aided Facility Management

CCMS Centralized Control and Monitoring Software

CMS Control and Monitoring Software

DDL Direct Digital Control

FM Facility Management

FM3D Facility Management 3D

FPS First-person shooter

GIS Geographical Information System

GUI Graphical User Interface

HUD Head-up Display

HVAC Heating, Ventilation, and Air-Conditioning

PLC Programmable Logic Controller

RPG Role-playing Game

RTS Real-time Strategy

SCADA supervisory Control and Data Acquisition

SG Serious Games

3D Tridimensional

VGE Virtual Geographic Environment

VIE Virtual Interactive Environment

VRML Virtual Reality

VRML Virtual Reality Modeling Language

Chapter 1

Introduction

The heart of a Smart Building or Smart Home is the Building Automation System (BAS) an automated control system that coordinates electric and mechanical devices installed in buildings, or homes, to improve the comfort and safety of the occupants while at the same time reducing energy consumption. One important component of a command and control system of large dimensions is the existence of a form of graphical representation of the location and distribution of devices and is crucial to the efficient use of the system. Through BAS, a user, should be able to graphically view the location of each device in the plan of a building, navigating in a fluid immersive matter through each floor and interact with each device or a group of devices in real-time, instantly perceiving its actions and the states of all controlled devices in the application. To design such interfaces there is a need to combine several complex variables, such as satisfactory system perspectives, fluid real-time rendering technologies, tools for modeling virtual geographical environments and geographical informations systems.

Command and control tools are developed by companies linked to the automation industry which do not, in general, have the motivation nor the ability to create sophisticated easy-to-use interfaces. As a result, most tools present huge gaps in user interface interaction, turning the interaction with the devices installed in the building an unpleasant and frustrating experience. Although conceptually simple, to the best of our knowledge, no command and control system has implemented graphic interaction functionalities in a satisfactory way. In video game genres, (e.g. such as the real time strategy genre) were users need to control multiple elements at the same, its used the overhead pull-out third-person perspective. This perspective is meant to supply the user with the ability to view all virtual world elements at the same time, permitting zoom in and out actions upon a given element, through an isometric view at any time. In the same manner, this concept can be applied in the interaction with BAS floor plans.

1.1 Problem Statement

Regarding the state of the art, Building Automation Control tools are often quite limited with respect to displaying spatial data. Many do not display data using any type of planimetric representation which makes navigating on spatial information quite limited. They lack of deficient navigation, use outdated interface technologies and can only be used by specialized personnel.

Tridimensional interactive environments are known for their characteristics of adherence to reality, immersion, natural interaction, visualization capabilities and adequacy to support simulation. They are effective at rendering spatial information and enable interactivity through direct manipulation.

Since 3D Interactive environments have been applied successfully for education and training purposes for some years, can they be used to improve the interaction with BAC systems? In particular, are there advantages in navigate through a 3D virtual representation of a building plant in real time and interacting with all existing building devices, and visualizing their status as well as their actions on the environment?

1.2 Goals and Contributions

This work aims at determining the usefulness of a 3D Interface for Building Automation Control. To that aim we will evaluate a 3D virtual facility model will be augmented with information regarding the space characteristics as well as the location and status of equipments, providing simple ways to control them. It is our aim to contribute to a satisfactory 3D Building Control system. In such system there is a plurality of software layers that are generally important, but in the scope of this work our main concern is the system's user interface. To build a proper 3D interface, the following goals must be achieved :

- Study the limitations of the current Building Automation Systems;
- Study existing visualization, navigation and manipulation techniques on 3D Virtual Interactive Environments to determine which ones can be transposed to the command and control of BAS;
- Study Computer Aided Design tools, Geographical Information System tools and how can they be used in the construction of a 3D virtual geographic environment;
- Select an appropriate tridimensional game engine according to the needs of command and control;

- Model a proper 3D environment;
- Implement the studied 3D environments' navigation and manipulation techniques;
- Implement the building devices and their behavior on the environment;
- Achieve realtime interaction upon navigation, command and monitoring activities;
- validate the build interface;

To validate the built interface and verify the actual capabilities of using a virtual 3D environment for visualization and interaction with integrated facility management information, a user evaluation study will be conducted. This study is to compare our prototype interface, applied to the centralized control of a building automation control system, with a corresponding legacy application, regarding both time effectiveness and user qualitative experience. The 3D virtual environment is to featur a part of the IST-Taguspark building where the legacy application Schneider Electrics' TAC Vista application is installed.

Ultimately, this work aims to an improvement on BAC systems by investigating the goodness of 3D environments in command and control systems, formalizing the requirements and techniques, needed to achieve satisfactory user interfaces. If successful, we intend to integrate our interface with the Lumina Platform. The Lumina is a software Platform intended to centrally manage building Automation systems. It is currently being developed at the INESC-ID in partnership with the area of Energy Efficient Systems of the MIT Portugal.

1.3 Document Organization

This document is divided in 6 chapters. The current chapter presents the motivation of this work as well its context and objective.

In Chapter 2, Literature Review, we introduce some of the most important concepts which are critical for understanding great part of this work. More precisely Section 2.1 describes the technologies and concepts relative to Building Automation Control. Then in Section 2.2, it is explored several aspects of interacting with Control and Monitoring Systems and in Section 2.3, we review Virtual Interactive Environments and discuss how they can be used to integrate several information domains into spatially visible information. Finally, in Section 2.4 we end with a chapter discussion.

Chapter 3, a 3D Building Control System it is composed of by three sections. More precisely, at Section 3.1, it is presented an overview of the problems inherent to current Building Automation

Control interfaces, discuss the 3D User Interfaces' views, commands, its general disposition and discuss the architecture that will support it. Then, in Section 3.2 we explain the architecture main modules and finally in Section 3.3 conclude with the chapter discussion.

Chapter 4 is referent to the implementation details regarding the components of our prototype, as well as the technologies involved and programming language used. As such, on Chapter 4.1 we will present such technologies and language, on Chapter 4.2 we will show some implementation details and conclude on Chapter 4.3.

Chapter 5 describes the evaluation of this project. It is composed by five sections, Section 5.1 in which we explain the methodology used at our evaluation, Section 5.2 Participants, were it is described in some detail, the test participants profile, Section 5.3 Tasks, were we present the tasks defined for the evaluation procedure 5.4 Results, were we discuss along two subsections, the qualitative and quantitative results obtained with our evaluation and finally Section 5.5 Discussion, were we discuss some evaluation remarks.

And finally Chapter 6, were the final conclusions are presented along two sections, the lessons learned from this work in Section 6.1 and the future work in 6.2.

Chapter 2

Literature Review

The study of possible applications of 3D Interactive Environments to the Building Automation Systems context is recent. Interactive environments have been applied for education and training purposes for some years, but when it is the case of explaining how to create proper 3D interactive interfaces for Building Automation there is no literature available.

In this chapter we will make a literature review of the concepts regarding Building Automation Systems, their controlled devices graphical representation and the visualization of their status within the existing user interfaces. Then we will explore several aspects of interacting with Control and Monitoring Systems. This is important to determine the formal areas of interaction concerning these systems. Then, we will review Virtual Interactive Environments and discuss how they can be used to integrate several information domains into spatially visible information.

2.1 Building Automation Control

The demand for more security, efficiency, convenience and comfort, has caused the level of automation in commercial and residential buildings to also increase Marcuse (2002). A building automation system coordinates electrical devices installed in a building where devices are attached to a dedicated digital network known as *fieldbus* network, enabling data exchange in the form of messages Merz *et al.* (2009). These networks have been used for many years in industrial automation and control, and opened the market to diverse application areas, such as home and building automation Dietrich & Sauter (2000). By comparison with traditional wiring, devices communicating through *fieldbus* networks are, in a sense, more decoupled. For example, a traditional wall mounted switch that would simply close an electric circuit, on a automatic control system, is an electronic device that sends a message through the *bus* to an actuator driving luminary. This decoupling grants a degree of flexibility that opens many new possibilities. Instead of delivering the message to the appointed luminary, the BAS can choose to redirect the message to another device, depending on the room layout, or simply not deliver it, to inhibit the switch when in after hours. Conceivably, the message could also be associated to multiple luminaries. There are many possibilities. All devices used in building automation, such as sensors, actuators, controllers, regulators and control panels, operate remotely and need dedicated communication systems to execute their functions and enable data exchange over *field buses* and networks in the form of messages Merz *et al.* (2009). Home automation is also building automation adapted to homes. The control is more of domestic activities, such as home entertainment systems, houseplant, yard watering, changing the ambiance scenes for different events and the use of domestic robots Kaur (2011). In this document both home and building automation as well as control and management systems(CMS) will be described as BAS.

2.1.1 Fieldbus Networks

A *fieldbus* network can be understood as a distributed device network of sensors and actuators nodes connected through a digital network Thomesse (2005). Sensors can be luminosity sensors, temperature sensors or occupancy sensors. Wall-mounted push-buttons, switches and dimmers are also considered sensors since they inject information on the network about which button was pressed or about the dimmer set-point. There are different kinds of actuators, the most commonly used are relays and dimmers that drive devices attached to them, such as HVAC systems, luminaries, blinds, windows or doors, valves among other. Each sensor or actuator contains an electronic interface that is responsible for, respectively, sensing electric signals and sending the appropriate messages into the network or, conversely reading data from the network and generating the appropriate electric signals. Each of these network interfaces is known as a *node*. Nodes send messages into the network that are received by other nodes, thus forming a distributed network. Plus, nodes can run embedded software applications. The main purpose for introducing such intelligent network is to *(i)* increase flexibility of device installations and *(ii)* the promise of sophisticated device behavior aiming at improving user comfort and and ensuring a rational usage of the building resources. Flexibility implies, that to meet new requirements, the behavior of the electric installation can be changed without undergoing rewiring (e.g., associating a switch with a row of luminaries should be a matter of configuration). Comfort implies that the devices attached to the electrical system respond more adequately to the requirements of the user in a given instance, for example automatically adjusting the room temperature to the user desire. Moreover, the building manager expects the automated control system to display

Table 2.1: Examples of *fieldbus* technologies and the applications they are used in taken from Merz *et al.* (2009).

Technology	Field of use
Controller Area Network (CAN) Davis *et al.* (2007)	Automobile engineering,BA
Process field bus (Profibus) Bender (1993)	Process and Factory Automation
Interbus White *et al.* (1996)	Factory Automation
Konnex (KNX) Konnex Association (2004)	Building Automation
Local Operating Network (LON) Corporation (1999)	Building Automation
Local Control Network (LCN) Hunt & Kalkkuhl (1996)	Building Automation
Modbus MODBUS (2002)	Building Automation
BACnet ASHRAE (2004)	Building Automation

a breath of automatic behavior that proactively reduces different types of waste, in particular reducing energy consumption. There are various types of *fieldbus* systems on the market. Their specifications (e.g. transfer rate, number of nodes, message type and reliability) vary according to the requirements of the applications they are used in. Table 2.1 presents a summary of existing *fieldbus* types and their particular application. The network span and number of nodes of a building control system is smaller than on industrial system. Hence, the network management complexity and overall installation cost are minimized. The aim of building automation is to provide better quality of life and, through improved convenience and enhanced entertainment features Schickhuber & McCarthy (1997). There have been defined several different industry standards for device communication, such as: BACNet ASHRAE (2004), KNX Konnex Association (2004), LON Corporation (1999) or Modbus MODBUS (2002), in addition to many other proprietary solutions. Some fieldbus technologies are being used in different types of applications. CAN is also used in building automation.

2.1.2 Centralized Control and Monitoring Systems

A *centralized control and monitoring system* (CCMS) increase the operator reach from having to act individually and locally on each peace of of a building or facility. These systems also allow the detection of abnormal conditions without being on proximity. Supporting the usage of technical alarms that indicate the need for repair and maintenance Kastner *et al.* (2005). In addition trend logs usage, provide valuable information for improving control strategies as well. This features are important for assessing the cost of operation and in performing scheduling maintenance. CCMS supply various levels of sophistication depending on the size of the building and desired operational function. The simplest system allows an operator to check the operational status of the heating, ventilating and air conditioning (HVAC), fire and security systems, and control various

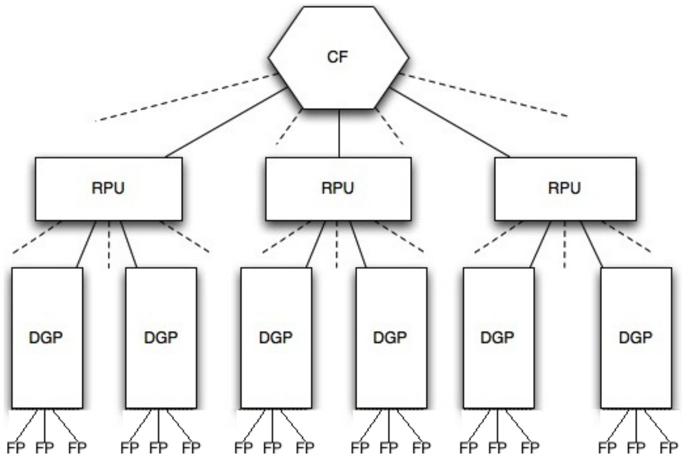


Figure 2.1: An example of a block diagram for Class B systems where *CF* represents the Central Facility, *RPU* represents the remote processing units, *DPG* represents the data gathering panels, the *FP* represent the system field points and the dashed lines indicate future expansions.

equipment remotely from a central console Haines & Hittle (2006). The most complex type of system are those that perform direct digital control (DDC), where a digital computer performs most of the work normally done by an operator, and other optimization and control functions Kastner *et al.* (2005). The market has defined several types of building automation systems, according with building sizes. These system are classified as *(i)* Class A, small monitoring and control systems for buildings with floor areas up to 20 000 m^2. These systems are designed to perform operations such as monitoring fire alarms and smoke detectors, security checks, and load cycling Elmahdy (1980). *(ii)* Class B Systems, similar to the Class A ones, being the difference that they can support larger buildings and minor building complexes. These systems can usually monitor about 2000 addressable points. When used for a small group of buildings or building complexes, the central control facility is connected to remote data gathering panels by means of one or more types of data communication links Elmahdy (1980). *(iii)* Class C Systems, this type represent the highest degree of sophistication in central control and monitoring systems and are usually referred as direct digital control (DDC systems) Froehling *et al.* (1985). They are often used for building complexes such as educational institutes and university campuses Elmahdy (1980).

A common type of centralized control and monitoring system is the SCADA[1] system. Conceptually SCADA is not a full control system, but rather focuses on the supervision level. It is positioned above the hardware to which it is interfaced, via PLC [2] and other commercial hardware modules Daneels & Salter (1999). A SCADA System consists of the three distinct subsystems, *(i) Human-Machine Interface*, which is the apparatus that presents the process data to a user,

[1] supervisory control and data acquisition
[2] Programmable Logic Controllers

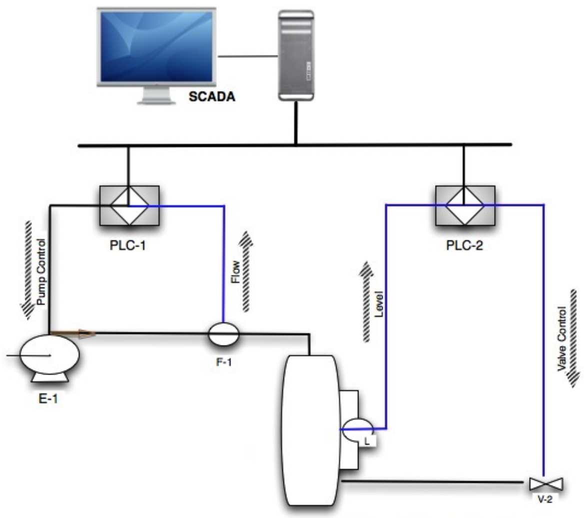


Figure 2.2: An example of a SCADA cooling water control process. In this process *PLC-1* compares the measured flow to the set point and controls the pump speed as required to match the flow to the set point. *PLC-2* compares the measured level to the set point and controls the flow through the valve to match the level to the set point.

and through which the user controls and operates the process. *(ii)* The *supervisory computer system*, who acquires data and sends commands controlling a given process. *(iii)* the *Remote Terminal Units* that connects to the sensors in the process, converting sensor signals to digital data and sends digital data to the supervisory computer system and *(iv)* the Programmable Logic Controllers, that are used as field devices because they are more economical, versatile, flexible, and configurable than special-purpose RTUs. The Communication infrastructure connecting the supervisory system to the Remote Terminal Units.

As can be seen the in figure 2.2, most control actions are performed automatically by the remote terminal units or by the programmable logic controllers . The functions that control are usually limited to basic overriding or supervisory level intervention. For example, a Programmable Logic Controllers may control the flow of cooling water through part of an industrial process, but the SCADA system can permit a user to change the set points for the flow, and enable alarm conditions, such as loss of flow and high temperature, to be recorded and displayed . Feedback control loop passes through the remote terminal units or PLC, while the SCADA system surveys the overall performance of the loop.

2.2 Interacting with Control and Monitoring Systems

One of the most important features of any automation system is the interface by which users interact with the system features of a sophisticated interface. Such interfaces permit one to program future operations and control different devices from a centralized system Stein *et al.* (2000).

For many years video game industry has studied to provide user's with intuitive and realistic interfaces through the usage game engines graphics systems. Current Building Automation Systems are not user friendly, and require highly trained personnel, familiar with the systems for its operation, costing time and money. Centralized command and control software for building automation should be created taking in account users needs by providing intuitive, graphical control interfaces. The interaction with these interfaces if happening in real time would provide a convincing feedback to the user and give the impression of natural interaction. A game engine based GUI when integrated with BAS would provide intuitive control of various electronic devices with minimum training thus reducing costs while at the same time contributing to a pleasing user In addition, user interfaces can also present a varying degree of sophistication, ranging from simple wall switches and control panels to touch screens or as this works intends to present, visual control panels that are rendered on demand in order to interact with a given device.

2.2.1 Aspects of interaction with Control and Monitoring Systems

When interacting with physical devices it is expected that for each user action, to exists a predefined perceived reaction Bales (1950). For instance, a user interacting with a particular bedroom lamp is presented whit the corresponding bedroom light switch. To that user, that light switch is perceived as the light ball interface and the action of turning it on, will cause the predefined reaction of the light, to be also turned on. But given the case of the light switch to not be in the same bedroom as the light ball, how could the user perceive if the light ball was actually turned on? In BAS systems, building controls aren't always in the same location as the devices that they are controlling, so there is the need to represent each device possible actions and state Wimsatt (2004). As there are different kinds of controlled devices, there is also different device actions and state representations. For instance comparing a light ball with a cooling fan, the first one may support on/off and intensity dimming actions, on the other and the cooling fan may support also on/off action and low/medium/high intensity functioning levels, booth needing different actions and state representations.

Device location and grouping are also functions provided in BAS Alkar & Buhur (2005). Through device grouping several devices can be manipulated simultaneously, simply by acting over a defined group. Grouping can also be achieved either by choosing manually in a virtual panel home layout witch devices will be assigned to the desired group, or implicitly by selecting a building floor, room or area and by doing so selecting all area containing devices. Such features, provide systems users with a huge building control flexibility, by automatically adjusting a large set of devices to a desired behavior towards a more rational use of a building resources.

Interaction through customized scenarios is also present as a desired feature in interaction with BAS systems Russ *et al.* (2003). With it, one could easily define a scenario where a group of devices would act upon. For example, when the BAS detects that nobody is at home, it would automatically turn off all lights, close all shutters and curtains and diminish all unnecessary energy consumption. As easily, on user arrival the system could automatically recognize the user, play the user favorite tune and adjust room temperature according with the user preferences.

2.2.2 Formal Interaction areas

Interaction with command and control monitoring systems can be arranged according to the type of activity, distinguished by their effect on the system as:

- *Command* activities are characterized by an action-reaction pattern, where one commands a system, group of devices, or individual device to perform the reaction task predefined by the chosen action input. This actions are almost instantaneous. A good example can be found in the light switch or even the fan of a HVAC system, where the user turns a light or a group of lights and that action is executed immediately or in the HVAC case, the user increases the room HVAC fan speed and almost immediately the speed increases. These task are used often in a daily basis, and can be achieved automatically and manually.

- *Monitoring* activities are meant to supply the user with the ability supervise CMS regular behavior. Monitoring activities act as a complement of the command ones, by allowing users to visualize the system behavior and comprehend if it is performing as commanded. For instance, through monitoring the illumination component of a building, one can understand if the lights that where commanded to be turned on are actually performing as expected by viewing their state, and if not act as required.

- *Commissioning* activities of CMS are responsible of administrating system scheduled changes and calibrations. A good example can be found in the summer-winter calibration and time scheduling of the exterior illumination of a building and interior temperature regulation, where in the winter the lights must be turned on much earlier than in the summer and the HVAC's air handling units set-point and calibration must be set to heat up the building instead of cooling it down. These changes are often commissioned to occur only seasonly, and over a great quantity of devices

Moreover, activities can also be classified according to functional areas. We can distinguish at least four relevant functional areas:

- *Alarms* are meant for monitoring and notification purposes, presenting information about unusual system changes and malfunctions, they are usually programmed to notify the system administrator about system malfunctions. Notification can be achieved through a computer, pager, phone or even an audible alarm.
- *HVAC* encompasses all interaction related HVAC related devices, from the HVAC central plant transformers and auxiliary power units for emergency power to local volume air-handling units.
- *Illumination* encompasses all interaction related to Illumination related devices, from photo-sensors, timers and dimmers to manual wall switches states.
- *Occupation* it is usually based on time of day schedules. On Occupancy mode, a CMS is meant to provide adequate lighting and comfortable temperature, often with zone-based control so that users on one side of a building have a different thermostat than users on the opposite side.

In order to evaluate and compare different market Control and Monitoring Software, a software comparison was made according with the observation of common tasks and subtasks commonly executed by building administrators on the observed control software. Prior to software comparison and evaluation, all tasks where classified according with their Functional Areas and Type of activity, as shown on table 2.2. The observed software where the TacVista installation on Instituto Superior Técnico, and Siemens Desigo installation on Lisbon's El Corte Ingles.

During the software observation it was noticed that several steps are performed during the execution of each task and that most of these steps are common to most task. In order to better comprehend the general task requirements a list containing these main steps was created and defined as a table of sub-task. As shown on table 2.3 each subtask is defined by a Sub-task ID and its description.

