POLITECNICO DI MILANO

Facoltà di Ingegneria dell'Informazione

Corso di Laurea Magistrale in
Ingegneria Informatica

Dipartimento di Elettronica, Informazione e Bioingegneria

Tesi di laurea

Security requirements, modelling and analysis in a courthouse case study

Relatore: Prof. Carlo GHEZZI

Correlatore: Dr. Christos TSIGKANOS

Laureando:

WALTER RUBINO

Matr. 819703

Anno Accademico 2015-2016

Acknowledgments

First of all, I would like to thank my supervisor prof. Carlo Ghezzi for having proposed this engaging and interesting thesis; I also thank my co-supervisor dr. Christos Tsigkanos for the job done with me and the guidance to the correct formalization of steps to be followed in this software engineering project; it was a very instructive experience.

Another big thanks goes to the team of architects led by professor Ezio Arlati and composed of the doctors Sofia Bigoni and Gabriele Mansueto who collaborated with us providing the BIM project of the Palace of Justice in Pavia.

I have to thank all the people we worked with at the Palace of Justice, in a particular way the President of Court Annamaria Gatto, the Prosecutor of the Republic Giorgio Reposo and judicial officer Nadia Monelli, who participated in the interviews for requirements gathering.

I also can't forget to thank the members of the Ministero Infrastrutture e Trasporti - Provveditorato di Milano, in a particular way, Ing. Maurizio Clarizia and Ing. Gianni Gatto with whom we have collaborated for BIM experimentation focused on security.

Special thanks to my parents who have allowed me to achieve this qualification, they always supported me.

A final thanks, not least, goes to all my friends and colleagues who have supported me over the years, sharing with me joys and difficult times.

"Sì come ogni regno in sé diviso è disfatto, così ogni ingegno diviso in diversi studi si confonde e indebolisce."

Leonardo da Vinci

Contents

List of Figures

Listings

Sommario

Nel settore delle costruzioni, i modelli di informazione dell'edificio (BIM) rappresentano lo standard per la specifica di informazioni complesse riguardanti la costruzione delle infrastrutture. Utilizzando il Palazzo di Giustizia di Pavia come caso di studio e considerando le problematiche di sicurezza di questo particolare dominio, seguiamo i principi dell'ingegneria del software per consentire concretamente l'analisi dei requisiti di sicurezza sul progetto dell'edificio. Inizialmente viene affrontata la problematica della raccolta dei requisiti, a cui segue l'analisi dei requisiti. In tale dominio possono essere presi in considerazione requisiti riguardanti strutture statiche presenti nel progetto del Palazzo di Giustizia oppure valutare complessi scenari che evolvono in comportamenti dinamici dei beni e delle persone che si trovano all'interno. Vista la disponibilità delle descrizioni BIM dello spazio fisico del tribunale che compongono il dominio, siamo andati ad utilizzarle come fondamenta, allo scopo di ottenere modelli sensibili all'analisi attraverso l'impiego di tecniche per la trasformazione tra modelli. In particolare sfruttiamo la struttura presente nelle descrizioni Industry Foundation Classes e andiamo a generare un grafo rappresentante la topologia delle entità importanti presenti nel progetto degli spazi fisici del Palazzo di Giustizia. Concludiamo mostrando le analisi effettuate mediante strumenti esterni sui modelli ottenuti di requisiti generali, presi ad esempio, sia di natura fisica che dinamica.

Abstract

In the construction industry, Building Information Models (BIM) are the standard for specifying complex information about building infrastructures. Using the courthouse of Pavia as a case study, and considering security concerns of this particular domain, we follow software engineering principles to concretely enable analysis of security requirements of the building design. The problem of security requirements elicitation is initially addressed, followed by requirements analysis. Requirements in such a domain can predicate about static structures inherent in the courthouse physical space design, or reason about complex scenarios that involve dynamic behaviour of assets or people inhabiting it. Since BIM descriptions of the courthouse physical space were available for the domain, we utilize those as the foundation to obtain models amenable for analysis through model-to-model transformation techniques. In particular, we exploit structure in Industry Foundation Classes descriptions and generate a graph representing topology of important entities in the courthouse physical space design. We conclude by showcasing analysis over obtained models of generic example requirements of both static and dynamic nature, through external tools.

Chapter 1

Introduction

Security represents one of the most recent and important issue that influence our daily life. With the development of technological progress we have a wide range of devices such as cameras, metal detectors, pass to gain access to certain areas and so on, yet unfortunate events against things and people occur in places where we feel safe; just think of the shooting of the April 9, 2015 which took place in the Palace of Justice in Milan [1]. In that case, a man, involved in a bankruptcy case, gained access into the building with a weapon killing a judge, a lawyer and co-defendant. Obviously, something has gone wrong in the security dynamics applied in the Courthouse. Therefore, there is a need to perform more complex security analysis in order to minimize such situations.