At last, with the definition of all mentioned metrics, it was created a software comparison functionalities table according with the observed software, main system tasks and needed sub-tasks that are needed in order to perform the main ones. In this table one can observe that for each

Func. Area	Type of Activity	Task Description	Task Id
Illumination	Monitoring	Visualization of a room illumination state	T1
	Monitoring	Visualization of the building exterior illumination state	T2
	Monitoring	Visualization of illuminated areas	T3
	Monitoring	Monitor manual action overrides over room illumination state	T4
	Command	Acting over room illumination	T5
	Command	Acting over exterior illumination	T6
	Command	Acting over common area illumination	T7
HVAC	Monitoring	Verification of Air Handling Units state (On/Off)	T8
	Monitoring	Verification of Air handling Units temperature set-poin	T9
	Monitoring	Monitoring Air Handling Units functioning features	T10
	Monitoring	Monitoring Air Handling Units Health	T11
	Monitoring	Monitoring air renovation sub-system efficiency	T12
	Monitoring	Monitor spent energy over each Air handling Unit	T13
	Commisioning	Change HVAC System temperature set-points	T14
	Command	Turn On and Off each Air handling Unit	T15

Table 2.2: Main CMS tasks classified according to their functional area and type of activity.

Sub-Task ID	Sub-Task Description
ST1	Navigate in to a space location
ST2	Navigate in to a device location
ST3	Visualization of the general state of a device
ST4	Visualization of the detailed subsystem's state
ST5	Command a device

Table 2.3: Sub-Task identification and corresponding description.

software it was executed all tasks form T1 to T15 and for each Task it was marked when a sub-task where required, not needed or unavailable in that system. This table is presented on table 2.4

2.3 Virtual Interactive Environments

"Virtual Reality" or "artificial reality" are some of the terms that have been used to describe a Virtual Interactive Environment (VIE) Ellis (1994). VI are environments are created through software and are meant to be presented to the user in such a way that the user suspends belief and accepts it as a real environment. On a computer, virtual Interactive environments are primarily experienced through two of the five senses: sight, sound. Also in new generations gaming consoles the experience is augmented with the addition of a third sense, touch.

The simplest form of VIE is a 3-D image that can be explored interactively through a personal computer, usually by manipulating keys or the mouse so that the content of the image moves in some direction or zooms in or out Roy (2003). More sophisticated efforts involve wrap-around

		Task Identification														
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
TacVista	ST1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
	ST2	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
	ST3	•	•	•	•				•	•	•		•	•		
	ST4									•		•	•		•	
	ST5					•	•	•							•	•
Desigo	ST1															
	ST2	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
	ST3	•	•	•	•				•	•	•	•		•		•
	ST4									•			•	•		
	ST5					•	•	•							•	•

Table 2.4: Functional Comparison of two CMS Softwares, where each task shows the sub tasks needed for the execution of each task. The ● sign is used to show that the sub task is needed to execute that current task.

display screens, actual rooms augmented through augmented reality or even haptics devices that let you feel the display images. Virtual Interactive environments are commonly divided into real environments simulations for training, gaming and education. In the following sections it will be described the types of VI environments and some applications used for their creation.

2.3.1 Video Games

Digital games, in their formats and genres, are a form of social and digital expression Fullerton *et al.* (2008). Video games became one of the most pervasive, profitable, and influential forms of entertainment Squire (2003). Conceptually a video game is an electronic game that supports interaction through a game interface, inputing commands through game controllers and generating visual feedback onto a video device. Video games are usually characterized by their genre, which will define game perspective and type of play. Complex video games are developed from game engines, which offer reusable components that can be reused to create sophisticated graphic interactive software other than games.

To design effective intuitive interfaces for a building automation system, several complex requirements have to be met, such as user desired perspectives of an application, fluid real-time rendering technologies, tools for modeling virtual geographical environments and geographical informations systems. In this section we consider such concepts, which can provide significant insights into the principles required for the design of effective interfaces.

2.3.1.1 Video Games Genres

Conventional video games can be classified, according to their game play, as First Person Shooter (FPS), Real Time Strategy (RTS) or Role Playing Games (RPG). In the FPS game play, the player interaction with its world, emphasizes combat and shooting from the point of view of the character controlled by the player. The first person perspective is used to give the player the feeling of being immersed, and allows the player to focus on aiming Pagan (2006). In the RTS game play, the player interacts with the world through a two dimensional map view, with a significant number of player controlled units (e.g an army and uses them to defeat their opponents). In the RPG, the game play is similar to the RTS the difference being that the player controls a single unit to explore a 2D (e.g. Dungeons and Dragons[1]) or 3D (e.g. World of Warcraft[2]) world. Also video game classification can be expanded in respect to the number of players of a given game, as single player game where other players can be simulated using game-agents, or multi player game where several players can interact in computer network or a virtual world.

In any game genre, a world perspective, or point of view, is a key factor to supply the user with an effective environment interaction. In fact, the same is true to at all kind of virtual geographic environments Shepherd & Bleasdale-Shepherd (2008). Mainly, there are used three types of video games perspectives, first person, third person and overhead pull-out third-person also known as *god view*.

In automation control system where users need to control multiple elements at the same, overhead pull-out third-person perspective could be a possible solution to supply the user with the ability to view all virtual world elements at the same time, permitting zoom in and out actions upon a given element, through an isometric view at any time.

2.3.1.2 Video Games Engines

Nowadays, building immersive complex video games is a hard and complex task often undertaken by a vast team of specialized software architects, programmers, writers, animators, that may spend years in production before launching a game. This complexity, has created the need to supply game developers with tools that simplify and speedup the creation of game components and the development process as a whole. One of these tool are game engines. Game engines offer reusable components that can be manipulated to bring a game to life. Loading, displaying and animating models, collision detection between objects, physics, input, graphical user interfaces, and even portions of a game's artificial intelligence can all be components that

[1] http://www.wizards.com/dnd/
[2] http://us.battle.net/wow/en/

make up the engine. By definition, game engines are the core software component of any video game with real-time graphics Wünsche, B. C. and Kot, B. and Gits, A. and Amor, R. and Hosking, J. (2005). Game reviews mention immersion as related to the realism of the game world provided by game engines Brown & Cairns (2004). Game engines can be divided into two categories:

- The open source ones like OGRE 3D[1], Delta3D[2],Unreal Developent kit[3], Irrlicht[4], Crystal Space[5], and The Nebula Device 2[6] among others.

- And the proprietary ones like Unity3D[7], RAGE[8], CryEngine[9], Unreal Engine[10], Avalanche Engine[11] among others.

All game engines have different characteristics according to their predefined purpose which is usually dependent on the game genre. There are engines that provide only 2D or 3D graphics and many of them don't support cross platform usage. Most open source engines provide the more basic functionalities, often relying on third party tool-kits and libraries to provide the remaining aspects playable games. For example OGRE 3D is only a 3D Rendering engine. Other features like sound can be accomplished through the use of a cross-platform 3D audio Application Programming Interface (API) like OpenAL[12]. On the other hand, proprietary engines provide a complete and integrated functionalities, at the downside of the licensing costs. The kind of game or application to be developed determines the choice of the appropriate game engine. A game application that relies extremely on good particle and physics simulation, should preferably be implemented on an engine that already supports those functionalities or that allows integrating them easily. Thus a given engine should support the main features of game design but not necessarily all of them Korva (2004-2005). Using a 3D game engine, in an automation control system, would bring several advantages, such as real time rendering, fluid zoom and plant navigation.

[1] http://www.ogre3d.org/
[2] http://www.delta3d.org/
[3] http://www.udk.com/
[4] http://irrlicht.sourceforge.net/
[5] http://www.crystalspace3d.org/
[6] http://nebuladevice.cubik.org/
[7] http://unity3d.com/
[8] http://rage.com/
[9] http://mycryengine.com/
[10] http://www.unrealengine.com/
[11] http://www.avalanchestudios.se/
[12] http://connect.creativelabs.com/openal/

2.3.1.3 Video Games Interfaces

In video games, player immersion strongly relies on spacial representation and on the so called interface elements. Video game interfaces are the means through which users interact, in order to accomplish their objectives. All interaction, both in the game and its spaces, is obtained through its gaming interfaces, which can be divided in to physical and logical game interfaces Taylor (2002). Physical game interfaces or game controllers (e.g. mouse, keyboard, game pads, joysticks or steering wheels, among others) are devices meant to supply users with means of effective game interaction, through their usage, users can directly issue commands to a game. To supply user immersion, physical game devices have evolved from general purpose types of software interaction devices, like the mouse or keyboard, to more game type specific ones. For instance, flight simulation games achieve a greater sense of realism and immersion when played with joysticks, than with a keyboard or a mouse simulating in a more realistic manner real aviation commands. Common physical controllers present multiple interface roles, i.e., it is normal for a user to be able to accomplish multiple effects with a single controller by combining different buttons in several sequences. A great example can be found in the Hugely successful Capcom's Street Fighter game series controls in which players cast different fight tricks according a predefined controller button sequence[1]. In order for game controllers to be used with a variety of software and tasks, all controller operations must be capable of being linked to user operations in some flexible way. For example, while playing a race car game in a Xbox360 controller, the button responsible for accelerating must be properly assigned to the accelerate operation. That is obtained through a correct controller button function mapping i.e. game functions are mapped into controller inputs, taking in high consideration how to effective enable users to control their software to accomplish the required tasks with little effort. The way that functions are mapped into input game controllers can determine the success or failure of a game Pagulayan *et al.* (2002).

Logical game Interfaces are visual feedback representations of the game. One of the most common logical game interface is the Heads-up Display. Historically the Heads-up Display (HUD) name came from the head-up displays used in pre-World War II for military fighter aircraft[2]. Its purpose is to supply visual information to the player, by providing a collection of persistent on-screen elements, whose function is to indicate player and world status. These interfaces act as virtual sensors that present users with real time essential on-screen information. HUD elements are commonly used to show, among many other things, the player direction in the world map, how much health the player has, its game rank and other players position in the world map[3]. These features make the HUD as an invaluable information conveying method tool Rhodes (1997).

[1] http://www.streetfighter.com/
[2] http://www.brighthub.com/video-games/console/articles/104980.aspx
[3] http://www.gamasutra.com/view/feature/2538/off_with_their_huds_rethinking.php/

2.3.1.4 Serious Games

Serious games are a category of video games that use the *video game-learning* concept, where video games can be used to educate users about a certain subject, expand concepts, reinforce development, or assist them in learning a skill as they play in a immersive and amusement manner. They are often used in the advertising, simulation and education areas and are commonly designed to run on personal computers or video game consoles Susi *et al.* (2007). Since the 90's there have been a vast interest in applying serious games to new purposes expanding serious game usage to several areas. As a result at the *Serious Games Summit at the Game Developers Conference 2008* it where presented an accepted taxonomy referring the current state of the serious games industry, the largest serious games areas of applications and main usage, currently the main ones are government management, defense, healthcare, marketing and communications, education, corporate and industry Sawyer & Smith (2008). In this taxonomy it is predicted the use of serious games (SG) for command and control purposes at the Industrial area which provides the possibility of SG application for building automation systems unexisting at the moment and a gap we intend to explore.

2.3.2 Computer Graphics Applications

Rendering and animating objects, such as buildings, floors and devices, displaying their usage status and information, are powerful concepts that can applied to automated building control systems Freeman *et al.* (1998). Such applications could supply users whit visually attractive tools to simplify building control and management by providing a better object graphic representations. In this section we will explore the concepts are useful in conceiving such applications.

2.3.2.1 Computer Aided Design

Computer Aided Design (CAD) applications are software packages for drafting and drawing of buildings (and also complex objects, like mechanical parts and even molecules), through the usage of primitive entities such as lines, polylines, arcs, circles and text. In the building context, CAD software is often used by architects, for planning and designing of buildings. The artifacts created by a CAD system are not mere drawings but objects that may have relationships among them. For example, in a house CAD plan a door object has a relation whit the associated wall that contains it. Moving will cause one door to move along. In addition, CAD software is capable of creating traceability links from objects into specification and construction documents or internal policies and relations documents. These links are useful to determine, which areas ore objects are likely to be affected by changes in the documentations. Spacial properties such as volumes and areas are updated automatically.

Rendering, mesh modeling and visualization are also available in CAD software, allowing architects, builders and especially buyers to navigate and visualize through several perspectives how the final building will look like Heesom & Mahdjoubi (2004). It helps to end all ambiguity about what the final outcome of a building will be, which is always a problem with 2D drawings. The same concept could be transposed to building control interfaces. For instance any user could have the ability to define their preferred system perspectives and scroll through a building plant, easily identifying all building areas and their containing devices.

Figure 2.3: Example of the CAD capability to render a 3D object. In this case a 3D isometric perspective of a building floor plant.

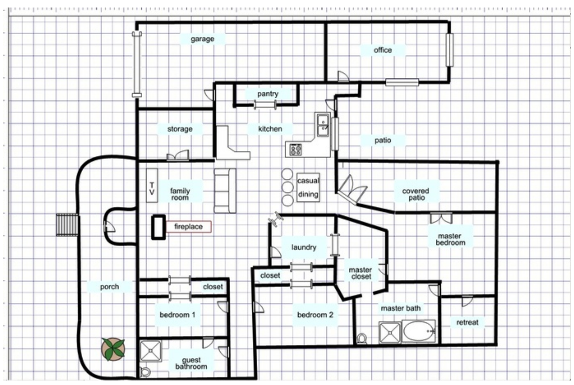


Figure 2.4: Example of a common 2D CAD representation of a building floor perspective.

As it can be seen in Figure 2.4, through a 3D perspective all floor areas and devices can be perceived naturally to anyone, in the other hand Figure 2.3 2D building perspective, presents to the unfamiliarized user an awkward, unnatural and ambiguous view of all devices.

2.3.2.2 Computer Aided Facility Management

Computer Aided Facility Management (CAFM) is the support of facilities management through information technology. Due to the growing complexity of large facilities, CAFM tools are meant to assist users in managing various activities within the facilities Keller & Keller (2005). Its common application spreads from maintenance and operations, facility budgeting, accounting, construction and project management, space and equipment inventory, architectural and interior planning, space forecasting among others Elmualim & Johnson (2009). Also, through CAFM users are meant to manage all facility assets, equipment locations, stock purchases, equipment replacements, work history carried out on equipment or assign strategies used in asset management with instructions to do so by schedule. CAFM systems are usually database oriented with awareness of a building which could integrate with knowledge-based systems for predicted automation control. The idea extending the functionality of a standard management tool capable of handling facility management and building control networks, is essential in practice and can be achieved by integrating CAFM systems with BAS to obtain a unified control software utility Himanen (2003). Such utility could present graphical representations of all controlled facility space and even show detailed inventory management and ordering facilities, together with barcode reading capabilities, so that even small consumable devices such as printer cartridges could also be recorded in the software database. Also through a Interface systems perspective such software should also could be complemented with CAD representations, in fact CAFM most systems that where integrated with CAD have been proven most effective Elmualim & Johnson (2009).

2.3.2.3 Geographical Information Systems

Geographical Information System (GIS) are systems that capture, store, analyzes and present data associated to geographic location data(i.e spatial data). It merges cartography, statistical analysis and database technology in a information system Maguire (1991). GIS applications display maps created using CAD software, and then, data related to several spatially located variables (i.e illumination, temperature data, among others) is overlaid on them. GIS systems are specially effective at presenting visual representation of spatial data, aiming at a more efficient analysis Rivest *et al.* (2005).

In building control, a GIS application can be used to better manage a building by improving information access and providing clearness of planning to the decision-making process Alesheikh & Behroz (2002). This result of the improvement in data visualization and cost savings. There are some well known cases of successful GIS implementations in large facilities, such as university

campuses (e.g University of Texas[1], Canadian Carleton University[2] or National University of Singapore[3]).
One advantage of a GIS with 3D modeling for building control would be 3D information query, spacial analysis, dynamic interaction and spacial management of a building Keke & Xiaojun (2010). Granting the ability to automatically monitor and visualize all building areas by illumination, occupation or other spatially located variables and manage them accordingly. For instance one could visualize electrical power consumption of the different building areas and improve efficiency consequentially reducing power costs.

2.3.2.4 Building Information Moddeling

Building information modeling (BIM) is a process of generation and management building data during its life cycle Lee *et al.* (2006). It allows for virtual, third dimensional models of facilities to be produced with reliable proportions giving the ability to work directly in the building's structure. Its very useful when changes need to be made, because BIM software makes necessary changes to the entire structure to compensate. It also works as an integrated database of coordinated information to which many participants in the design process contribute and modeling it provides short term and continuous analysis of a project design, schedule, scope, cost information and other matter Zeiss (2011). BIM tools are different from CAD tools, in the scale that a set of toy soldiers is different from a battle-oriented computer game, it supports on-line simulation of a design, on-line simulation of construction (called 4D CAD), on-line simulation of a building's operation, mechanically as well as the people organizations within it. A growing number of case studies have shown the benefits to users who have used a building model to apply BIM technology(see [4]). Building models and BIM technology are predicted to become the standard representation and practice for construction within few years Eastman *et al.* (2011).

2.3.2.5 Virtual Geographic Environments

Virtual Geographic Environments (VGEs) are defined simulations of physical and human geographical environments. These systems where conceived to empower geographers carrying out research work on elaborate geo-problems in an effective way Lin (2006).
A VGE features sophisticated representations of geographical data and incorporates technologies such as GIS and 3D rendering Lin *et al.* (2003). Before a landscape, a city or even a simple

[1] http://www.utexas.edu/maps/
[2] http://www2. carleton.ca/campus/
[3] http://www.nus.edu.sg/ campusmap/
[4] http://bim.arch.gatech.edu/reference.asp?mode=case&id=519

building can be produced, the environment is accomplished through GIS. This process is often performed in a manner where users are given access to the VGE world, in order to provide the chance for users to understand and explore the environment, enabling them to adjust diverse components, in the effort to solve problems that can realize better designs.

Nowadays, there are several VGE applications are mainly targeted to social interaction, through avatars (i.e representation of a user or the user's alter ego or character Parrinder (1970)), where users explore a virtual world, meeting other users socializing and virtually participating in several individual activities Donath (1997). Some well known examples of such applications are the Second Life[1], CyberTown[2], Active Worlds[3], among other.

2.4 Discussion

We discussed the limitations of the current Building Automation Systems state of the art regarding: the graphical representation of the location of devices and the visualization of their corresponding status within the existing user interfaces for command and control of large BAS. We reported possible ways to Interact with Control and Monitoring Systems according with specific formal and functional interaction areas. We analyzed several applications usage of Virtual Interactive Environments (VIE) and computer aided design tools employed in the VIE construction.

From this study we observed that existing BAS tool interfaces are often quite limited with respect to displaying spatial data. Many do not display data using any type of planimetric representation which makes navigating on spatial information quite limited. This forces users to switch to another screen to analyze the information regarding different zones. Facilities plans are often static pictures used for navigating from one space to another. These tools exhibit a general lack of spatial dynamism because they do not offer smooth transition between space zones. Another important aspect of navigation is alternating between different levels of detail, between aggregated data and detailed data. Management, supervision, and diagnostic activities require to quickly switch form a managerial into an operational view alternating between indicators at different levels of aggregation or from an overview perspective to a detailed inspection perspective. 3D interactive environments are known for their characteristics of adherence to reality, immersion, that they natural interaction, visualization capabilities and adequacy to support simulation, they are effective at rendering spatial information and enable interactivity through direct manipulation. The application of 3D Interactive Environments could be used to overcome the current BAS tools interfaces limitations by implementing a 3D interactive interface integrating several information domains into

[1] http://secondlife.com/
[2] http://www.cybertown.com/main_nsframes.html
[3] http://www.activeworlds.com/

spatially visible information.

Chapter 3

A 3D Building Control System

From the information gathered about the current literature, BAC tools and their real user needs, we designed a 3D Interface for a Building Control System which hopefully will improve the interaction experience with a BAS system by providing the ability to access and control building information systems in a more efficient manner and at the same time providing a more pleasing experience for BAS users. To support the 3D interface we devised a modular architecture design meant to allow an easy adaptation to different technologies.

3.1 Overview

Current Building Automation Interfaces are confusing, too technical and hard to understand. They lack of deficient navigation use outdated interface technologies and can only be used by specialized personnel.

Through a Building Control system interface, it should be possible to navigate through a 3D virtual representation of a building plant in real time and interact with all existing building devices, visualizing their status as well as their actions on the environment. Current 3D Virtual Interactive Environment technology provides features such as real time interaction and 3D navigation that would permit to render 3D virtual representations of devices allowing to simulate their effects on a 3D building environment, but as far as we know, 3D Virtual Interactive Environment technology have not been applied to the Building Automation domain.

It is our aim to contribute to a satisfactory 3D Building Control system. In such systems there is a plurality of software layers that are generally important, but in the scope of this work our main concern is the system's user interface. In its conception, our interface it will be featured by a 3D interactive environment meant to provide access to building information management in a multitude of platforms, from mobile devices to desktop computers through on-line interaction. This is considered a major benefit if we take into account the freedom of the geographical location of the professional charged of the facilities management and offers a powerful, yet easy, way to supervise and control small, medium and large facilities.

Regarding how to interact with such environments, we convened the visualization concepts that are common to command and control simulation games and are effective in this type of interaction. In such games types, the user make use of two view types: a main view, where most interaction occurs allowing the user to navigate in the environment, from a global viewpoint to detailed local exploration and a mini-map view, where the user is able to visualize the complete environment, offering a fast and easy way to change from one point to another.

3.1.1 User Interface

Our interface intends to make use of simple controls to help the user explore the environment, inspecting and commanding several devices within a building. To assist the user in navigating through the 3D model, our interface will offer two distinct views of the building simultaneously, the main view and the mini-map view:

Main view is the broader of the two views and where all device interaction occur. To navigate in the main view area it is available at the most right part of the screen three navigation control components: the rotation controls, panning controls and zooming controls. In this view, device visualization by functional area is also available, these options are available at the top left area of the screen and they act as functional area filters, allowing to only render the functional areas that the user selects. The room search functionality is at the top right of the screen and at the top center of the view is the building floor identification.

Mini-map view is the smaller one located at the bottom left corner of the screen. This view allows users to view the complete virtual building and choose the floor that will be displayed at the main view. Through this view it is expected to improve the user awareness of the virtual environment.

Figure 3.5 shows the sketch of the purposed user interface, illustrating the layout of its components, signalizing the main view, the mini map view, the navigation controls and visualization options.

In comparison with our preliminary interface sketch our final prototype interface aspect remained relatively unchanged. To the final interface it was added some floor changing menus located at

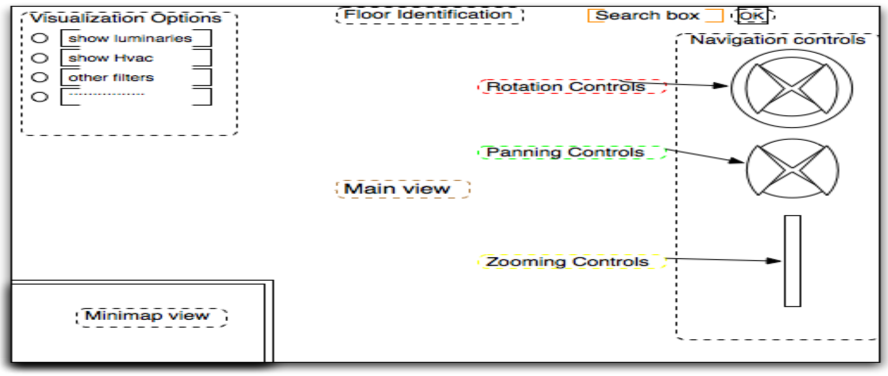


Figure 3.5: Proposed Interface sketch, where at the bottom left corner is the mini map view, on the top left corner are displayed the visualizations options, at the top center of the screen is the identification of the current viewed floor. The building division search box are at the top right corner, and at the right side are all the Navigation controls. Finally, the main view occupies all the remaining visualization area.

the bottom right corner of the screen that were meant to help users to better identify the current selected floor and as well as supply a more familiar floor selection interface. Also the building division search box changed place from the top right corner to the top left corner above the visualization options.The final interface is depicted it Figure 3.6

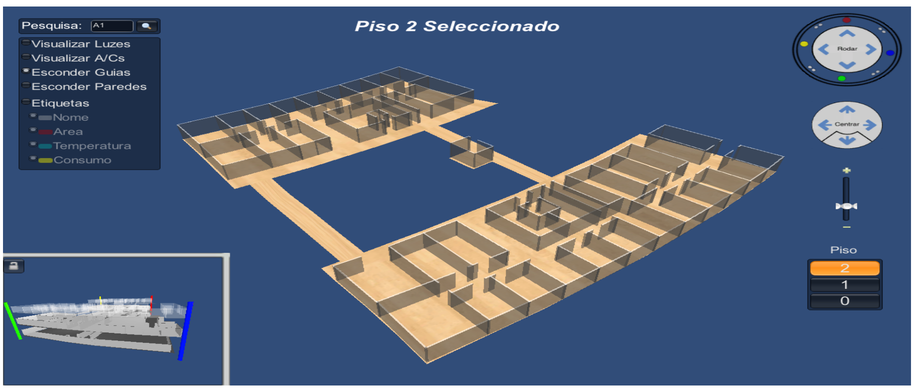


Figure 3.6: Final prototype interface, where at the bottom left corner is the mini map view, on the top left corner are displayed the visualizations options, at the top center of the screen is the identification of the current viewed floor. Now the building division search box is located above the visualization options, and at the right side are all the Navigation controls. Some auxiliary floor changing controls are located at the bottom right corner. Finally, the main view occupies all the remaining visualization area.

3.1.2 Architecture

To study the adequacy of 3D interactive environments for rendering and exploring integrated BAS information, we devised a modular architecture approach that allows easy adaptation to different technologies. The architecture will consist of four layers: device and sensor; network; application and interface.

Devices and sensors hardware layer composed by all all control and monitoring physical devices of a given building.

Network used for communication between the application layer and the device and sensors one. In this layer communication will be handled through the use of *fieldbus* technologies, being the main ones described in Section 2.1.1 Fieldbus Networks. This layer can be classified in two modules according with the flow of the communication using the *fieldbus*. The upper network communication and the lower network communication module. The upper one will handle all communication between the *fieldbus* and the application. On the other hand the lower one will encompass all communication between the *fieldbus* and the device and sensor layer.

Application this layer is divided in two components: the application backbone and the application model.

- The Application Backbone handles all the exchanged information with the lower layers. It is connected to the application database and it is responsible for the services that will be supplied to the application model component.
- The Application Model component has access to the control, monitoring and commissioning services that are supplied by the backbone. In this component is where all graphical application resource models are defined, as well as all graphical interaction and physics. It is here that the graphic engine resides and it supports two types of access to the same model: local and distributed. Their main difference resides essentially in the graphical and performance output, that is perceived by the interface layer.

Interface This layer is composed by two types of interfaces that have access to the application model of the application layer. These types of interfaces are the local interface and the browser interface.

- The local Interface is the one that it is executed in a local computer as a common program. This interface, as a stand alone application, has better access to the local computer's resources, which allows the program to make full use of the computer resources as well as the graphics board ones. As a result this distribution is faster, fluid and with better graphics quality.

- The Browser Interface, being the one executed by the internet browser, is not as responsive or capable of better graphics quality as the local one. Despite the mentioned limitations, this type of interface approach has the advantage of being able to supply the user with the ability to control a building in every internet browser connected communication device he carries.

Notice the we only concentrated in developing the interface and application model component of the application layer. In Figure 3.7 we can see a diagram of the proposed architecture as well as the components that we worked on, signaled in a darker grey.

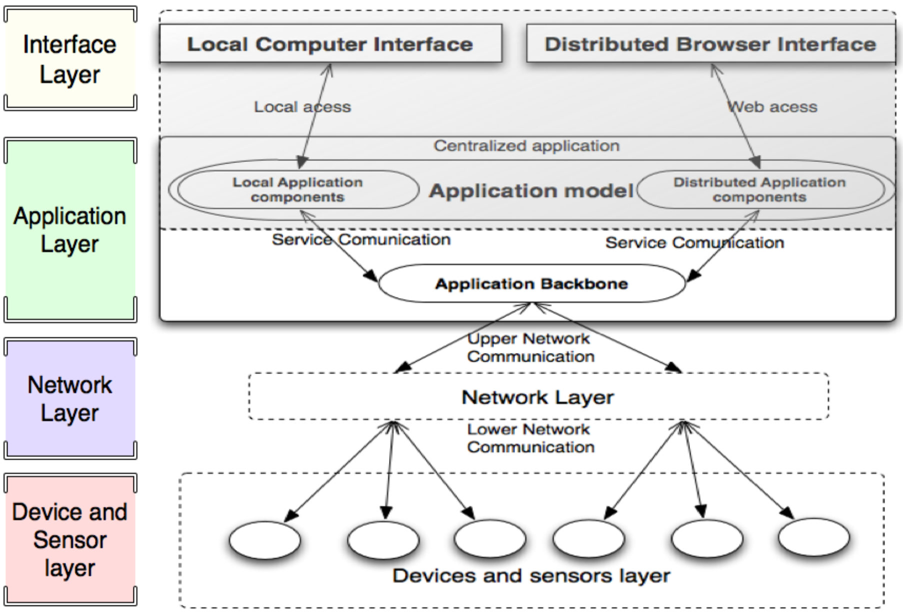


Figure 3.7: Architecture of the 3D building control system, can be distinguished by four layers namely: Interface layer, Application Layer, Network Layer and Device and Sensor Layer. The darker area represents the part that was implemented.

3.2 Architecture Modules

The beauty of a modular architecture is that you can replace or add any module without affecting the rest of the system. In the case of this architecture there is an exception, the GUI Command Module.

The GUI Command Module is the one that connects and relays commands to all others. Without it user interaction with system would no be possible. Through it, the Interface GUI is created and it is defined a screen area for the Command components that will execute each of the other modules.

The remaining main modules are the Device Module, the Tags Module and the Navigation Module Each of them is responsible for their behavior according with the users' command. Illumination and HVAC is delegated to the Device Module, navigation to the Navigation Module spatial information monitoring to theTags Module.

The main benefit of this Modular disposition is that it allows us to edit, add or remove any module (other than the GUI Module) without affecting the rest of the system.

3.2.1 GUI Command Module

The GUI command component is the one by which a user interacts with our prototype. It is located at the application layer and it is used to command the other modules. It is composed by six components, Interface GUI, Illumination Command, HVAC Command, Tag Command, Extrusion Command and Navigation Command. Figure 3.8 shows a representation of this architecture.

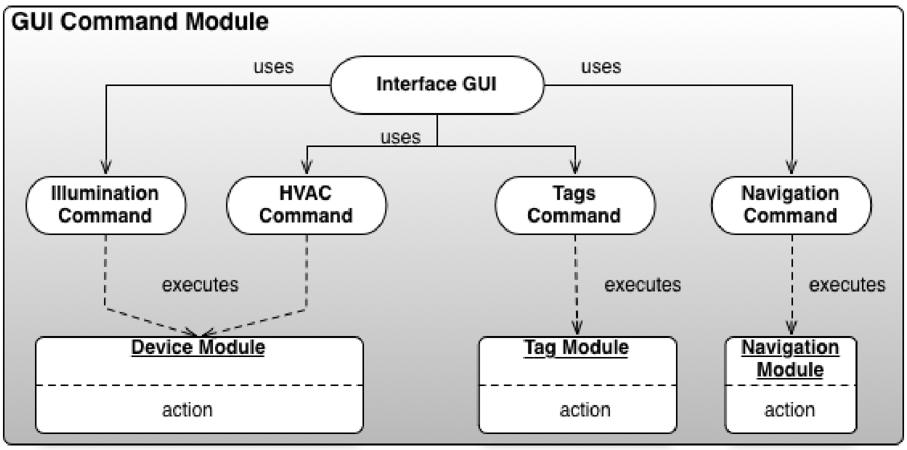


Figure 3.8: GUI Command Module architecture regarding its main components and their interaction flow, where the Interface GUI component uses all the connected Command components to command the Device Module, the Tag Module and the Navigation Module.

Interface GUI is responsible for the state, positioning and action upon all GUI interface sub-components. It is through this component that any user can relay commands that will act upon the main modules.

Illumination Command is a GUI interface sub-component. When required, it relays the user illumination functionality commands to the Device Module. With this command a user can activate or deactivate all luminaries and their respective effects on the building environment, as required.

HVAC Command is a GUI interface sub-component. When required, it relays the user HVAC functionality commands to the Device Module. With this command a user can activate or deactivate all HVACs and their respective effects on the building environment, as required.

Tags Command is a GUI interface sub-component. It relays the user Tag functionality commands to the Tags Module. Through this component a user can choose to activate or deactivate all existing Tags. When activated, it allows to select the Tag Types that will be rendered.

Navigation Command is a GUI interface sub-component. It relays all user navigation commands to the Navigation Module such as zooming, panning, rotating and camera selection.

The visual aspect of the Command Components is defined within each one. This allows an easier customization. The Interface GUI only has the ability to position the Command Components in our screen and define their starting selection state.

3.2.2 Navigation Module

This module is responsible by camera navigation and world visualization. In our case we need to control the two camera types that are responsible for the visualization of our 3D environment, the main camera and the mini-map one. The mini-map camera is meant to rotate only around their local Up-Axis, while the main is required to support both Up and Limited Left-Axis (Figure 4.18). It was required that, at some point, both camera's visualization perspective could become synchronized independently of their local rotation angle. To do so we designed an architecture defined by three main components, the Camera Manager, Main Camera Control and the Mini-map Camera Control, depicted at Figure 3.9.

The Main Camera Control works as an interface acting upon our Main Camera. Here is where we set the camera position, rotation and zooming according with the parameters received.

The Mini-map camera Control also works as an interface. This one is responsible for the Mini-map camera actions. It only allows Up-Axis rotation.

Camera Manager this manager uses both Mini-map and Main Camera Control components. It detects which camera is selected and applies the user commands to the selected camera. When the synchronization mode is turned on, it matches the Main Camera Up-Axis rotation with the Mini-map one and while this mode is active it synchronizes both camera vertical rotations.

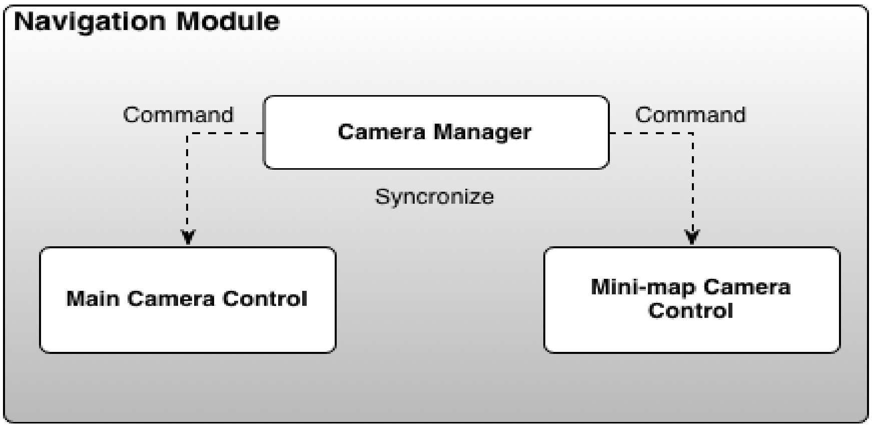


Figure 3.9: Navigation Module three main components: Camera Manager, Main camera Control and Mini-map Camera Control, where the Camera Manager is used to command both Main Camera Control and Mini-map Camera Control Components.

3.2.3 Devices Module

A major problem concerning the visualization of information is the possibility of its superposition. In our prototype all information is transmitted visually. So, in order to avoid the problem previously referred, we established as a requirement, the possibility to filter which information components should be active and consequently its objects rendered. At our GUI filters menu, we select the information type that we want to be displayed, which means that we need to be able to control the state of all luminaries and HVACs, activating and deactivating them as required. To do so we designed an architecture defined by three main components, the Device Manager, Devices Registry and the Device Client, depicted in Figure 3.10.

Device Client is a Component attached to every Device. After the object creation phase, it registers the Game Object that it is attached to the Device Register.

Device Registry has a container with all existing Device Game Objects. When necessary, it is responsible for feeding all Devices Client objects of a given type to the the Device Manager.

Device Manager is responsible for activating and deactivating all existing Game Objects of a certain type, as well as their children. When necessary, it connects to the Device register to obtain all registered Device Objects of the desired type.

After the creation of all Game Objects, each Game Object that has a Device Client component attached is registered at the Device Registry through the Device Control component. On run time, when the user unselects the Illumination or the HVAC functionality of the prototype, an event is

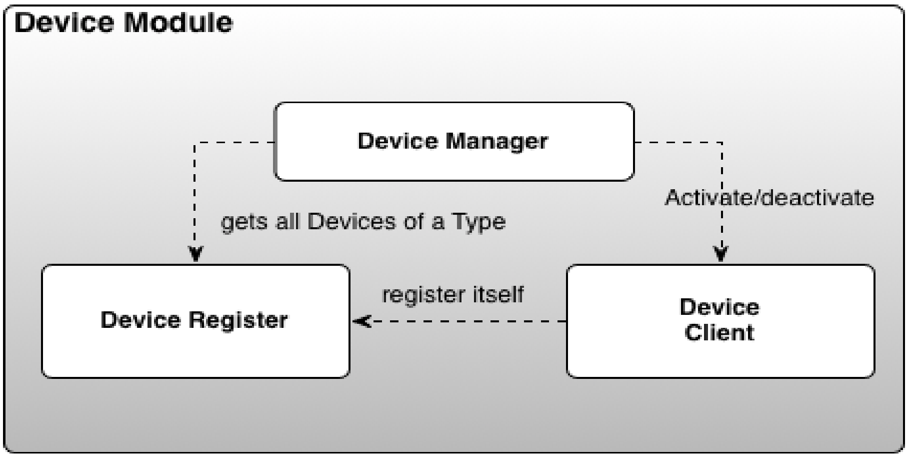


Figure 3.10: Devices Module main components regarding the main components and their interaction flow, where on creation each Device Client is registered at the Device register component to later be managed by the Device Manager.

triggered on the Device Manager to request all existing Game Objects of the unselected functionality type from the Device Registry. After that, the Device Manager deactivates the obtained Game Objects as well as their children. A similar procedure occurs when the user performs a re-selection of the unselected functionality of the prototype, thus reactivating it.

3.2.4 Tags Module

Through the researched literature we understood that a physical space, such as a building, is composed of several other types of spaces. These spaces can be both of physical and logical nature. A logical space may be defined as a logical aggregation of physical spaces. Each space may contain several types of sensors who are constantly performing readings on their environment and present many types of properties, which defines them. Also, each type of space may have exclusive types of devices only applicable to them. Given the heterogeneity of all mentioned devices, its readings are commonly conveyed to us through different types of metrics that can be displayed through to different units.

Since our Tags Module is responsible for the visualization of most space related information, we designed it to support a great number of space related information types from both physical and logical spaces in a modular way. The great number number of devices types, space types and property types created a huge complexity of variables that were needed to be taken in consideration while designing this module architecture. Given the complexity of this architecture, we will describe it in a modular manner. Figure 3.11 shows an overview of the Tags Module architecture.

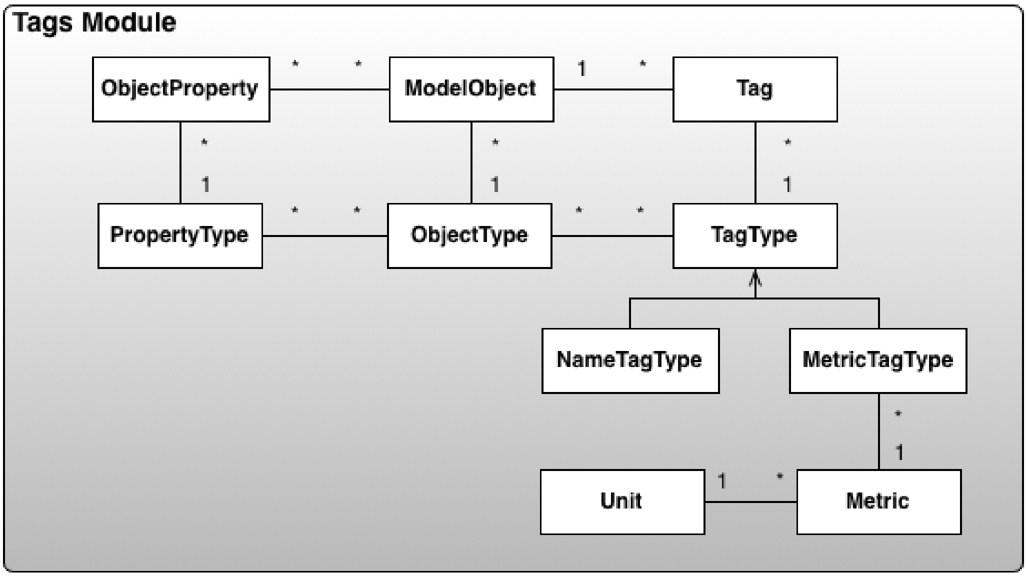


Figure 3.11: Tags Module UML diagram overview, showing the connections between the main components used to create spatial information 3DTags.

The ModelObject is an abstract class from where devices and concrete spaces (such as Rooms and *Nucleos*) extends from. Every object extended from this class has an ObjectType which will help define its ObjectProperties, and several Tags associated. The ObjectType is an abstract class from whom SpaceObjectType and EquipmentObjectType extend, furthermore both SpaceObjectType and EquipmentObjectTypes are not a final concrete form of the of the ObjectType but merely specializations or sub-classes. The final concrete forms that extend from the SpaceObjectType are the NucleoObjectType and the RoomObjectType. They are defined by the PropertyTypes they contain and will be used to create the ObjectProperties of the concrete forms of the ModelObject. The ObjectProperty is an abstract class from whomPercentObjectProperty, MetricObjectProperty and NameObjectProperty extend. Each of these concrete forms has a PropertyType associated. PropertyTypes work as models by which ObjectProperties will be created. Figure 3.12 illustrates a partial view of theTag domain; it is possible to visualize the concrete forms of the ModelObject, Object Type, ObjectProperty and PropertyType and well as the relation between them.

The Tag class has the needed parameters to create a 3DTag. Each Tag class is related to a single ModelObject and it is defined by a TagType. The TagType is an abstract class form which the NameTagType and the MetricTagType are extended and its concrete form will define the type of information that the 3DTag will present.The difference between these two tag types is that while the NameTagType will make the 3DTag display a name, the MetricTagType has a Metric

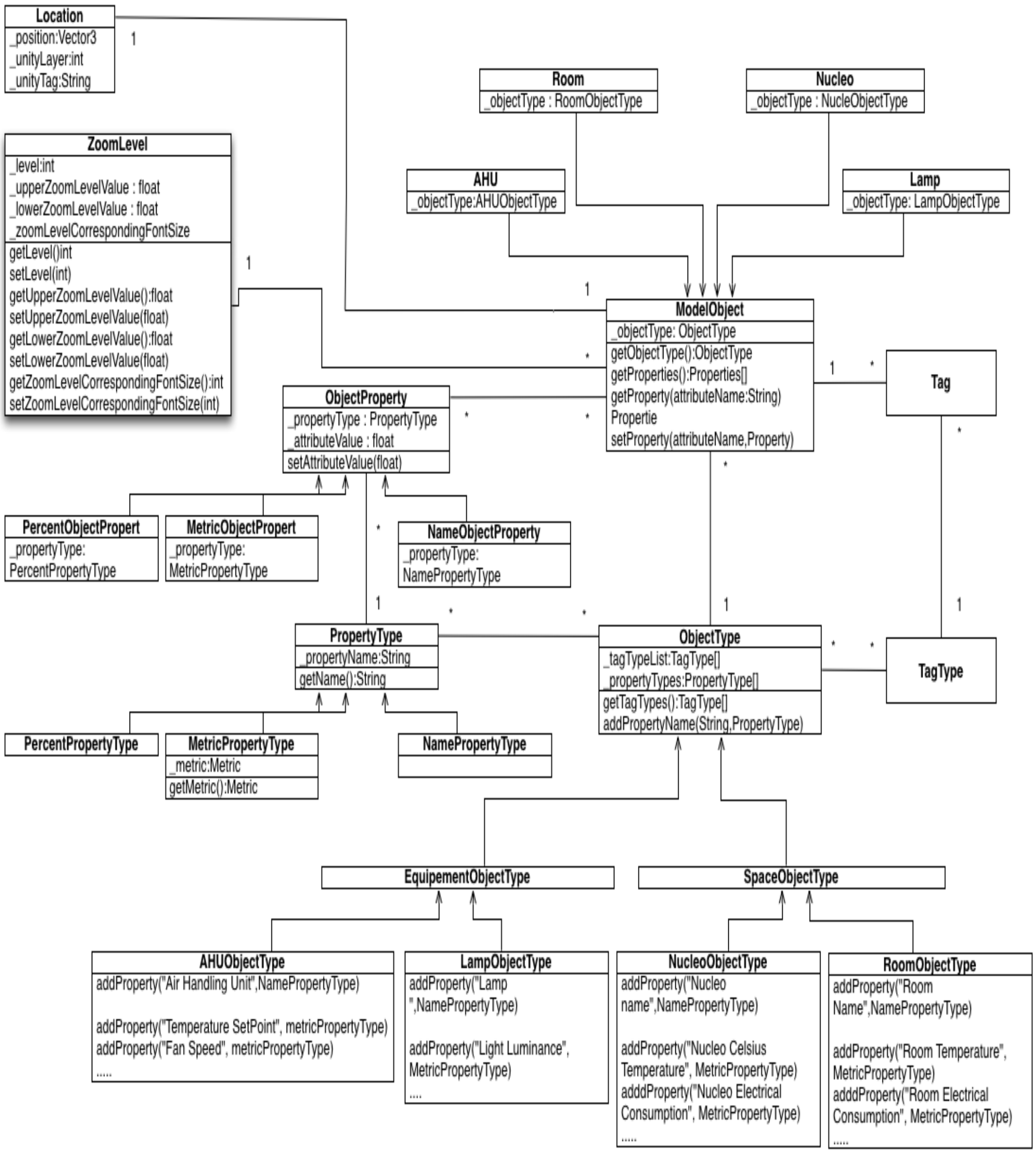


Figure 3.12: Partial diagram of the implemented architecture of Tag domain, it is possible to visualize the concrete forms of the ModelObject, ObjectType, ObjectProperty and PropertyType and well as the relation between them

associated to it which will cause the 3DTag to display a value and its metric. The Metric is an abstract class defined by its name and Unit, some of its implemented concrete forms are the PowerConsumptionMetric, CostConsumptionMetric and the AreaMetric. The Unit is the measure by which the Metrics will define a given value. Figure 3.13