1.1 Motivation

The proliferation of smart spaces, such as building automation systems, is providing new opportunities to improve building comfort, energy efficiency, and security. These systems are used to control lighting, heating, ventilation, air conditioning (HVAC), as well as other household appliances, communication systems, and security devices [2]. The systems used for the access control, mostly found in large organizations, must manage the potentiality of each employee's access to environments in function of his role and this may be particularly problematic in cases where the latter may vary frequently. With evolution also new challenges arise in security, in particular those related to the exploitation of the interaction between virtual and physical world, intrinsic vulnerabilities of systems that often are ignored. All this new technology, in fact, can support us in an effective manner in the storage and subsequent consultation of sensitive data, but also entails an increase of the security risks which is not immediately detectable.

1.1.1 Engineering dependable spaces

The increase of connectivity of devices has overcome the physical barriers that previously could represent a valid security system, facilitating in this way the access to the data as there is no longer the need to be physically present in the same place in which the devices are placed to access them. For example, cyber-enabled physical attacks can occur when physical assets are cyber-controlled (e.g., digital access control to buildings). Similarly, physically-enabled cyber attacks can occur when physical access to assets enables cyber attacks (e.g., access to a particular computer may facilitate malicious access to digital information held on that computer) [3].

Understanding and management of security threats characterizes a fundamental challenge, despite the literature being rich in accounts of methods of evaluation of security risks, they are not sufficiently effective for the sustaining of the operational environment changes. There is therefore the need for a renewal of the standards regarding security in public buildings, which are characterized by a high flow of people and sensitive assets. Standards which take into account new attack dynamics within an increasingly computerized environment. To achieve the increased reliability of the buildings that are designed, we're going to follow software engineering concepts to obtain analyzable models in order to perform the checks of the security requirements.

1.1.2 Security as a horizontal system

In the domain of a public building coexist different areas of interest that can be viewed as vertical systems, such as infrastructure, the public area, the area for the staff, people, devices and databases. These areas do not co-exist independently, but tend to intertwine and establish logical relations. In general the infrastructure will be split in public area and private area where people, objects and data exist together; therefore, in light of these dependencies, it is clear that they overlap as in Figure 1.1. The challenge is represented by the aim of having to intervene in all areas of interest making it, in fact, a horizontal security system intrinsically complex to implement. One of the aspects of fundamental importance that comes into play is certainly a correct modelling of the environment, on which the security may intervene in an effective way.

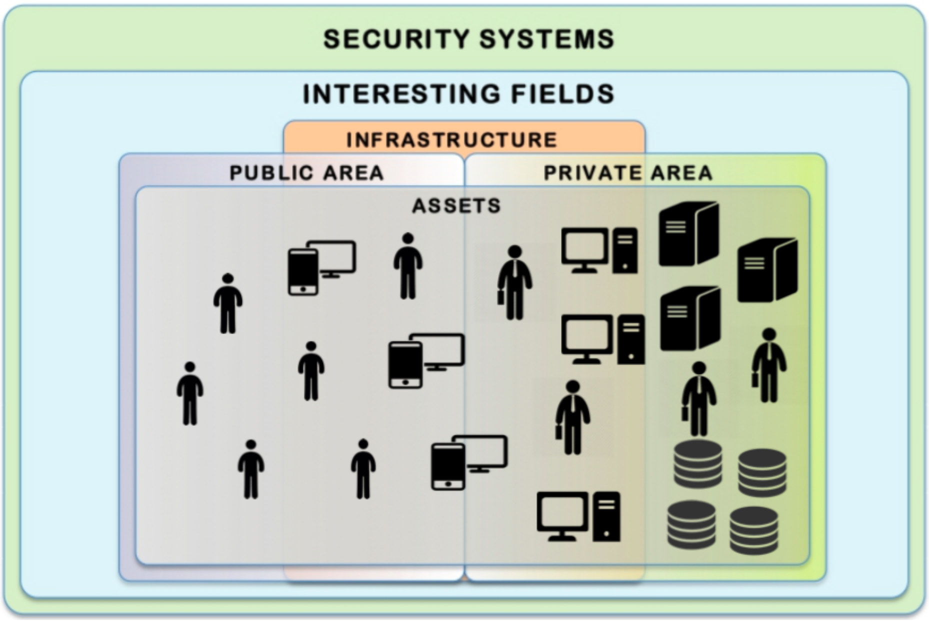

Figure 1.1. Areas of interest and security.

1.1.3 Operational environment

Security systems aim to protect critical assets in the face of changes in their operational environment. The key idea is to develop security systems able to perceive topological changes in the environment, to identify possible violations of security requirements, and finally, to select a set of security controls that prevent such violations [4]. We introduce an horizontal system representing a key characteristic for engineering security: the topology of the operational environment. It can denote the structure of a physical space, such as a building, the location of assets and agents in that space and their relationships. These relationships may determine whether assets or agents are close, if an agent can access a specific asset or location, or if an asset, agent or an area is enclosed by another area. In addition to existing context models, topology can provide a system with both structural and semantic awareness of important contextual characteristics that can affect security concerns. Similarly to the case studies related to a physical space, the topology of a cyber space can for example, capture the configuration of a network and the computing devices connected to it, also including relationships such as the containment of a file stored in a device, the proximity of two connected digital devices, and reachability of a file stored on a device from another one connected to it. To summarize, a topological representation of cyber and physical spaces can provide security-relevant contextual characteristics and support the verification of security requirements.