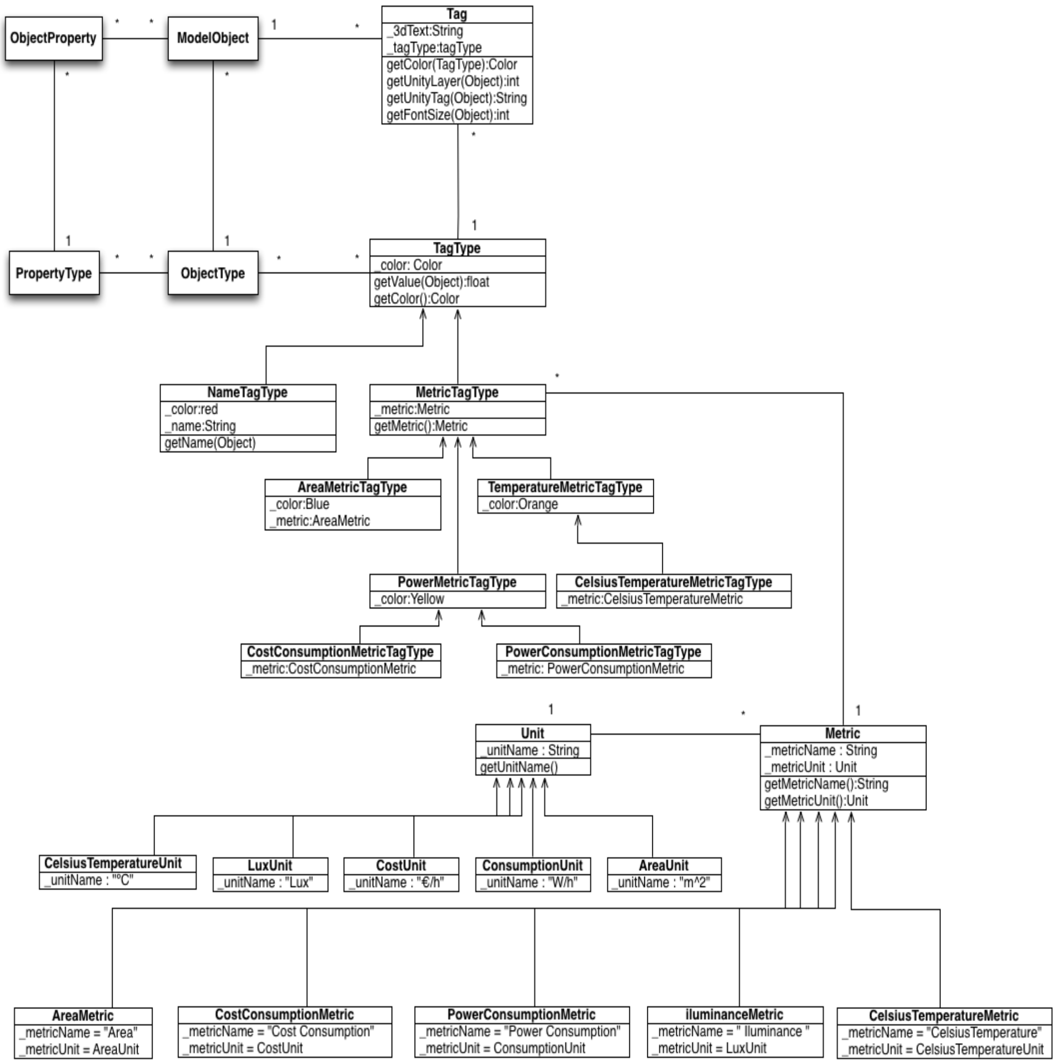


Figure 3.13: Partial diagram of the implemented architecture of Tag domain, it is possible to visualize the concrete forms of the TagType, Metric, and Unit as well as the relation between them

On creation, the room object uses the RoomObjectType to generate its ObjectProperties according with the RoomObjectTypes's PropertyTypes. These ObjectProperties will be used to create the Room Tags and correlate it with the TagTypes requires. For instance, if the NucleoObjectType has a NamePropertyType the Room will have a NameObjectType which meant that the Tag will be of a nameTagType. At the appendix C we present the complete Tag System diagram architecture implemented.

3.3 Discussion

Current Building Automation Interfaces lack in simplicity, they use outdated rendering technologies and can only be used by trained users. With this in mind, we presented a solution based on a user interface featured by a 3D interactive environment. This interface it is to be supported by a modular architecture designed to be easily adapted to existing BAC technologies, improving the user interaction experience and at the same time providing the ability to access and control building information systems in a more efficient manner. This interface was designed using visualization concepts that are common to command and control simulation games, proved effective in this type of interaction. Much like simulation games, BAC Interfaces present great complexity in the number of variables that are to be controlled. Simulation games minimize the visual variables complexity by setting the camera to render only a selected area at each time. Users can freely select which area they want to visualize by selecting it from a mini-map view. This way command and simulation is made through a main view and area selection through the mini-map one.

At the 3D Building Control Interface only one floor at a time is rendered in the main view, the so-called active floor. The user selects which floor should be activated through the mini-map. The selected floor is initially rendered on the main view only with the walls and no devices or sensors shown. The user can then select which categories of sensors and devices should be displayed. Using the navigation commands the user can navigate to the desired space in the building to inspect it. When the view gets closer to a room, additional information is depicted, ensuring that the user will not be overloaded with unnecessary information. The modular architecture supporting our interface, allows each module to manage its functionality independently, making it easy to add, edit or remove modules without affecting the rest of the system.

Chapter 4

Implementation

Specific problems are involved in the process of building a proper 3D interface for Building Automation Control. It is necessary, to select an appropriate rendering engine, create realistic object models, texture them and apply fitting techniques to simulate a proper environment behavior.

In the previous chapter we revealed an interface mock-up and a proper architecture was designed. Now, we will start by describing the technologies involved, programming languages used and the implementation details regarding the main components of our prototype, explaining the choices taken. Later, we end with a small chapter discussion.

4.1 Technologies Involved

In order to make our 3D automated building control tool a reality, it was decided that using a game development engine was the best approach. Following this requirement we study a game engine concerning the main following features: High level programming; 3D support; Deployment on multiple platforms; 3D modeled object importation and Free distribution. Later we decided to use the Unity3D game development engine tool. As known in the gaming community, Unity it has established itself as a reliable tool, used in several commercial titles and that has proven to be an easy to use and fast prototyping tool. Unity 3D is a multi-platform game development tool. The editor runs on Windows and Mac OS X and can produce games for the Windows, Mac, Xbox360, Playstation, Wii and iPhone platforms. It can also produce browser games. A good feature for 3D artists is the integrated support for native formats from Maya, 3Ds Max, Blender and such. When a model is edited, it can be refreshed in Unity, skipping the steps of having to re-export and import it again. For the 3D models we used Autodesk 3DS Max 2009.

The texturing was done using Photoshop CS4 and Pixelmator. Implementation to realtime was done in the game engine Unity 3D. Unity supports scripting via Mono, which is an open-source implementation of the .NET Framework. For scripting, the Unity framework allows programmers to use C#, UnityScript (a custom language with ECMAScript-inspired syntax), or even Boo which has a Python-inspired syntax. Unity also comes whit-in its installation a customized version of MonoDevelop for debugging scripts. In our approach we used the UnityScript, the main reason for that choice was that most unity forum support and examples where scripted using UnityScript. This choice allowed us to easily familiarize with the Unity 3D tool.

4.2 Implementation Details

To better comprehend the implementation details of the prototypes' building process, we start by an overview of how Unity 3D works. After that, we explain how we crated the 3D environment and implemented the user interface. Finally, we present the techniques used in implementation of the Navigation, Illumination, HVAC and Tags components, explaining the choices taken.

4.2.1 Unity

In Unity3D, a game is an hierarchy of Scenes, Game Objects and Components 4.14. Scenes contain the Game Objects of the game. They are usually used to create game menus, individual levels, and everything else. In each Scene you can create environments, objects, and decorations. Essentially they are used for designing and building a game like a puzzle. In Unity3D, every object in a game is a Game Object. However, Game Objects don't do anything on their own. Basically they act like generic containers that represent entities in the game. These can be organized into an hierarchy by putting Game Objects as children of other Game Objects. On the other hand, Components are small sets of data or behavior that can be attached to a Game Object to change its functionality, visibility, appearance, and behavior.

All Game Objects begin with a Transformation Component that defines its position, rotation and scale in the Scene. From there, all other components can be added or grouped as needed for a specific type of entity. For instance, any entity with a 3D representation will need a 3D model to represent it on the scene (a Model Component), a texture to place on the model (a Material Component) and a way to combine both (a Render Component).

In Unity 3D a Component/Script can only be executed when bound to an active scene's game object. With that in mind we defined two types of game objects. The first one is an invisible

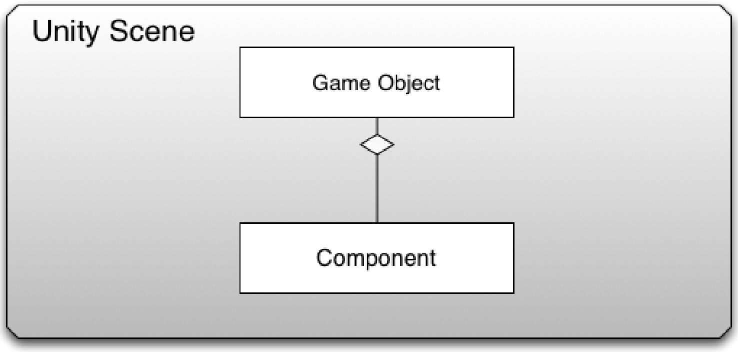


Figure 4.14: Unity 3D hierarchy, where each Scene has one or more Game Objects and each Game Object has one or more Components.

game object, the *"Gestor"*, where we attach all environment management scripts and GUIs. This object is always active which allows its attached scripts to be always running. The second type correspond to all visible game objects, they can be activated and deactivated, usually through manager scripts attached to the *"Gestor"* game object.

4.2.2 Environment Creation

Before creating our virtual environment we had to design all 3D objects and textures. So, we used 3Ds Max to design all tridimensional objects, this program allows not only to create our objects as well as texture them making them appear more realistic. After object creation we exported them using an Unity 3D supported format, the *.FBX* .

At this point, the building object creation was the hardest task. We only had at our dispose bi-dimensional building plants of each floor and no way to automatically extrude them. So, our solution was to draw 3D walls over the entire block E 2D plants and attach a plane to each of the designed building floors. This was not the best solution but it was the fastest option available.

Each of the three floors was defined as a different game object as depicted on Figure 4.15. Later we found that although this was a reasonable choice, if we had created individually each floor's room and hierarchically group them, we could achieve a direct association between each tridimensional room and its logical information, which was not possible in our current implementation.

After floor modeling, we created the remaining 3D objects, the luminaries and air conditioners (HVACs). Some of the luminary models we designed and exported from 3Ds Max to Unity 3D to represent the light sources are depicted at Figure 4.16.

The creation of the User Interface was our next step. To do that we needed to design and edit

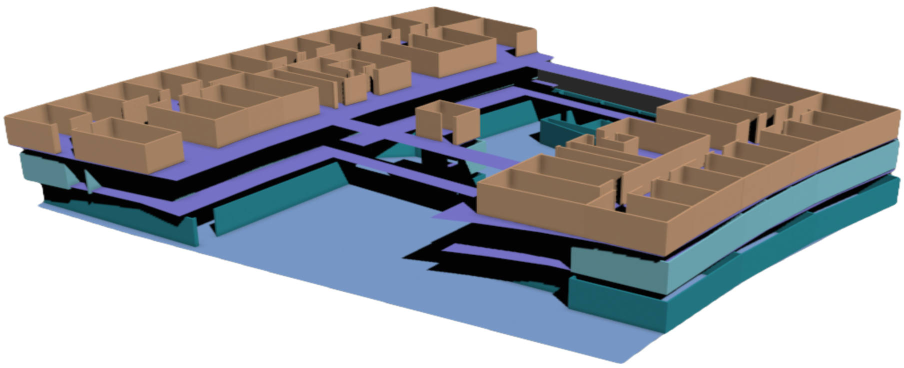
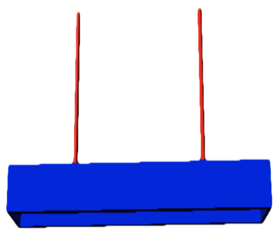


Figure 4.15: IST-Taguspark Block E floors models designed at 3Ds Max



(a) Luminary model type 1

(b) Luminary model type 2

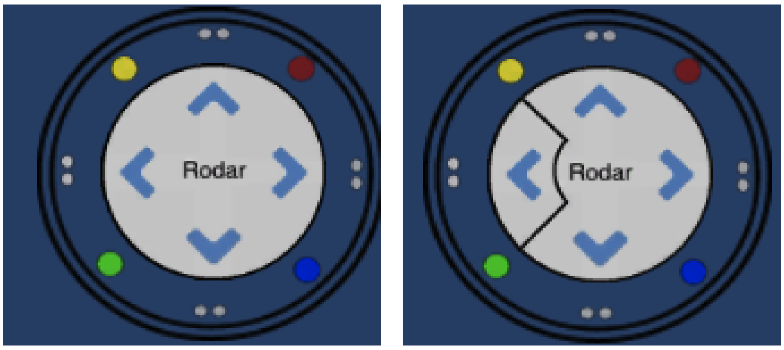
(c) Luminary model type 3

Figure 4.16: All luminaries models designed at 3Ds Max

the GUI textures. This was made using the Photoshop and Pixelmator. Unity has a specific GUI implementation which helped with the creation of most GUI buttons and scroll bars. The main issue was that every button that Unity GUI supplies has a rectangular form, not allowing other shape of buttons. To resolve this issue we define our own GUI buttons. This consisted mainly on applying the created textures on a defined area and changing these textures while the mouse was over or pressed on the desired texture area. Figure 4.17 shows two textures representing a circular shaped button before and after being pressed.

4.2.3 Navigation

Navigation is the vital component in the exploration of the virtual building and it plays a great part in the objective of granting user's freedom of movement. Through an immersive navigation, this system intents to improve the perception of the status of the system and allow the visualization of all building components and devices. Navigation is divided in three smaller components: rotation



(a) Unpressed circular button representation texture (b) Pressed circular button representation texture

Figure 4.17: Textures representing a circular button while unpressed Figure (a) and pressed Figure (b).

panning and zooming all relative to a central object. The rotation component is defined by a vertical and horizontal rotation, where the rotation around the target's local X-Axis is the vertical rotation and the rotation around the target's local Z-Axis is the horizontal one. These rotations are denominated vertical or horizontal due to the movement that the rotation will cause on the camera's positioning. Rotation around target's X-Axis make the camera move vertically where rotation around Z will make the camera move horizontally. Besides being around different axis both rotations also differ in their permitted rotation angles. The vertical rotation is limited to a range from 0º to 90º degrees, this angle limit is ensure that the camera never position itself under the building model. In the horizontal rotation case there is no angle limitation meaning that the camera can rotate around the target as many times as the user wants. Figure 4.18 represents in more detail the pretended axial rotation, where the green area shows the allowed vertical rotation, the red area represents the allowed horizontal rotation and the black rectangle represents the target which the camera rotates around.

This implementation of the rotation component will result in an hemispherical area around an object, where all objects will be visible from several angles. Figure 4.19 depicts the vertical and horizontal boundaries of the camera rotation.

The panning and zooming components work as an extension of the rotation one. As rotation is always centered around a target, panning will allow the camera to move to a left, right, up or down point of the current perspective. This feature will permit to temporarily center camera visualization on another part of the targeted object as desired and later with zoom in or out the perspective over that position, that is over the center of the camera position.

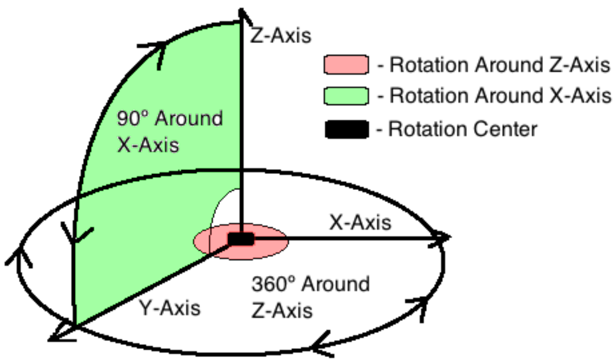


Figure 4.18: Representations of the angular areas of the camera vertical and horizontal rotation. The green area shows the allowed vertical rotation, the red area represents the allowed horizontal rotation and the black rectangle represents the target which the camera rotates around.

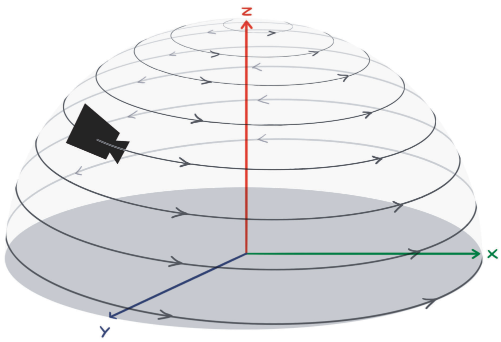


Figure 4.19: Representation of the main Cameras' the permitted rotation area.

4.2.4 Illumination

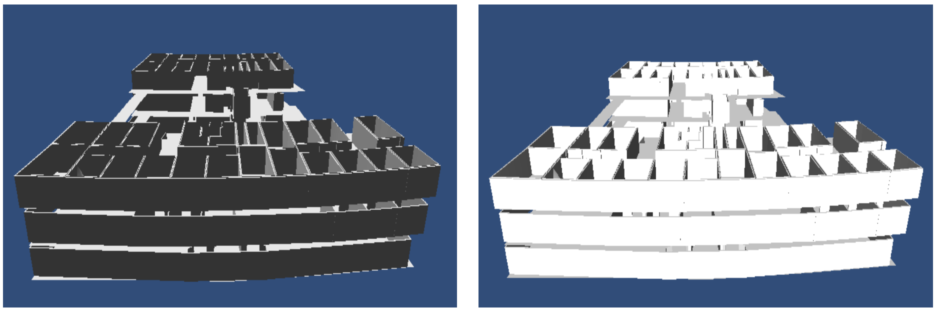
Illumination has an important role on the implementation of any virtual environment. Every Game Engine uses illumination to create a photo realistic simulation effect of an environment. Without it any rendered object would not be visible. In our work there where considered two categories of illumination, Global and Local illumination. Were, Global illumination is refereed to all the lighting that reaches a surface either directly from the light source or indirectly through its interaction with the scene and Local Illumination is the simulation of light reflection at surfaces in a local restricted area. In this section we will introduce the existing Unity 3D illumination types, and explain the techniques we used in the implementation of both Global and Local Illumination types.

Unity Light Types

In Unity, lights are used to illuminate the scenes and objects in order to create the perfect visual mood. They can be used to simulate the sun, burning match lights, flashlights, gun-fire, or even explosions. In our case we need lights to illuminate the scene as well as to simulate Luminaries and their illuminated areas. There are four types of lights in Unity: *(i)* Point lights which are meant to shine from a determined location equally in all directions, like a light bulb. *(ii)* Directional lights, which are used for Global illumination and are placed infinitely far away affecting everything in the scene. *(iii)* Spot lights built to shine from a point in a direction and only illuminate objects within a cone - like the headlights of a car. *(iv)* Area lights, shine in all directions to one side of a rectangular section of a plane. These types of light are only available on the pro version which we did not have access to.

Global Illumination

In our prototype implementation we applied a global light to be able to visualize all the scene objects. For that, we used the directional light that Unity 3D provides us. Although this seemed the obvious choice it was not without its problems. By having a directional light it automatically causes a shadow to appear in the opposite side of an object, in this case our building. So when rotating around the building there were some areas that would not be visible or would be involved in darkness (Figure 4.20(a)). To resolve this issue we attached the Directional Light object to our Camera game object. This makes that the rotation component of our camera to be applied to all their children and so it will be like using a helmet miner light, pointing the directional light always in the camera's viewpoint direction. Figure 4.20 shows both cases. Figure 4.20(a) shows the directional light that is pointed in the opposite direction of the camera, involving all that is not illuminated with a dark color aspect. In the other hand, at Figure 4.20(b) the Directional light is already attached to the camera illuminating everything in the same direction that the camera is pointing at.



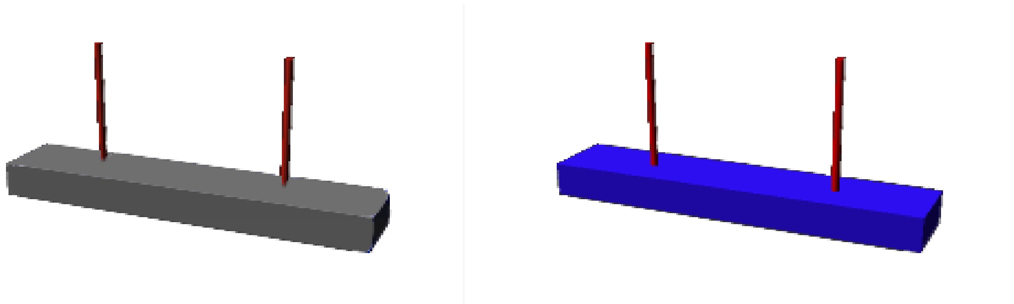
(a) Building illuminated by the Global Light when **unattached** to the camera component.

(b) Building illuminated by the Global Light when **attached** to the camera component.

Figure 4.20: Building illuminated by the Global Light, unattached (a) to the camera and (b) attached to the camera. In (a) the light is static an it is located at the opposite direction of the camera, at (b) due to being attached to the camera the directional light is pointed at the same direction of the camera

Luminaries

As mentioned before, to create the luminary objects we used 3Ds Max resulting in the models depicted at Figure 4.16. These models were meant to virtually represent our building luminaries physical aspect and alone were not able to represent their actions on the environment or their status by their aspect.

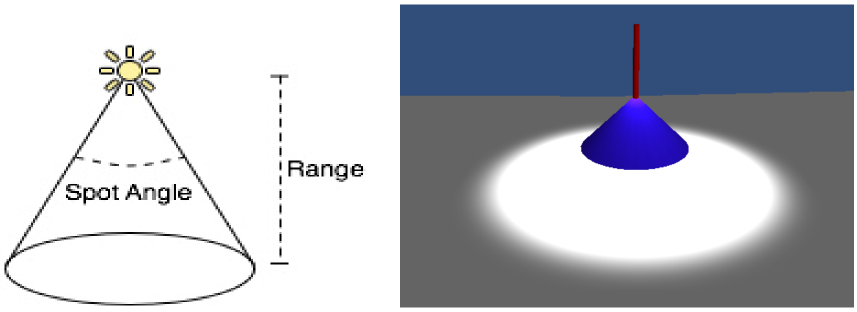


(a) Luminary aspect when the illumination is Off.

(b) Luminary aspect when the illumination is On.

Figure 4.21: Luminary aspect when (a) its light is turned off and (b) when its light is turned on.

The aspect part was resolved by associating a component script to all luminary types that would change the luminary color according with the light state. Making the luminary have a grey aspect when its light was turned off and a colorful one when on. Figure 4.21 shows an example of both turned on and off cases. To represent a luminary action on its environment we tried two illuminations techniques: the first one was to use Unity integrated spot light. As we described before, spot lights only shine in one direction, in a cone, which would perfectly apply to our illumination needs because we would be simulating the real effect of a regular chandelier lamp, as depicted at Figure 4.22.



(a) Spot light scheme. (b) Actual spot light implemented at our prototype.

Figure 4.22: Scheme representation of a spotlight at (a) and its actual application at our prototype (b).

One of the problems that we found with this type of illuminations is that spot lights are the most expensive on the graphics processor, so in a slower computer, having dozens of lights would make the computer really slow. But this was not their main issue. The problem is that our Unity version does not support shadows(only available ate Unity Pro), this causes all types of light to illuminate through walls. For example, if we had a luminary set inside a room its light would illuminate the room floors and walls but would also illuminate some parts of the adjacent rooms. Figure 4.23 shows how a room with this kind of illumination would look like. As we can see the room interior is illuminated in the same way that a real chandelier would but the outside of the room is also being illuminated when it should not.

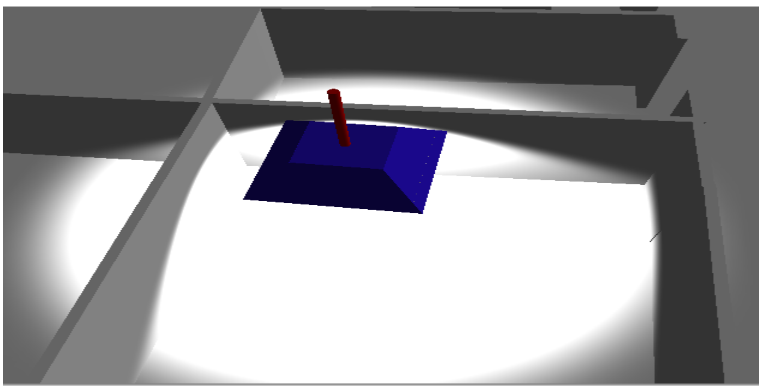
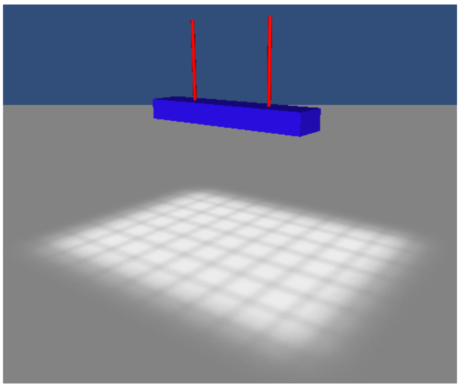
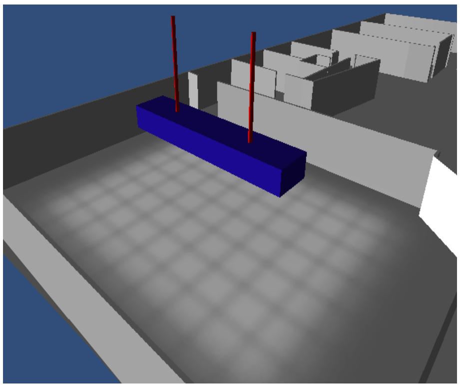


Figure 4.23: Luminary using Unity spotlight, where the room interior is illuminated in the same way that a real chandelier would but the outside of the room is also being illuminated when it should not.

Given the tools we had, we tried a different technique. We knew we had to only simulate individual rooms illumination, so instead of make use of Unity lights we used Unity Projectors. A Projector allows you to project a Material onto all objects that intersect its frustum. It is commonly used to implement projected shadows. In our case we used it to render a white grid texture directly bellow our luminary model, thus creating our desired illumination effect. With this solution, we resolved both the light performance issue as well as allowed us to position the projection light only inside a room without the risk of also illuminating the adjacent ones.



(a) Initial Light projection in a scenery without walls.



(b) Final Light projection in the actual building scenery

Figure 4.24: Light projection at initial testing (a) and already implemented at our building scenery (b) .

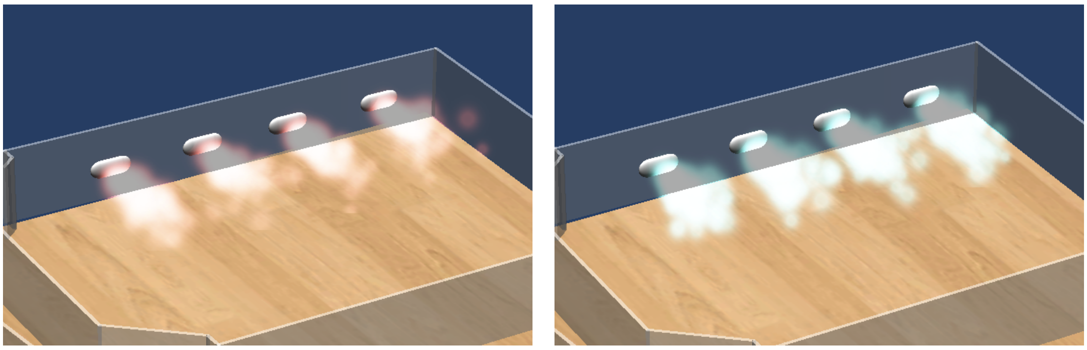
As we can see by looking at Figure 4.24, this projection technique presents a smooth environmental illumination. Also its shape as well as color can be easily changed by switching the projector texture at unity editor.

4.2.5 HVAC

Air movement is invisible and hard to visually represent. We usually know that our HVAC is turned on by hearing its work noise or feeling its air flow. When studying this problem we verified that we can not represent its action on the environment in a familiar way. As we know heat and cold are commonly associated to red and blue colors, so our solutions had to use this in some way. Although not familiar, television weather forecast charts are the most common way to visually represent hot and cold air flows. So in our first approach we thought of using Unity Projectors to show animated heating charts projected on the floor. With these projections we would be able to not only visualize each individual room temperature as well as the air movement direction emanating from our HVAC 3D model. The main problem found with this approach, was that this projected charts would ultimately conflict with the illumination ones. If both the illumination view as well as the HVAC one were selected at the same time each projection would be overlapped and generate a confusing image.

Another possible approach was the usage of colored volumetric fog. This hypothesis was easily deprecated because it would mess up with all illumination effects that we created.

At last our solution was made through the usage of colored particle systems. They are commonly used to make clouds of smoke, steam, fire and other atmospheric effects, which made them perfect for simulating a red or blue smoke coming out of an HVAC. Particles are essentially 2D images rendered in 3D space. A Particle System is made up of three separate Components: Particle Emitter, Particle Animator, and a Particle Renderer. The Particle Animator moves particles in different directions and change colors. In our case we simply draw the particles leaving from the front of our HVAC model, making them disperse along their lifetime until they get recycled. As the HVAC temperature was set to cooler temperatures, the particles that were recycled would come out with a darker blue color and when set do warmer temperatures they would change to a darker red one. Figure 4.25 shows both visual results.



(a) HVAC set to a warm temperature   (b) HVAC set to a cold temperature

Figure 4.25: HVAC implemented using a dynamic particle system, where at (a) the HVAC is set to a warm temperature and at (b) is set to a cold temperature.

This particle system is easily changed through the Unity editor, where we can increase particle speed and lifetime along others particle tunings. In the future we should implement a better particle shading that increases the hot/cold particle contrast.

4.2.6 Tags

The tag system was meant to supply our prototype with a simple way to visualize space related information. Through Tags, one can simply select the intended functionality filter and observe in real time the whole floor desired information or just zoom in a specific space, such as an individual room, and visualize all the informational Tags belonging to that room. In case of existing aggregate logical spaces, such as a *Nucleo*, the *Nucleo* information Tags would supply the aggregate value of all its physical spaces. For instance if a *Nucleo* encompasses room one to three, its Electrical Consumption information Tag would be the sum of all its rooms electrical consumption.

Our approach in this matter was to observe a worldwide used tool, google earth, and transpose its geo-referencing information method. Put in a simple way, this tool simply places a billboard over a location to display Information. In our case we adopted a similar technique. On loading time we create every Tag and placed it on its belonging space. Each Tag consists in 3DText Mesh with an rectangular shaped billboard attached. Each billboard presents a certain level of translucidity, which allows us to see behind the Tag object as if it was a window glass. Each Tag has a color regarding its functionality, for instance Electrical consumption Tags are yellow and the location name ones are in a white tone. Figure 4.26 shows an example of some created Tags, where the white ones refer to information regarding the name of a location and the yellow one to the electrical consumption of it.

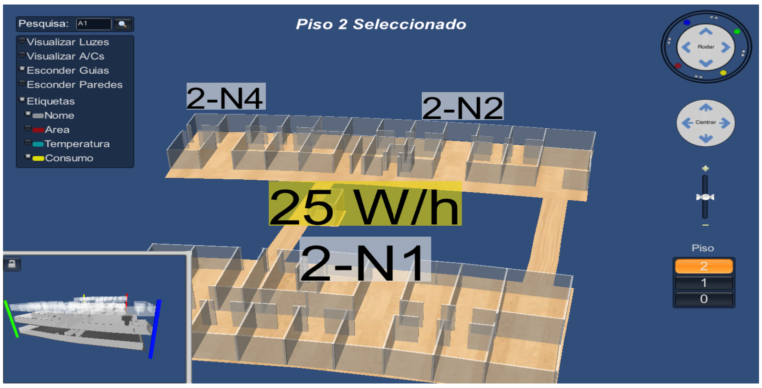


Figure 4.26: Some of the created Tags where the white ones refer to information regarding the name of a location and the yellow one to the electrical consumption.

Explaining in more detail, on load time the 3DText mesh is generated according with the information obtained from the domain. After that, the billboard is created and its size and position are calculated according with the bounding box of the 3DText mesh. Its color is attributed according with the type of information provided by the domain. Figure 4.27 shows the created object hierarchy.

When a space has more than one associated Tag, only the main one remains active and it is rendered. This was meant to improve system performance while rendering. When a user wants to visualize all tags of a given space, he clicks on that space main tag and an event triggers active all of that space existing Tags, positioning them above each other like a Tag pile. Reordering all Tags in an individual location is also possible, after triggering all Tags to active in that space one

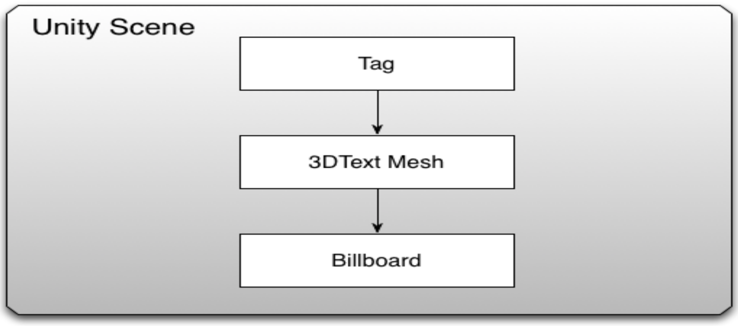


Figure 4.27: Tag object hierarchy within a scene, where the Tag is composed by two more game objects, the 3DText Mesh game object and the billboard one.

can simply select which one of them he wants to become the main one. Figure 4.28 illustrates the sequence of events described. In this sequence we are not considering the menu events such us the the ones form the GUI interface Tag types menu selection.

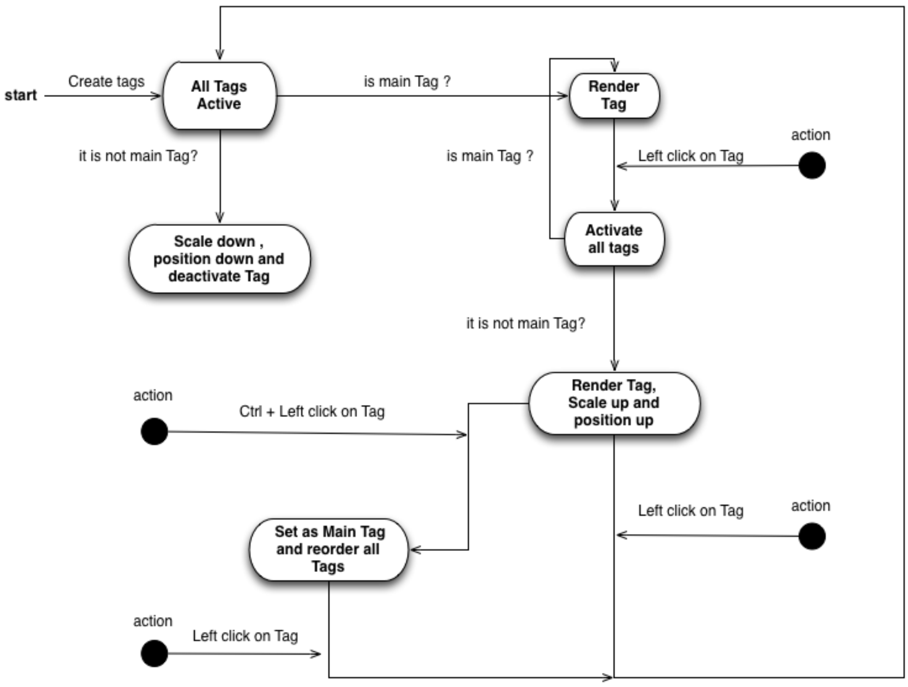


Figure 4.28: Tag events sequence diagram

There are some aspects of the Tag system implementation that should be modified. For instance when a space as several Tag types (about more than 5) it should not be possible to visualize them all in the Tag pile. Without this limitation, the system would be open to such a great pile of Tags that it would not be possible to visualize them all in our camera's field of view. In these cases, the visibleTags customization could be achieved through the GUI interface Tag types menu selection.

4.3 Discussion

In this Chapter, we started by summarizing the technologies involving the implementation details of the FM3D prototype. During the development process, we had to choose between different techniques to resolve some of the problems we encountered. Both, the implemented techniques as well as some of the deprecated ones, were justified taking in consideration why we chose them, by reporting our line of thought. At the end of each section, we tried to advance some considerations about what we thought it should be improved about each section subject and how to.

In Building Automation systems, there is a large amount of available information regarding each space. If all available information was rendered at the same time, it would make our 3D environment very confusing and hard to interact with. To prevent that, It should be possible to select the type of information that we want to be visible at a time.

In our implementation we represented each of the formal areas of interaction through different visualization techniques, which if rendered at the same time will not interfere with each other For space illumination simulation, we used texture light projections on the space floor, for HVAC space action we used particle systems representing the hot/cold air flow that comes from the HVAC device and when it was the case of visualizing other types of spatial information, we used information billboards combined with filters allowing the user to select and almost instantaneously become visible the information type he wants to. Through the filtering option we can choose what information we want to be visible at a time thus limiting the visible information to a manageable number.

As for the Usability requirements, we will now evaluate them in the next Chapter, with the report of our user evaluation results.

Chapter 5

Evaluation

Our evaluation process consists of a comparative study that contrasts our 3D interface applied to the centralized control of a building automation system with a corresponding legacy application. This main goal of the evaluation is to investigate the reliability and possible benefits of 3D virtual environments for Automated Building by performing a quantitative as well as a qualitative analysis on both systems through user's interaction test sessions. This methodology and the results of this evaluation are described in the following sections.

5.1 Methodology

To evaluate our FM3D prototype application, we executed a comparison between our prototype and an existing legacy application for centralized control and monitoring, featuring a traditional 2D Window-Icon-Menu-Pointer (WIMP) interface. With this intent, we used the existing Schneider Electrics TAC Vista application interface depicted in Figure 5.29 already installed and working at the IST-Taguspark main building. This comparison proceeded along two testing stages, the early prototype stage and the final prototype stage. With the early prototype stage we intended to get a first perspective of how users would react to our 3D interface. At this time all main functionalities were already implemented so the feedback gathered from this phase did not only contribute to infer possible adjustments to our final prototype as well as allow us to obtain a good preliminary quantitative and qualitative analysis of our prototype's main functionalities.

In both stages our evaluation is structured by the following steps: a pre-test questionnaire to establish user profile; a briefing about test purposes and several tasks, preceded by a short training where users freely explored each application for three minutes; ending with a questionnaire after

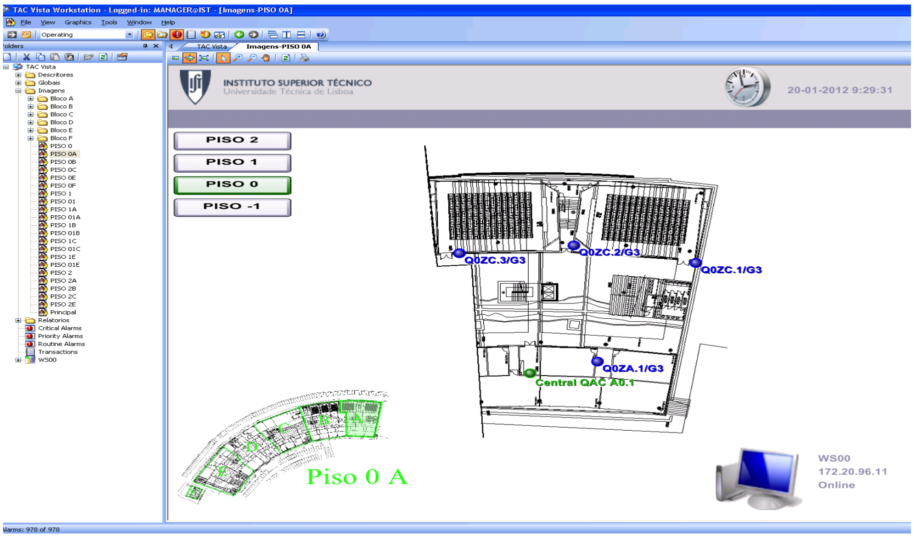


Figure 5.29: A screenshot of Schneider's TAC Vista user interface.

completing the tasks in each application. This was meant to ensure an even test distribution of the applications. It should be mentioned that at the second phase it was added two more tasks to be tested only on our prototype. With this new task we intended to evaluate some secondary functionalities not currently available on the legacy application.

During task execution we measured the time that each user took to complete each task on each application. If a task was not completed after three minutes, the task was considered incomplete. From these data we were able to perform a quantitative comparison between the two applications. Additionally, the complete test session was filmed for posterior in-depth analysis, allowing the identification of details that might have been missed during the evaluation.

With the post-test questionnaire, we were able to elaborate a qualitative analysis. It contains direct questions related to the use experience, with special emphasis in the difficulties users faced during task execution.

5.2 Participants

Involving real users is a major aspect of our evaluation. Having this in mind, we preferred to have a small number of participants, instead involving ad-hoc users in the tests, which could produce biased results. Therefore, the evaluation was carried out by a total of 7 users and divided in two evaluation phases. The early prototype stage and the final prototype stage.

- At the **early prototype phase** only two participants were involved. These participants (Users A and B) belong to the security service operating in the building, have 62 and 69 years old, basic education levels, low proficiency in the usage of software applications and have no previous experience in using building automation systems. Indeed, this was the first time both used such a system, but they have deep knowledge of the building and are used to interact directly with the physical devices and sensors.

  This software could, in principle, assist personnel with minimal training in controlling the facilities from any location in the building using a web-based 3D interface carrying out certain maintenance tasks. The choice for these participants involved in the first evaluation stage was made with the belief that these were good representatives of the target user group of FM3D.

- At the **final prototype phase** a total o five participants were involved. They were composed by three building security service elements and two administration elements. For these tests one of the participants was used for test control purposes, he was an advanced user with high knowledge the building.

  The advanced participant is a 31 year old male with secondary education level, high proficiency in the usage of software applications and high experience in using our legacy application at our building. As depicted in Table 5.5 the advanced participant will be identified as UserG. The other participants were three males and a female, with an age spanning from 34 to 56 years old, secondary education levels, some proficiency in the usage of software applications and no previous experience in using building automation systems.

Tables 5.5, 5.6 and 5.7 give us a more precise view of all participants profile such as sex, age, academic education and informatics applications usage frequency among others. This was possible through the evaluation of the pre-test user questionnaires. At Table 5.5 it is defined the User Id that will be used identify each user through the evaluation Chapter.

Stages	User			
	User ID	Age	Sex	Academic Qualifications
Early Eval. stage	User A	62	Male	11º Grade
	User B	69	Male	4º Grade
Final Eval Stage	User C	39	Male	9º Grade
	User D	34	Female	12º Grade
	User E	54	Male	11º Grade
	User F	56	Male	11º Grade
	User G	31	Male	12º Grade

Table 5.5: User's basic profile depicting the user's evaluation Id, age, sex and academic qualifications according to the evaluation stage they where in.

Through the observations of Table 5.6 we were certain that every participant was familiar with computer applications and had used, at least a few times, computer and cellphone interfaces. Mainly 2D ones.

Stages	User ID	Users Informatics Application Usage	Informatical Interfaces Interaction Acquaintance			Informatical Interfaces Interaction Acquaintance	
		Frequency	Cellphone	Computer/ Laptop	Touch Screen	2D	3D
Early Eval. stage	User A	Few Times	yes	yes	no	yes	no
	User B	Some Times	yes	yes	no	yes	no
Final Eval Stage	User C	Frequently	yes	yes	no	yes	no
	User D	Some Times	yes	yes	yes	yes	yes
	User E	Frequently	yes	yes	no	yes	no
	User F	Few Times	yes	yes	yes	yes	no
	User G	Frequently	yes	yes	yes	yes	yes

Table 5.6: User's basic profile according with the evaluation stage they where part of. (Part I)

Independently of the evaluation stage, this was the first time that every participant had contact with a 3D BA System, this should be useful to determine the participant's learning curve, intrinsic to 3D applications such as ours.

Stages	User ID	Building Control Interfaces Usage
Early Eval. stage	User A	none
	User B	none
Final Eval Stage	User C	none
	User D	none
	User E	none
	User F	2D
	User G	2D

Table 5.7: User's basic profile according with the evaluation stage they where part of. (Part II)

Our tests were performed at the facilities of IST - Technical University of Lisbon, Taguspark campus. All participants agreed to let themselves be filmed for our testing sessions.

5.3 Tasks

As introduced above, the test session was comprised of two applications: TAC Vista and FM3D, both running on a desktop computer with the traditional mouse, keyboard and screen setup, as depicted in Figure 5.30. For each application the participant was required to execute several tasks. Before the beginning of each task, the system was set to an initial state, to ensure that all tasks were executed from the same starting conditions.

Figure 5.30: Picture of the test participant executing a task in TAC Vista during test session.

A member of the development team was always present during the entire duration of the evaluation and acted as an observer not interfering with task execution. This person was responsible for reseting the system state. All tasks were presented to the users in written form at the beginning of the task.

Due to limitations on the features available on the legacy application and on the availability of the test users, it was not feasible to perform in-depth tests to a wide variety of functionalities. Instead, at early prototype stage, we focused only on two specific contexts: HVAC and illumination. From previous interviews with the actual responsible for the facility, we had identified four simple tasks common to both tools that were selected as representative of everyday activities. These tasks, were executed, at both stages by all participants during the tests.

During previous interviews with several building automation users, it was identified that the legacy application lacked to offer proper spatial information monitoring functionalities such as: area electrical power consumption, electrical cost consumption or temperature monitoring. So at the final prototype stage the test Spatial information monitoring context was added. As mentioned before functionalities where not available at the legacy application so we decided to add two more task to be tested only at our prototype. All tasks, executed by the participants during the tests are discriminated at Table 5.8 according to their evaluation context.

Evaluation Context	Task Id	Tasks
HVAC and illumination	T1	Check illumination status of room A1
	T2	Verify illumination of area in front of room A2
	T3	Confirm that HVAC of room A2 is On and set to Heat
	T4	Set HVAC of room A2 to 22ºC
Spatial Information monitoring	T5	Verify through the use of Labels, whats the current electrical consumption of the Nucleo-2-N2 area.
	T6	Verify through the use of Labels, whats the current Temperature of the A1 Anfitheater.

Table 5.8: Tested task description according with their evaluation context.

Observing Table 5.9, the Spatial information Monitoring tasks, *T5 - Verify through the use of Labels, what is the current electrical consumption of the Nucleo-2-N2 area* and *T6 - Verify through the use of Labels, what is the current Temperature of the A1 Amphitheater* were only evaluated on the FM3D application at the final evaluation stage.

	Early Evaluation Stage		Final Evaluation Stage	
Task Id	TacVista	FM3D	TacVista	FM3D
T1	X	X	X	X
T2	X	X	X	X
T3	X	X	X	X
T4	X	X	X	X
T5				X
T6				X

Table 5.9: Tested task according with their evaluation stage and application, where TacVista is the legacy application and FM3D our prototype. The X indicates that a given Task Tn was evaluated on the corresponding application.

5.4 Test Results

We present two different perspectives on the analysis of the results from our user study. First, we present a quantitative analysis drawn from task completion times in each application by each user. Second, we discuss a qualitative analysis based on the questionnaires and several observations captured throughout the test sessions.

5.4.1 Qualitative analysis

After each application testing, all subjects filled out a post-test questionnaire. This questionnaire consisted in a group of twelve questions selected from the ISO 9241 standard from the International Organization for Standardization, covering ergonomics of human-computer interaction. These questions, depicted at Table 5.10, were selected taking in consideration five fundamental aspects for a good interface construction: Design; Functionality; Easiness of use; Learning and Satisfaction. Table 5.10 Question ID will be used to identify each question through the rest of the evaluation Chapter

	Question ID	Question
Design	Question 1	I liked using de application interface.
	Question 2	I find the application interface pleasing to use.
Functionality	Question 3	The application as every functionality that I expected
	Question 4	The system's available information is helpfull to complete all tasks
Easy to Use	Question 5	I find the application is easy to use
	Question 6	It is easy to find the needed information.
	Question 7	Globally I find the application easy to use
Learning	Question 8	It is easy to learn how to work with the application
	Question 9	The information provided by the application is easy to learn and understand
Satisfaction	Question 10	I felt confortable working with the application
	Question 11	I generally can complete all task in an effective manner
	Question 12	I am globally pleased with my tasks performance

Table 5.10: Questions used for the qualitative analyses software comparison divided according with their context purpose.

All questions, were designed so that the collected answers could be treated using an Likert scale with five admissible values, from 1-"Totally Disagree"; 3-"Not agree nor disagree" to 5-"Totally Agree" , where each participant could state their opinion of both applications in a comparative manner. The collected data was analyzed using 2-D Column charts to determine if significant differences existed between both versions. During the presentation of the results, we have a chart for each group of questions that were defined at Table 5.10. We will also use the User ID from Table 5.5 to identify each user's opinion on the FM3D and TacVista applications. Also, the questions posed to the users were both positive and negative to reduce the impact of users with no idea biasing the survey, since some users could resign themselves to a positive response and some could resign themselves to a negative one.

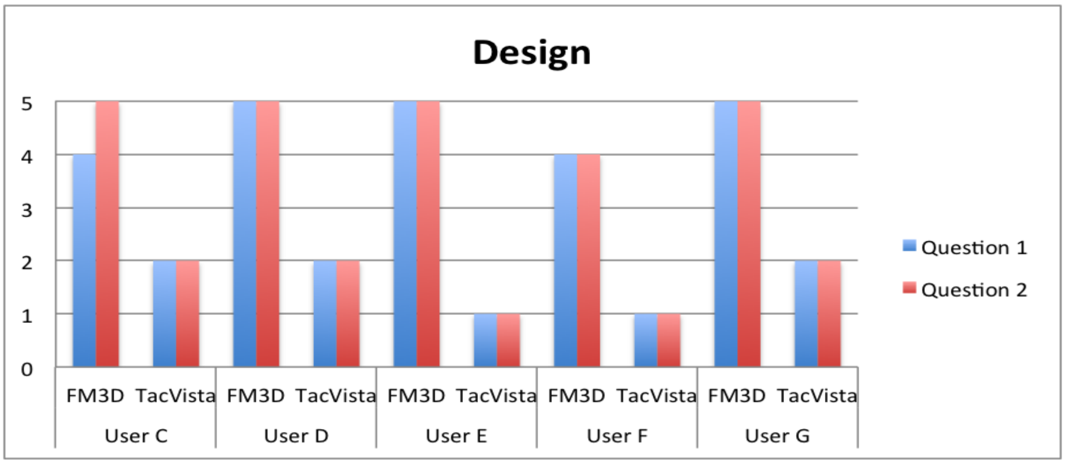


Figure 5.31: Results obtained from the *Design* context purpose questions (Questions one and two) for each user and application.

Questions one and two were meant to evaluate and compare the Design component of both application's interface, in particular, which application is more pleasing to use. Figure 5.31 presents the results obtained through these questions. As it can be seen, most participants gave the highest possible score of five to the FM3D design regarding pleasingness of interaction and likability about the interface design, where by comparison the best results that the legacy application obtained were a low score level of two. This demonstrates that despite most participants were not familiar with 3D applications they found that 3D interfaces to be more pleasing pointing out the FM3D interaction experience as a more enjoyable than of the legacy application.

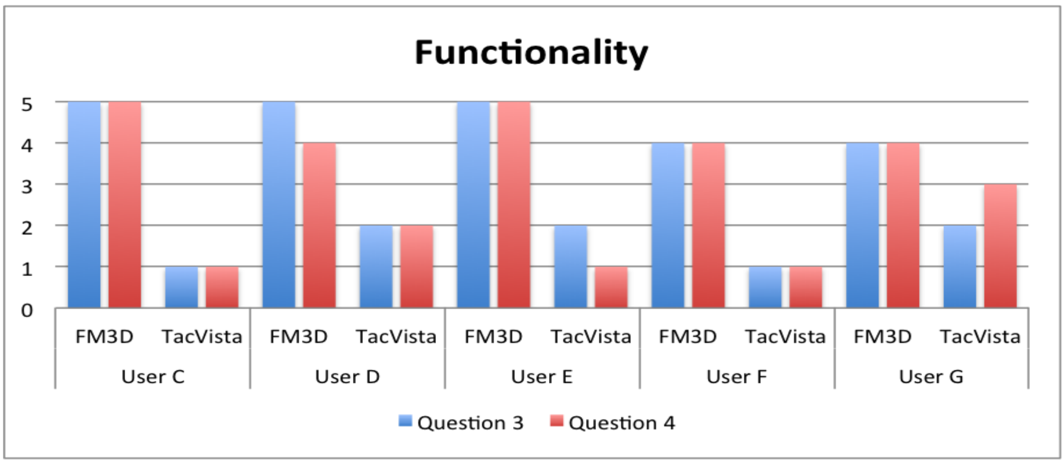


Figure 5.32: Results obtained from the *Functionality* context purpose questions (Questions three and four) for each user and application.

Questions three and four were meant to evaluate and compare the Functionality component of each application's interface, in particular, which application presented a more helpful interaction information while trying to perform and complete the test tasks. Figure 5.32 presents the results obtained through these questions. Once again, all participants gave a high score of four and five to the FM3D application against a low score, of one to two, to the functionality component of the legacy application. The main reason evoked by the participants for such low scores on the legacy application was that the information that the legacy application presented was too confusing and difficult to spatially relate to a physical area within the building. In the other hand, due to the 3D building representation and navigation presented by the FM3D application it was easy to understand the spatial relation between the information and its corresponding area. In fact, most users mentioned that the reason why they did not give the maximum score to the FM3D was that some menus should be bigger and easier to read.

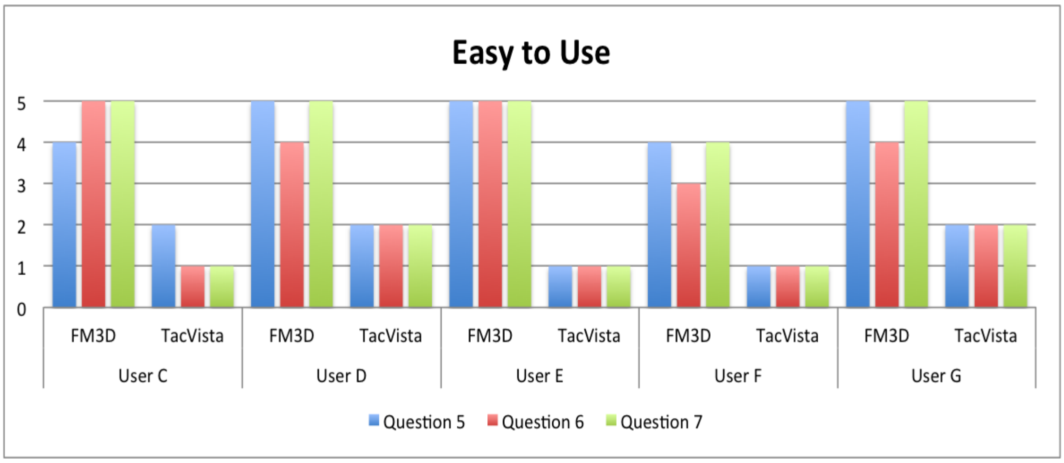


Figure 5.33: Results obtained through the *Easiness to Use* context purpose questions (Questions five six and seven) for each user and application.

The easiness of use of each application's interface was evaluated through questions five, six and seven. These questions are meant to evaluate how easy it is to visualize the desired information, navigate and interact with each application. Figure 5.33 presents the results obtained through these questions. Globally the FM3D results were good and every participant's score to FM3D was higher than the ones given to the legacy application, but in comparison with previous usability components results, the easy to use component of our application should be improved. Question six *"It is easy to find the needed information"* was the one with lower scores and the one that should be improved. The reason users evoked for Question six slightly lower score was that, in the FM3D application, by design, some information is only visible according with the current navigation zoom level. For instance, if a participant is viewing the entire building the system only provides information of big aggregate areas like a *Nucleo*, to see information regarding a room in a *Nucleo*, such as the room's temperature, the participant has to zoom in on that *Nucleo* and only then the system will show the *Nucleo's* rooms temperature. This is one of the usability issues that certainly will have to be studied and improved for future work.

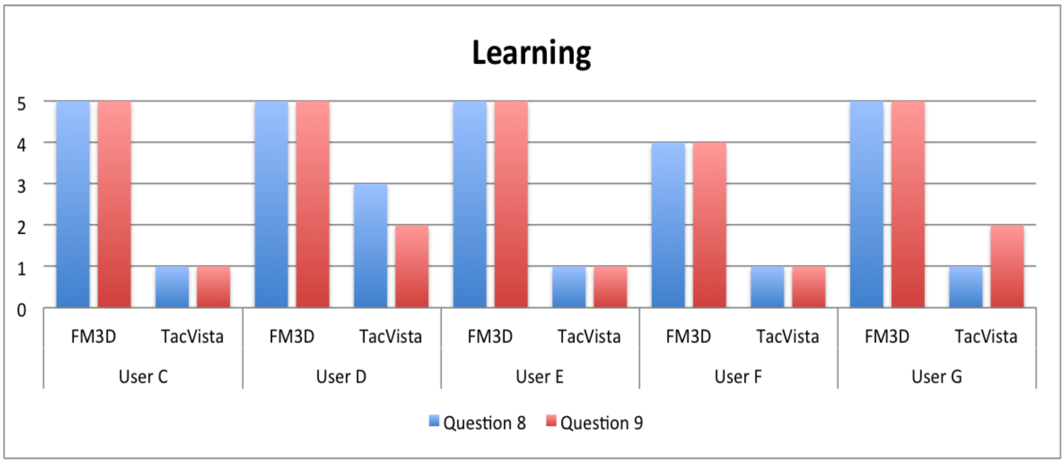


Figure 5.34: Results obtained through the *Learning* context purpose questions (Questions eight and nine) for each user and application.

Questions eight and nine were meant to evaluate and compare the Learning component of each application interface, in particular, which application is easier to learn, that encompasses functions connected to navigation, command and information retrieval. Figure 5.34 presents the results obtained through these questions. Once again FM3D obtained great results, all participants though that the application was really easy to learn and almost with no need of "before use" training. This can be translated as a great advantage because users would not need to spend time and money on expensive training courses for an effective system manipulation.

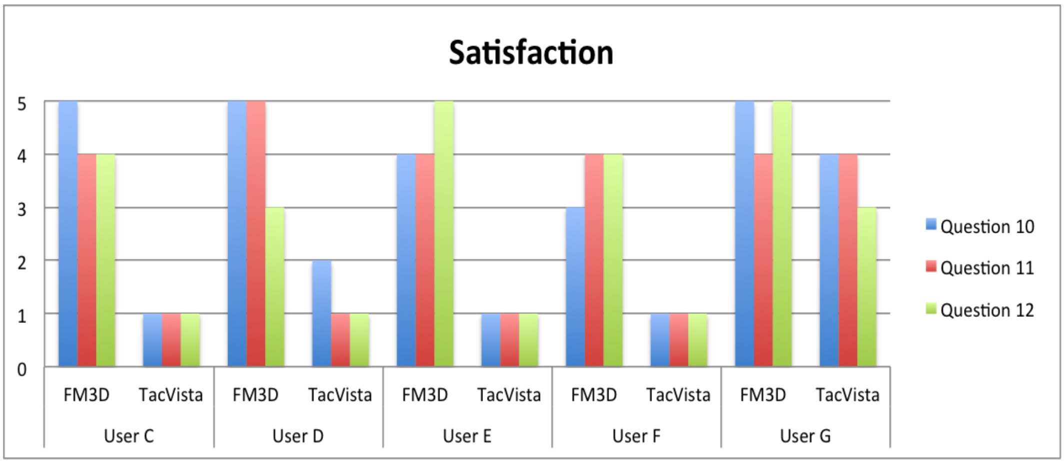


Figure 5.35: Results obtained through the *Satisfaction* context purpose questions (Questions ten eleven and twelve) for each user and application.

The Satisfaction component was evaluated through question ten, eleven and twelve. With these questions we intended to determine how comforTable each participants where on interacting with both applications and their satisfaction towards their own performance while using them to complete the provided tasks. This way it should be possible to actually observe if the participants find the application useful to their performance and needs. Once again the legacy application's results where bad. Almost all users gave a score of two in all satisfaction questions. In fact only the advanced participant displayed a average/good opinion on the satisfaction context questions of the legacy application. This discrepancy on the advanced user results by comparison with all others will be discussed in a deeper manner in the Advanced Participant section. Figure 5.35 presents the results obtained with the Satisfaction component questions.

5.4.2 Quantitative analysis

As mentioned before, at section 5.1, during task execution the time that each user took to complete each task on both applications was measured, moreover if a task was not completed after three minutes, the task was considered incomplete. Moreover, due to the reduced number of participant no statistical analysis was made. Instead, our quantitative analysis will be described through the discussion of the obtained raw data.

Given these premisses, we defined a time scale ranging from zero to three minutes which was applied to every task result chart, so any unfinished task will be represented as taken exactly three minutes to complete.

Along this section we will discuss the chart Figures corresponding to each task and its evaluation context, as referred on Table 5.8 of Section 5.3 Tasks. The discussion results will be divided in two parts referring to both task's evaluation contexts, namely HVAC, Illumination and Spatial Information Monitoring.

HVAC and Illumination

The HVAC and illumination evaluation context test, depicted at Table 5.8, consists of four tasks. Each task result will be presented and discussed individually to aid in the interpretation of what happened. Since our participants performed each task on both applications, their results will be displayed as two side by side columns to better visualize and compare each task time completion value on each application. In the end, we intend to verify if a 3D interface can actually improve user performance in BAS activities.

From Figure 5.36, we can verify that all users were more efficient with FM3D, since all users

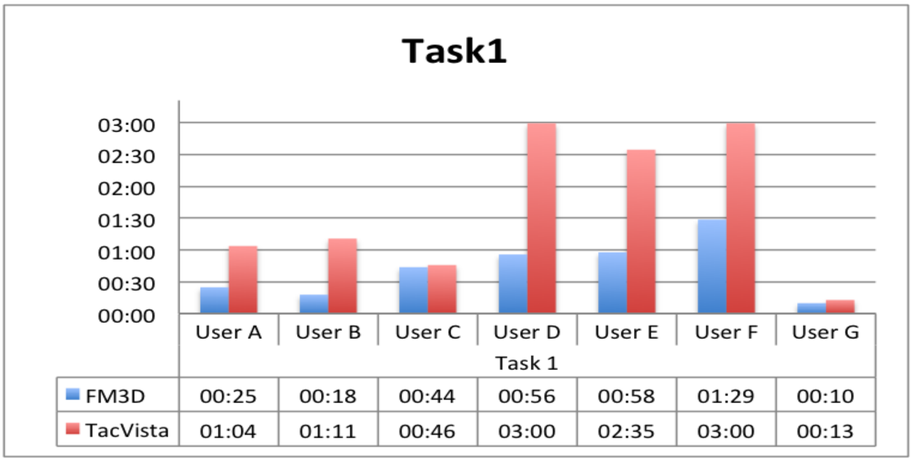


Figure 5.36: Time chart displaying the time taken by each user to perform task one on each application

performed every task faster in our prototype than in TAC Vista. Indeed, on most cases the execution time of the Task 1 in FM3D is less than half of the time required to perform the same task in the traditional application. It is also noticeable that User D and F were not able perform task one at TacVista, where they achieved same task on our application with few difficulties in less than half of the max time possible.

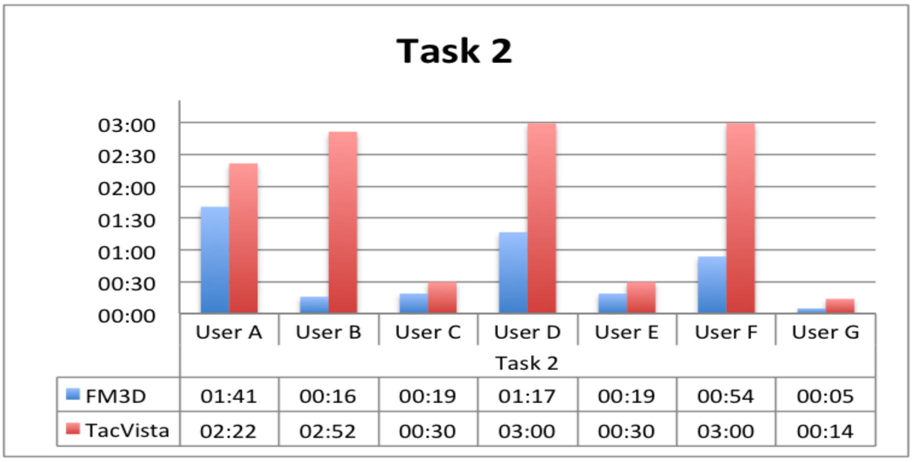


Figure 5.37: Time chart displaying the time taken by each user to perform task two on each application.

Looking at the measurements displayed at Figure 5.37, task 2 turned out harder to complete, that is, when the used application was the traditional one. In FM3D, the time taken for task completion remained low, and only User A presented a completion time above one minute and thirty seconds. Also, in this situation both User D and F when using TacVista, still were not able to complete their given task. By talking with the participants they praised the fact that with our prototype they were able to just look and see the illuminated areas only needing to find the correct room. With the legacy application, that was not possible, since the participant had to know where the room was located and infer if the lights were turned on.

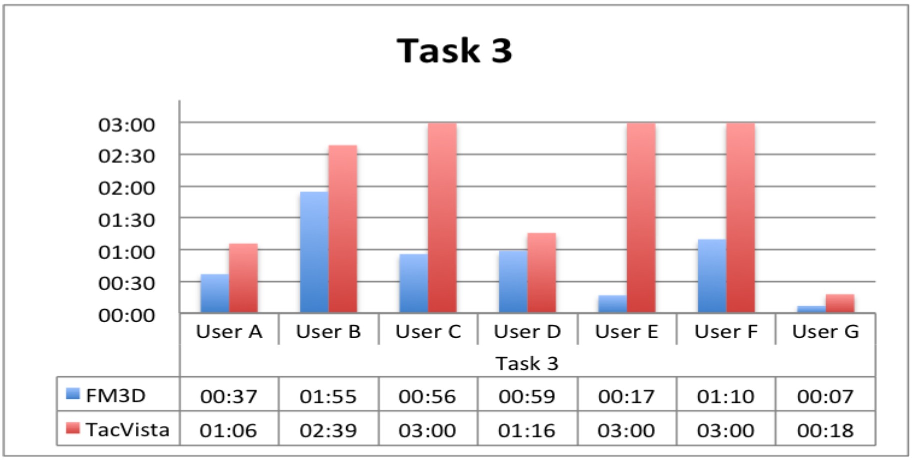
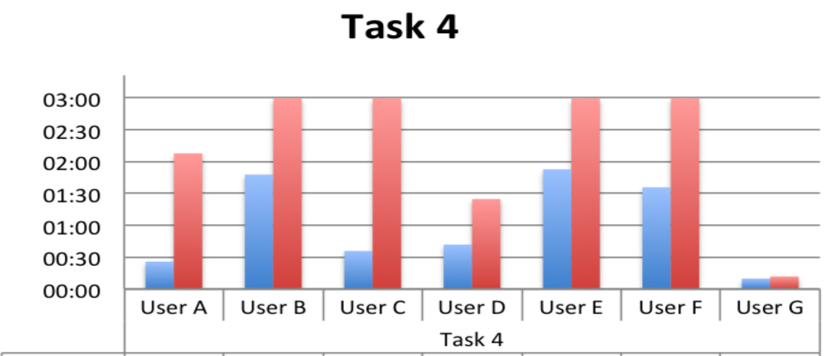


Figure 5.38: Time chart displaying the time taken by each user to perform task three on each application.

The TacVista's results became worse on Task 3, the number of users that were not able to complete the task increased from two to three. In fact User C and F thought that TacVista was so frustrating that they given up before the tree minutes passed. Their main difficulty was that they could not find the HVAC location menu required for the fulfillment of this task. In fact, there was no direct correlation with the area which the HVAC acted upon and its menu. On the other hand, at the FM3D the HVAC's were easy to find and users where able to directly visualize their state upon the environment which facilitated all users to complete task 3 well in time. These results are presented at Figure 5.38

At the final HVAC and illumination evaluation context task, depicted at Figure 5.39, the time differences between both applications became much more expressive, approximately sixty percent of the participants were not able to finish the task while using TacVista. The mains reasons stated by the participants for this results were the same as before, there was no direct correlation with the area which the HVAC acted upon its menu so they could not find the HVAC's menu. Even User B, that was able to previously find the HVAC's menu location and fulfill Task 3, found the menu so confusing that couldn't change the HVAC's temperature set point and finish Task 4. Again our prototype presented satisfactory results with a one hundred percent task completion.



	User A	User B	User C	User D	User E	User F	User G
FM3D	00:26	01:48	00:36	00:42	01:53	01:36	00:10
TacVista	02:08	03:00	03:00	01:25	03:00	03:00	00:12

Figure 5.39: Time chart displaying the time taken by each user to perform task four on each application.

Spatial Information Monitoring

The Spatial Information Monitoring evaluation context test, depicted at Table 5.8, is composed of two tasks. These tasks were made to test a feature that did not exist on the legacy application. These features consist on the visualization of single and aggregate information, such as temperature or electrical consumption, on a single room or a group of rooms like a *Nucleo*. This feature consisted on a 3D Tag object located in the center of each location, the aggregate location's Tag were visible on the farthermost zoom and the single ones were only visible on the closest ones. As we know, the legacy application did not offer similar features which would allow us to compare with our prototype, so these context measurements will only serve to test if the current implementation would allow users to easily perform the intended task within a reasonable time. As mentioned before in another Chapter, this evaluation context was tested only at the final evaluation phase in which User A and B did no participate and so only users C to G results will be considered.

The first Spatial Information Monitoring evaluation context task results were satisfactory and all users were able to perform the given task under the specified time. In fact, as depicted on Figure 5.40, only user D took more than thirty seconds to finish the task. This task only required to visualize an aggregate area type of Tag which did not require any kind of zoom making it easy to see the tag; then the user had to change the tag from the location name to electrical consumption, through the vertical menu.

In the other hand the last task results where not as good as expected, User F almost did not finish the task in time and all user's average task time was one minute and thirty nine seconds, which although it is not a bad result, but by comparison with all other FM3D performed tasks, it

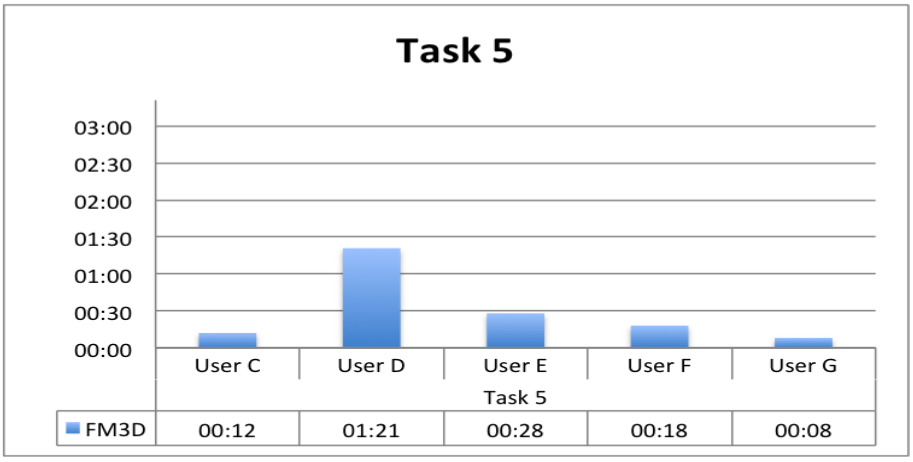


Figure 5.40: Time chart displaying the time taken by each user to perform task five on FM3D.

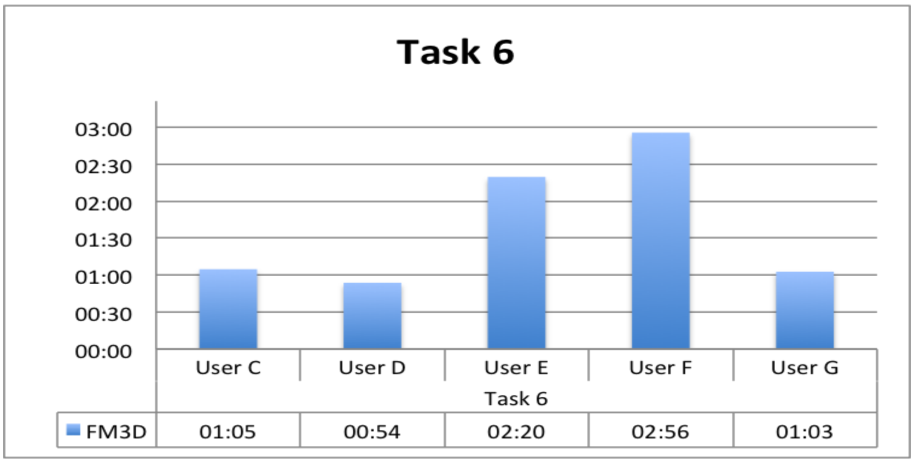


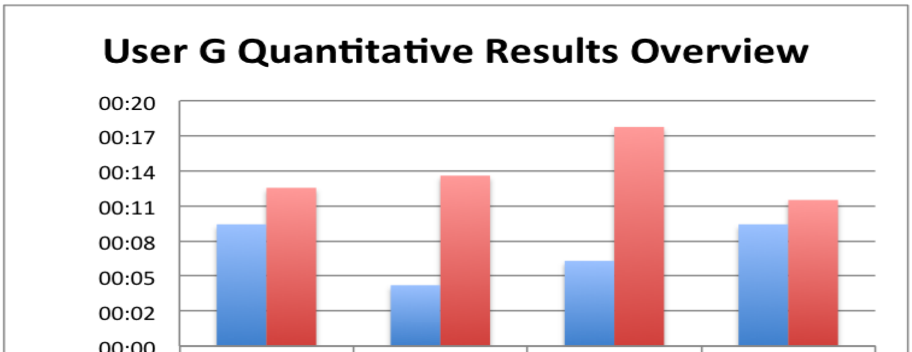
Figure 5.41: Time chart displaying the time taken by each user to perform task six on FM3D.

was by far the worst one. When we asked the participants why they have taken so much time to complete this task, it was stated that they took time to visualize the A1 Amphitheater's Tag because they did not knew that A1 was a single area, and so it was necessary to zoom in for the tag to show up. That is, they selected the correct floor and knew where A1 was located, but got stuck on but got stuck on the process. They where looking at the room and there was no tag above it and for awhile, they did not knew what to do. Only after a while they started interacting with the zooming menu and found out how that the correct single room tag only appeared on zooming in that location. We think that this aspect should mandatorily be improved in further prototype versions.

5.4.3 Advanced Participant results overview

The advanced participant is the one that often uses this type of software, so its feedback offers a more realistic opinion on the needs that our building automation systems should respond to, as well as validate the premisses we have taken during our prototype design of what our application should do and how it should do it. Due to the importance of a good advanced user feedback in this topic, we will try to elaborate a deeper analysis of his answers and opinion on both the qualitative and quantitative part of the results.

We will start through the observation of this quantitative study results.

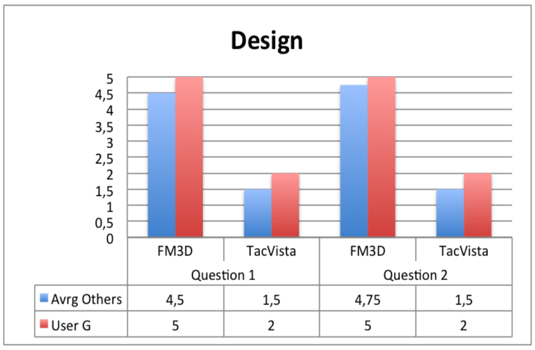


	Task 1	Task 2	Task 3	Task 4
FM3D	00:10	00:05	00:07	00:10
TacVista	00:13	00:14	00:18	00:12

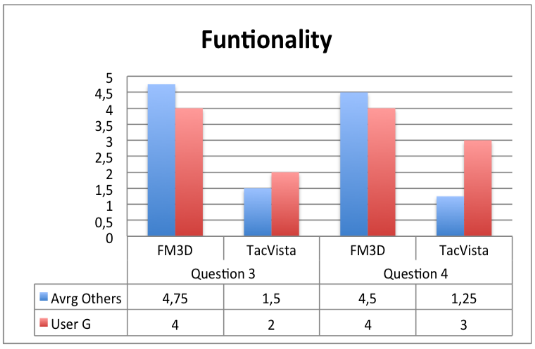
Figure 5.42: Time chart displaying an overview of the time taken by the advanced participant to perform the HVAC and Illumination evaluation context tasks on both applications.

The advanced participant interacts daily with the legacy application and we expected him to present the fastest TacVista results of all participants. In fact that was the case, at all tasks he was the fastest TacVista user with an average legacy application task time execution of fourteen seconds and with the worst task time execution of eighteen seconds. The results were truly remarkable, that we thought it would be hard to surpass. With this in mind, our concern was to see how would be its task execution times on our prototype, being so that he was using the FM3D software for the first time.

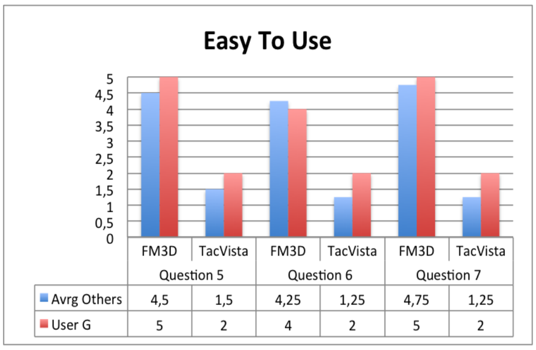
Due to the fact that he had to learn how to interact with our application and that he was already an expert on TacVista interaction we were prepared to obtain much better results from his interaction with TacVista than with our prototype. So it was with great surprise that when we compared both results and verified that this was not the case. As depicted on Figure 5.42, even with no experience on using FM3D application measured an average time of eight seconds, that is, almost half the time he presented on his common use application. In some cases, such as at task two he was three times faster with a five seconds task completion value against fourteen seconds with the legacy one.



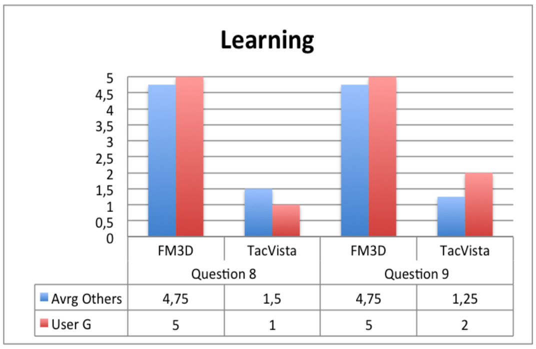
(a) Design



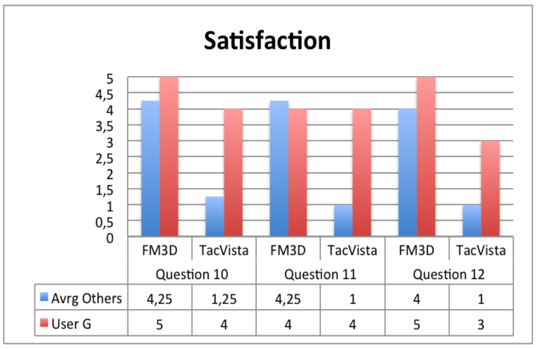
(b) Funtionality



(c) Easy to Use



(d) Learning



(e) Satisfaction.

Figure 5.43: Overview of the average qualitative results of other participants versus the advanced user. Results shown are relative to the aspects of: (a)Design, (b)Functionality, (c)Easy to Use, (d)Learning and (e)Satisfaction context purpose questions.

From Figure 5.43 we can compare the advanced participant's score value against the average value of all other participants on the corresponding software and question. Generally speaking, the advanced user qualitative evaluation presented similar results to all other user's average score. The main differences occurred on the functionality and satisfaction aspects of the evaluation context, where user G rated TacVista with some significative better score than all other participants. This fact was not surprising and can be easily explained. Starting by the functionality component, at question 4 *The system's available information is helpful to complete all tasks* the user's average result was of 1,5 where user G rated as 3. Being an advanced user he is used to work with TacVista to perform its daily task so it is understandable he knows where to find the system information he needs, by comparison all other users did not, which explains the 1.5 value discrepancy between both values. Relative to the satisfaction evaluation context, most users were not able to complete all tasks on TacVista. They found it really hard to interact without training or direct supervision so it is comprehensible that they gave a really low average score value to that application. In the advanced user case, as mentioned before he uses TacVista daily and completed all tasks which made him give a much higher score value. Although User G gave satisfactory scores to TacVista at functionality and Satisfaction contexts at the evaluation of all evaluation contexts he, as well as all other participants, rated FM3D with much better values allowing us to conclude that although our FM3D application is still an unfinished prototype it is clear that qualitatively speaking, its results surpassed the legacy one in all evaluation contexts making it a much better, user friendly application to interact with.

5.5 Discussion

This evaluation validates literature reports pointing to an increase in performance of 3D interfaces over traditional ones and shows that new approaches to interact with spatial information information are not only feasible but desirable. Our prototype results demonstrate that it is possible to create appealing 3D interfaces for visualizing BAS information in an integrated manner concerning space configuration, as well as, their respective operation conditions and impact within the space.

Through the usability tests that we have conducted we have reasons to state that 3D interactive environments have the potential to significantly increase the productivity in maintenance tasks. In these tests, users without training demonstrated a high degree of engagement and performance operating our 3D interface prototype, specially when compared with that of the legacy application. The potential drop in user time and increase in engagement with a 3D environment could eventually translate into lower cost and to an increase in quality, potentially turning 3D based interface the option of choice in future IT tools for BAS.

In our approach it remains unclear to what extent the integration at the interface level is contributing to increase the users productivity. Presumably, not all maintenance activities benefit from an approach such as the one we propose in the same way. Additional experiments are required to gain insight on which aspects of a 3D interface contribute to certain maintenance activities. The work reported here, is part of a wider effort that aims at creating disruptive new ways to explore building information. As for future work, we plan to incrementally develop our prototype into a data integration and visualization platform capable of integrating and interacting with BAS data from different types of tools in novel ways.

Chapter 6

Conclusion

6.1 Lessons Learned

We discussed the limitations of the current Building Automation Systems state of the art regarding: the graphical representation of the location of devices and the visualization of their corresponding status within the existing user interfaces for command and control of large BAS. We analyzed several applications usage of 3D Virtual Interactive Environments (VIE) and computer aided design tools employed in the VIE construction. Through this analysis we observed that existing BAS tool interfaces are often quite limited with respect to displaying spatial data. Many do not display data using any type of planimetric representation which makes navigating on spatial information quite limited. This forces users to switch to another screen to analyze the information regarding different zones. Facilities plans are often static pictures used for navigating from one space to another. These tools exhibit a general lack of spatial dynamism because they do not offer smooth transition between space zones. Another important aspect of navigation is alternating between different levels of detail, between aggregated data and detailed data. Management, supervision, and diagnostic activities require to quickly switch form a managerial into an operational view alternating between indicators at different levels of aggregation or from an overview perspective to a detailed inspection perspective. Navigating back and forth among different levels of detail and aggregation should be a smooth and fast operation that does not imply jumping between screens.

At the end of our literature research, we concluded that none of the literature gathered from this area of study endorsed an architecture that fulfills all of the identified needs. With this in mind we looked at a possible solution to the thesis problem by taking advantage of the features provided by

3D virtual interactive environments. As we know tridimensional interactive environments can be cost effective since they can be operated by ordinary users with less formal training. Its interaction is becoming commonplace and therefore more and more users are becoming proficient at using them. Also, a great advantage, is that tridimensional interfaces are quite effective at rendering spatial information but also enable interactivity.

Putting these concepts into practice, we designed and implemented an building automation system interface architecture supporting them. Our prototype implementation was called FM3D and to prove that a building automation control can greatly benefit from tridimensional interactive interfaces we conducted a two phased evaluation test. This would allows us to determine if our FM3D prototype application when compared it with an existing legacy application for centralized control and monitoring would actually benefit user engagement and performance while performing building automation tasks. Featuring the legacy application we choose the existing Schneider Electric's TAC Vista application interface that was already installed and working at the IST-Taguspark were the tests took place. At the end of these tests we verified that our prototype presented at both qualitative and quantitative, levels much better results then the legacy application. Also the advanced participant testing and feedback corroborated all our results which meant that although our application was still a prototype, as intended, its implementation could prove to be a useful tool for building automation interaction.

At the end, we concluded that this we work validates the literature reports pointing to an increase in performance of 3D virtual environments over traditional interfaces and shows that new approaches to interact with spatial information are not only feasible but desirable. In addition, our prototype also demonstrates that it is possible to create appealing 3D interfaces for visualizing FM information in an integrated manner concerning space configuration, equipment location as well as their respective operation conditions and impact within the space. The usability tests we have conducted indicate that these kind of interfaces have the potential to significantly increase the productivity in maintenance tasks. Our participants, without training demonstrated a high degree of engagement and performance operating our prototype interface, when compared with that of the tested legacy application. We think that the potential drop in user time and increase in engagement with a 3D interface will eventually translate into lower cost and to an increase in quality, potentially turning tridimensional based interface the option of choice in future IT tools for building automation.

6.2 Future Work

In our approach it remains unclear to what extent the integration at the interface level is contributing to increase the users productivity. Presumably, not all maintenance activities benefit from an approach such as the one we propose in the same way, so it is thought that additional experiments should be required to gain insight on which aspects of a tridimensional interface contribute to certain maintenance activities. In addition, the advantages of a web-based interface, accessible from mobile devices, were not subject to user testing. To that end, further evaluation should be carried out with different use cases, which could be tested not only in a central location like the IST-Taguspark facility, but form any place.

Although proved effective, some functional as well as graphical components should be improved. That is the case of the graphical aspect of both the HVACs particle system and the devices GUIs. Due to the shaders used, the HVAC particle effect was not as visually pleasing as we expected, it lacks on the visual contrast between the hot and cold types of temperature. That is, the zone were is a great concentration of particles being drawn get an subtractive color effect become a white zone instead of a red or blue one as the intended temperature. The devices GUI, present a unhewn look they should be visually redesigned to supply users with a more fluid and enjoyable working interface.

There are some aspects of the Tag system implementation that should be modified. For instance when a space as several Tag types (about more than 5) it should not be possible to visualize them all in the Tag pile. Without this limitation, the system is be open to such a great pile of Tags that it is not be possible to visualize them all in our camera field of view. In these cases, the visibleTags customization could be achieved through the GUI interface Tag types menu selection. Also, the whole building model should be redesigned in a more modular way, dividing each room individually and grouping them hierarchically. Through this modularization we could establish a direct association between each tridimensional room and its logical information, which was not possible in our current implementation.

The work reported here is part of a wider effort that aims at creating innovative new ways to explore building information. In the future, we intent to develop our prototype into a data integration and visualization platform capable of both integrating and interacting with BAS data from different types of tools in novel ways. Our vision is that the nature and quantity of information that a 3D virtual interactive environment can support, together with a distributed and mobile access to such information, will encourage more applications than the ones devised until now.

Bibliography

ALESHEIKH, H., A.AND HELALI & BEHROZ, H. (2002). Web GIS: Technologies and its applications. In *Symp. on Geospatial Theory, Processing and Applications*.

ALKAR, A. & BUHUR, U. (2005). An Internet Based Wireless Home Automation System for Multifunctional Devices. *IEEE Transactions on: Consumer Electronics*, **51**, 1169–1174.

ASHRAE (2004). *BACnet A Data Communication Protocol for Building Automation and Control Networks (Standard 135-2004 – ANSI Approved)*.

BALES, R. (1950). *Interaction Process Analysis: A Method for the Study of Small Groups*. Cambridge, Addison-Wesley P.

BENDER, K. (1993). *Profibus: The fieldbus for industrial automation*. Prentice-Hall, Inc.

BROWN, E. & CAIRNS, P. (2004). A Grounded Investigation of Game Immersion. In *Proc. CHI 2004, ACM Press*, 1297–1300.

CORPORATION, E. (1999). *Control Network Protocol Specification (ANSI/EIA/CEA 709.1)*.

DANEELS, A. & SALTER, W. (1999). What is SCADA? In *Proceedings on the International Conference on Accelerator and Large Experimental Physics Control System, Trieste, Italy*.

DAVIS, R., BURNS, A., BRIL, R. & LUKKIEN, J. (2007). Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, **35**, 239–272.

DIETRICH, D. & SAUTER, T. (2000). Evolution Potentials for Fieldbus Systems. In *IEEE International Workshop on Factory Communication Systems WFCS*.

DONATH, J. (1997). Inhabiting the Virtual City: The design of social environments for electronic communities.

EASTMAN, C., TEICHOLZ, P., SACKS, R. & LISTON, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Wiley.

ELLIS, S. (1994). What are virtual environments? *Computer Graphics and Applications, IEEE*, **14**, 17–22.

ELMAHDY, A. (1980). An overview of central control and monitoring systems for large buildings and building complexes. *Building Research Note*, **1**, 1–26.

ELMUALIM, A. & JOHNSON, A. (2009). Application of computer-aided facilities management (CAFM) for intelligent buildings operation. *Facilities*, **27**, 421–428.

FREEMAN, W., ANDERSON, D., BEARDSLEY, P., DODGE, C., ROTH, M., WEISSMAN, C., YERAZUNIS, W., KAGE, H., KYUMA, K., MIYAKE, Y. *et al.* (1998). Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications*, **1**, 42–53.

FROEHLING, P., ANDERS, J., STROJNY, L. & REHBERGER, J. (1985). Direct digital control apparatus for automated monitoring and control of building systems. US Patent 4,497,031.

FULLERTON, T., SWAIN, C. & HOFFMAN, S. (2008). *Game design workshop: A playcentric approach to creating innovative games*. Morgan Kaufmann.

HAINES, R. & HITTLE, D. (2006). *Control Systems for Heating, Ventilating, and Air Conditioning*. Springer Verlag.

HEESOM, D. & MAHDJOUBI, L. (2004). Trends of 4D CAD applications for construction planning. *Construction Management and Economics*, **22**, 171–182.

HIMANEN, M. (2003). *The Intelligence of Intelligent Buildings The Feasibility of the Intelligent Building Concept in Office Buildings*. Ph.D. thesis, VTT.

HUNT, R., K.AND HAAS & KALKKUHL, J. (1996). Local controller network for autonomous vehicle steering. *Control Engineering Practice*, **4**, 1045–1051.

KASTNER, W., NEUGSCHWANDTNER, G., SOUCEK, S. & NEWMANN, H.M. (2005). Communication systems for building automation and control. *Proceedings of the IEEE*, **93**, 1178–1203.

KAUR, I. (2011). Microcontroller Based Home Automation System With Security. *International Journal of Advance Computer Science and Applications*, **1**, 60–65.

KEKE, X. & XIAOJUN, N. (2010). Design and Implementation of 3D Virtual Campus Roaming System Based on GIS. In *2nd WRI Global Congress on Intelligent Systems (GCIS)*, vol. 2, 137–139, IEEE.

KELLER, J. & KELLER, C. (2005). Bringing strategic efficacy to facility management through CAFM tools. *Journal of Facilities Management*, **3**, 125–144.

KONNEX ASSOCIATION (2004). *KNX Specification Version 1.1*.

KORVA, T. (2004-2005). Open Source and Low Cost Game Engine. Tech. rep., LudoCraft.

LEE, G., SACKS, R. & EASTMAN, C. (2006). Specifying parametric building object behavior (BOB) for a building information modeling system. *Automation in construction*, **15**, 758–776.

LIN, H. (2006). A Collaborative Virtual Geographic Environment: Design and Development. *Collaborative Geographic Information Systems*, **1**, 186.

LIN, H., GONG, J. & SHI, J. (2003). From maps to GIS and VGE-a discussion on the evolution of the geographic language. *Geography and Geo-Information Science*, **19**, 18–23.

MAGUIRE, D. (1991). An overview and definition of GIS. *Geographical Information Systems: Principles and Applications*, **1**, 9–20.

MARCUSE, H. (2002). *One-dimensional man: Studies in the ideology of advanced industrial society*. Taylor & Francis Ltd.

MERZ, H., HANSEMANN, T. & HÜBNER, C. (2009). *Building Automation: Communication Systems with EIB/KNX, LON und BACnet*. Springer Verlag.

MODBUS (2002). *Application Protocol Specification V1.1*.

PAGAN, W. (2006). Simultaneous View and Point Navigation. US Patent App. 20,080/143,722.

PAGULAYAN, R.J., KEEKER, K., WIXON, D., ROMERO, R.L. & FULLER, T. (2002). User-centered Design in Games. In *The human-computer interaction handbook*, 883–906, L. Erlbaum Associates Inc.

PARRINDER, E. (1970). *Avatar and Incarnation*. Barnes & Noble.

RHODES, B. (1997). The wearable remembrance agent: A system for augmented memory. *Personal and Ubiquitous Computing*, **1**, 218–224.

RIVEST, S., BÉDARD, Y., PROULX, M., NADEAU, M., HUBERT, F. & PASTOR, J. (2005). SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. *ISPRS Journal of Photogrammetry and Remote Sensing*, **60**, 17–33.

ROY, S. (2003). State of the art of Virtual Reality Therapy (VRT) in phobic disorders. *PsychNology Journal*, **1**, 176–183.

RUSS, G., DIETRICH, D. & TAMARIT, C. (2003). *Situation-dependent behavior in building automation*. Ph.D. thesis, Technische Universität Wien.

SAWYER, B. & SMITH, P. (2008). Serious games taxonomy. In *Slides from the Serious Games Summit at the Game Developers Conference*.

SCHICKHUBER, G. & MCCARTHY, O. (1997). Distributed fieldbus and control network systems. *Computing & Control Engineering Journal*, **8**, 21–32.

SHEPHERD, I.D.H. & BLEASDALE-SHEPHERD, L.D. (2008). Towards Effective Interaction in 3D Data Visualizations: what can we learn from videogames technology. In *International Conf. on Virtual Geographic Worlds*, 7–8.

SQUIRE, K.D. (2003). Video Games in Education. *Int. Journal of Intelligent Games & Simulation*, **2**, 49–62.

STEIN, M., KAUFMAN, T., RICHARZ, Y., TARLOW, K. & NESBITT, B. (2000). User interface for home automation system. US Patent 6,140,987.

SUSI, T., JOHANNESSON, M. & BACKLUND, P. (2007). Serious games–An overview. Tech. rep., School of Humanities and Informatics, University of SkÃ¶vde.

TAYLOR, L.N. (2002). *Video games: Perspective, point-of-view, and immersion*. Ph.D. thesis, University of Florida.

THOMESSE, J.P. (2005). Fieldbus technology in industrial automation. *Proceedings of the IEEE*, **93**, 1073–1101.

WHITE, T.C., WONG, C.W., NGUYEN, K., SHETH, J.V. & HARRIS, C.W. (1996). Interbus interface module. US Patent 5,519,883.

WIMSATT, W. (2004). Home automation contextual user interface. US Patent App. 20,040/260,427.

WÜNSCHE, B. C. AND KOT, B. AND GITS, A. AND AMOR, R. AND HOSKING, J. (2005). A Framework for Game Engine Based Visualisations. In *Proceedings of Image and Vision Computing New Zealand,*.

ZEISS, G. (2011). Building information modeling. In *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications*, ACM.

Appendix A

Evaluation Script

Estudo com Utilizadores
Aplicações para Gestão de Edifícios

1 Introdução

O presente documento visa apresentar o protocolo a seguir durante o estudo comparativo entre a aplicação de gestão de edifícios **TacVista** e o protótipo preliminar da aplicação **FM3D**. Este estudo surge no âmbito de um projecto de mestrado cujo objectivo é investigar a aplicabilidade do uso de interfaces com ambientes virtuais 3D nas aplicações de gestão de edifícios.

Para efectuar o referido estudo, serão efectuados testes com utilizadores aos quais será pedido para realizarem um conjunto de tarefas semelhantes em ambas as aplicações. Com base nos resultados quantitativos e qualitativos desses testes será possível aferir a validade da abordagem proposta, identificando vantagens e desvantagens relativamente aos sistemas tradicionais.

1.1 TacVista

A aplicação TacVista trata-se de um sistema de BMS de monitorização e controlo das componentes de HVAC, Iluminação e controlos de acesso de um ou varios edificios. É baseado em LONWorks e é supostamente o sistema mais IT-friendly e escalavel existente no mercado.

1.2 FM3D

A aplicação FM3D trata-se de um prototipo de BMS de monitorização e controlo das componentes de HVAC, Iluminação de um ou varios edificios. É um prototipo ainda em fase de testes que pretende ser uma melhoria em relação aos sistemas de BMS existentes através do uso de interfaces com ambientes virtuais 3D nas aplicações de gestão de edifícios.

2 Metodologia

A sessão de testes com os utilizadores está organizada em várias etapas, identificadas na próxima tabela, juntamente com a respectiva duração prevista. Segue-se à tabela uma explicação de cada uma das referidas etapas.

1	Introdução à experiência e às aplicações	10 min
2	Preenchimento de questionário "Perfil"	5 min
3a	Experiência de utilização da aplicação FM3D	15 min
4a	Preenchimento de questionário "Avaliação Aplicação A"	5 min
3b	Experiência de utilização da aplicação TacVista	15 min
4b	Preenchimento de questionário "Avaliação Aplicação B"	5 min

Etapa 1 - Introdução à experiência e às aplicações: No início da experiência explica-se ao utilizador o propósito da sessão e tenta-se familiarizá-lo com as aplicações em causa. Para o efeito, serão apresentadas as principais funcionalidades de cada aplicação e, de seguida, decorrerá uma curta fase de treino, composta por um período de três minutos para interagir livremente com cada uma das aplicações.

Appendix B

Questionnaire

Estudo com Utilizadores

Parte 1 - Perfil do Utilizador

____

Idade:

Habilitações Académicas:

Sexo:

____

Proficiência no uso de aplicações informáticas:

Usa poucas vezes ☐ Usa algumas vezes: ☐ Usa frequentemente: ☐

Nunca usou ☐

a) **Familiaridade na interacção com interfaces informaticas:**

   Telemovel ☐ Computador ☐

   Ecrã Tactil ☐ Outra :

b) **Familiaridade na interacção com interfaces 2D :**

   Ja usou ☐ Não usou ☐

c) **Familiaridade na interacção com interfaces 3D :**

   Ja usou ☐ Não usou ☐

____

Proficiência no uso de interfaces de controlo de edifícios:

Já usou ☐ Nunca Usou ☐

Apenas resposta as seguintes questões se a sua resposta anterior tiver sido "Já usou"

a) **Tipo de interface de controlo de edificio usadas:**
   2D ☐ 3D ☐

b) **indique quais as interfaces de controlo de edificios que ja utilizou.**

   ____

____

____

Parte 2 - Realização de tarefas sobre a aplicação **FM3D**

____

Explicação do procedimento:

No inicio de esta fase ser-lhe-ão apresentadas as principais funcionalidades da aplicação e, de seguida, decorrerá uma curta fase de treino, composta por um período de três minutos para interagir livremente com a aplicação.

De seguida ser-lhe-á pedido que execute um conjunto de tarefas na aplicação.

No início de cada uma das tarefas o sistema é sempre colocado num estado inicial. Para o efeito após terminar a tarefa, ou esgotar o tempo para o fazer, e antes de iniciar a nova tarefa, um dos membros da equipa de desenvolvimento deverá repor o estado inicial da aplicação. Durante este curto intervalo em que é resposto o estado do sistema ser-lhe-á pedido que descreva as principais dificuldades que teve no decorrer da tarefa anterior sendo-lhe de seguida apresentada a tarefa seguinte.

O tempo máximo de realização de cada tarefa é de 3 minutos.

Tarefa 1 – Ligar a iluminação do anfiteatro 1

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 1:

____

____

Tarefa 2 – Verificar o estado de iluminação da area localizada em frente ao anfiteatro A2.

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 2:

____

____

Tarefa 3 – Verificar se o AC do anfiteatro A2 está ligado e se está a aquecer

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 3:

____

____

Tarefa 4 – Colocar o AC do anfiteatro A2 a 22C

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 4:

____

____

Tarefa 5 – Verificar através da utilização das **Etiquetas**, qual o **Consumo** do Núcleo 2-N2

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 5:

____

____

Tarefa 6 – Verificar através da utilização das **Etiquetas,** qual a **Temperatura** do Anfiteatro A1

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 6:

____

____

____

Parte 3 - Questionário sobre a apreciação da aplicação **<u>FM3D</u>**

____

<u>Software testado – FM3D</u>

Nesta fase ser-lhe-á pedido que responda às seguintes questões utilizando a escala lateral apresentada abaixo. Estas questões destinam-se à apreciação da aplicação do ponto de vista do utilizador.

1 – Discordo totalmente

3 – Não concordo nem discordo

5 – Concordo totalmente

	Escala				
Pergunta	**1**	**2**	**3**	**4**	**5**
Design					
1. Gostei de utilizar a interface da aplicação					
2. A interface da aplicação é agradável de utilizar					
Funcionalidades					
3. A aplicação tem todas as funcionalidades que eu esperava					
4. A informação disponível pelo sistema (sendo gráfica ou não) é eficaz na ajuda que dá para completar as tarefas					
Facilidade de Utilização					
5. A aplicação e fácil de utilizar					
6. É fácil encontrar a informação que preciso					
7. A aplicação e globalmente fácil de utilizar					
Aprendizagem					
8. É fácil aprender a usar a aplicação					
9. A informação fornecida pela aplicação é fácil de aprender					
Satisfação					
10. Sinto-me confortável a utilizar a aplicação					
11. Consigo completar eficazmente as tarefas					
12. Na globalidade estou satisfeito com a minha resolução das tarefas					

<u>Comentários/Observações</u>

____

____

____

Parte 4 - Realização de tarefas sobre a aplicação **TacVista**

Explicação do procedimento:

No inicio de esta fase ser-lhe-ão apresentadas as principais funcionalidades da aplicação e, de seguida, decorrerá uma curta fase de treino, composta por um período de três minutos para interagir livremente com a aplicação.

De seguida ser-lhe-à pedido que execute um conjunto de tarefas na aplicação.

No início de cada uma das tarefas o sistema é sempre colocado num estado inicial. Para o efeito após terminar a tarefa, ou esgotar o tempo para o fazer, e antes de iniciar a nova tarefa, um dos membros da equipa de desenvolvimento deverá repor o estado inicial da aplicação. Durante este curto intervalo em que é resposto o estado do sistema ser-lhe-á pedido que descreva as principais dificuldades que teve no decorrer da tarefa anterior sendo-lhe de seguida apresentada a tarefa seguinte.

O tempo máximo de realização de cada tarefa é de 3 minutos.

Tarefa 1 – Ligar a iluminação do anfiteatro 1

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 1:

____

____

Tarefa 2 – Verificar o estado de iluminação da area localizada em frente ao anfiteatro A2.

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 2:

____

____

Tarefa 3 – Verificar se o AC do anfiteatro A2 está ligado e se está a aquecer

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 3:

____

____

Tarefa 4 – Colocar o AC do anfiteatro A2 a 22C

Descreva sucintamente as principais dificuldades que teve durante a realização da Tarefa 4:

____

____

____

Parte 5 - Questionário sobre a apreciação da aplicação **TacVista**

____

Software testado – TacVista

Nesta fase ser-lhe-à pedido que responda às seguintes questões utilizando a escala lateral apresentada abaixo. Estas questões destinam-se à apreciação da aplicação do ponto de vista do utilizador.

1 – Discordo totalmente

3 – Não concordo nem discordo

5 – Concordo totalmente

Pergunta	**Escala** 1	2	3	4	5
Design					
1. Gostei de utilizar a interface da aplicação					
2. A interface da aplicação é agradável de utilizar					
Funcionalidades					
3. A aplicação tem todas as funcionalidades que eu esperava					
4. A informação disponível pelo sistema (sendo gráfica ou não) é eficaz na ajuda que dá para completar as tarefas					
Facilidade de Utilização					
5. A aplicação e fácil de utilizar					
6. É fácil encontrar a informação que preciso					
7. A aplicação e globalmente fácil de utilizar					
Aprendizagem					
8. É fácil aprender a usar a aplicação					
9. A informação fornecida pela aplicação é fácil de aprender					
Satisfação					
10. Sinto-me confortável a utilizar a aplicação					
11. Consigo completar eficazmente as tarefas					
12. Na globalidade estou satisfeito com a minha resolução das tarefas					

Comentários/Observações

____

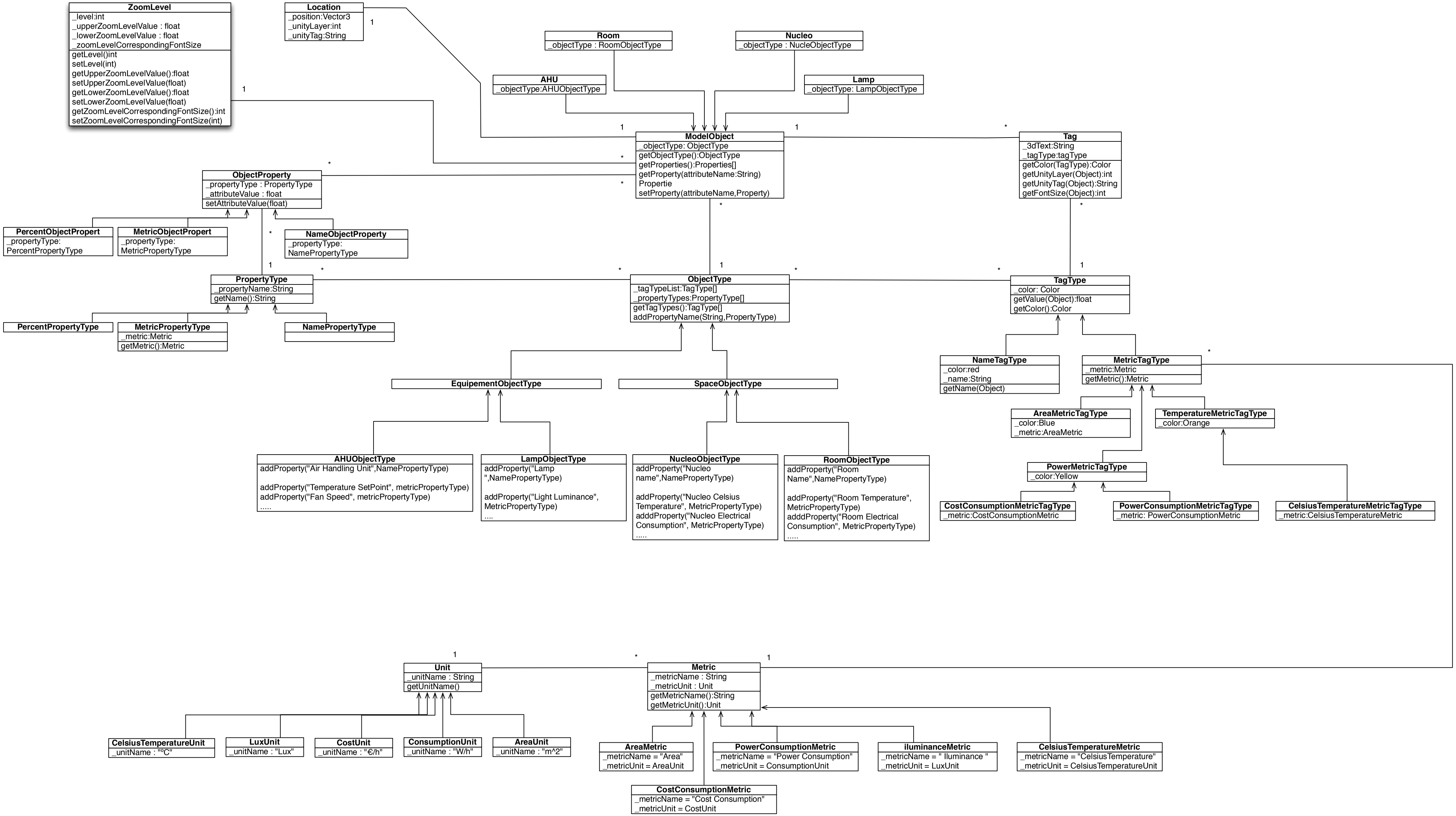
____

____

Obrigado pela sua participação!

Appendix C

FM3D Domain Class Diagrams


ZoomLevel
_level:int
_upperZoomLevelValue : float
_lowerZoomLevelValue : float
_zoomLevelCorrespondingFontSize
getLevel()int
setLevel(int)
getUpperZoomLevelValue():float
setUpperZoomLevelValue(float)
getLowerZoomLevelValue():float
setLowerZoomLevelValue(float)
getZoomLevelCorrespondingFontSize():int
setZoomLevelCorrespondingFontSize(int)
Location
_position:Vector3
_unityLayer:int
_unityTag:String
Room
_objectType : RoomObjectType
Nucleo
_objectType : NucleObjectType
AHU
_objectType:AHUObjectType
Lamp
_objectType: LampObjectType
ModelObject
_objectType: ObjectType
getObjectType():ObjectType
getProperties():Properties[]
getProperty(attributeName:String)
Propertie
setProperty(attributeName,Property)
Tag
_3dText:String
_tagType:tagType
getColor(TagType):Color
getUnityLayer(Object):int
getUnityTag(Object):String
getFontSize(Object):int
ObjectProperty
_propertyType : PropertyType
_attributeValue : float
setAttributeValue(float)
PercentObjectPropert
_propertyType:
PercentPropertyType
MetricObjectPropert
_propertyType:
MetricPropertyType
NameObjectProperty
_propertyType:
NamePropertyType
PropertyType
_propertyName:String
getName():String
ObjectType
_tagTypeList:TagType[]
_propertyTypes:PropertyType[]
getTagTypes():TagType[]
addPropertyName(String,PropertyType)
TagType
_color: Color
getValue(Object):float
getColor():Color
PercentPropertyType
MetricPropertyType
_metric:Metric
getMetric():Metric
NamePropertyType
NameTagType
_color:red
_name:String
getName(Object)
MetricTagType
_metric:Metric
getMetric():Metric
EquipementObjectType
SpaceObjectType
AreaMetricTagType
_color:Blue
_metric:AreaMetric
TemperatureMetricTagType
_color:Orange
AHUObjectType
addProperty("Air Handling Unit",NamePropertyType)
addProperty("Temperature SetPoint", metricPropertyType)
addProperty("Fan Speed", metricPropertyType)
.....
LampObjectType
addProperty("Lamp
",NamePropertyType)
addProperty("Light Luminance",
MetricPropertyType)
....
NucleoObjectType
addProperty("Nucleo
name",NamePropertyType)
addProperty("Nucleo Celsius
Temperature", MetricPropertyType)
adddProperty("Nucleo Electrical
Consumption", MetricPropertyType)
.....
RoomObjectType
addProperty("Room
Name",NamePropertyType)
addProperty("Room Temperature",
MetricPropertyType)
adddProperty("Room Electrical
Consumption", MetricPropertyType)
.....
PowerMetricTagType
_color:Yellow
CostConsumptionMetricTagType
_metric:CostConsumptionMetric
PowerConsumptionMetricTagType
_metric: PowerConsumptionMetric
CelsiusTemperatureMetricTagType
_metric:CelsiusTemperatureMetric
Unit
_unitName : String
getUnitName()
Metric
_metricName : String
_metricUnit : Unit
getMetricName():String
getMetricUnit():Unit
CelsiusTemperatureUnit
_unitName : "ºC"
LuxUnit
_unitName : "Lux"
CostUnit
_unitName : "€/h"
ConsumptionUnit
_unitName : "W/h"
AreaUnit
_unitName : "m^2"
AreaMetric
_metricName = "Area"
_metricUnit = AreaUnit
PowerConsumptionMetric
_metricName = "Power Consumption"
_metricUnit = ConsumptionUnit
iluminanceMetric
_metricName = " Iluminance "
_metricUnit = LuxUnit
CelsiusTemperatureMetric
_metricName = "CelsiusTemperature"
_metricUnit = CelsiusTemperatureUnit
CostConsumptionMetric
_metricName = "Cost Consumption"
_metricUnit = CostUnit