1.1.4 Static and dynamic analysis

When talking about a building we can define two moments of fundamental interest, the first, during its construction and the second during its opening to the public. During the planning stages, a security analysis at a static level makes possible a proper placement of the rooms where critical activities will be performed, with the aim to reduce or eliminate the likelihood of the risk scenarios related to assets which must be protected. The flow of people that can characterize a public building determines, instead, the need for an analysis at a dynamic level that allows the system to react proactively to the occurrence of situations that could potentially cause the violation of security requirements. The static aspect in the first moment is due to the lack of independent variations of topology while in the next step there is the need to deal with continuous changes in it, triggered by the movements of objects or agents in physical and in the virtual space. This last point leads to constant changes in the possible attack in the environment operational configurations where the topology changes dynamically.

1.2 Case study description

The Palace of Justice in Pavia is subject to a restructuring, since a team of architects of the Politecnico di Milano is taking part in, this has created the opportunity of a security trial-oriented, focusing at first on an analysis at a static level and then also to study a use case characterized from dynamic aspects. The project is proposed as a collaboration between architects and engineers in order to promote interoperability between these two worlds in the perspective of innovation-oriented design methods and construction of a public building.

1.2.1 History of Palazzo di Giustizia di Pavia

The building of the Palace of Justice in Pavia consists of several parts or blocks each of which has its own history, since the construction of the first part of the entire building or complex is still evolving. [5]

Figure 1.2. Panoramic picture of the Palace of Justice in Pavia.

Originally the building was formerly a convent of the Dove, which already existed in 1140 and was inhabited by Augustinian monks, it subsequently passed to the Lateran Canons and, in 1513, was united Hospital of St. Matthew. In 1539 the complex was granted to the Somaschi Fathers who owned it until 1810.

In 1810 the use of the building known as "La Colombina" changes and so the convent turns into a court. In 1820 the church Colombina is demolished prior a letter to the municipal congregation to obtain the permission. In 1837 several institutions are created: prisons and district court and twenty years after the court of first instance civil, criminal and mercantile, the state attorney and the prison service. In 1875 the building named the "Colombina" was sold. Since 1878 and for the next ten years the square of the court is restored and the year after its installation, restoration activities of the Palace of Justice began and they will be officially completed in 1918. In 1925, the courthouse is complete but already in 1931 it will expand its premises. In 1958 a renovation and subsequent expansion of the Palace of Justice took place.

Project completion and retrofitting of the Palace of Justice in Pavia has its roots in the early 2000s, a time when it was drafted, approved and contracted the project ”Reconstruction and enlargement of the Palace of Justice in Pavia”.

Subsequently, the Ministry of Justice has refinanced the work. The works have been redesigned according to the general scheme already approved at the time and subsequently divided into functional lots and covered by funding from the Ministry of Justice, in order to speed (as required by current legislation on Public Procurement) as much as possible the execution of works by providing space and functionality required by the Administration Usuaria. The project has been adjusted based on the demands and needs represented by the Administration of the Tribunal that have been implemented in progress. The new courtroom is a recent project that dates back to 2005. Between the two arms of the historic building a triangular block was inserted on multiple levels. The works which costed 5 million euro, include the construction of new courtrooms, the underground car park and some archives. The works, which will provide new spaces in view of the final amalgamation of the courts in Pavia, were divided into three lots. They will begin on the third floor of the former convent of the Dove, and then proceed to the outside buildings. The overall project also includes the courtrooms and parking areas: for these interventions had been expected to end by 2014, but due to the delays that have built up, jobs were shifted and are still being completed.

1.2.2 Problem Definition

After the analysis of the Court House, the architects realized the project of the entire building, from here we detect all the information content that can be important to the security analysis such as the layout of rooms, doors, windows etc. neglecting, however, all of purely aesthetic character information. The purpose of the project aims to create a model that enables the analysis of a static level security during design and then moving to the dynamic level during the building lifecycle. The model must, therefore, allow verification of requirements related to the construction project, such as the connection of related environments between them in terms of end-use or, on the contrary, the distance of environments for activities incompatible with each other. In addition, you must also be able to make a verification of the requirements related to the interactions of people with the assets and the environment in general, such as the isolation of activities reserved from the user public.

1.3 Thesis Structure

In Chapter 2 the theme of the requirements collection and issues related to it, is introduced. By gathering information on the state of art of global security standards, in particular the US, we develop a proper methodology of requirements gathering that we put into practice by questioning directly the key people of the Palace of Justice in Pavia. The purpose is to understand the needs and the possible problems that may occur in the organization of a security system, effective both at a static level and dynamic.

In Chapter 3, we introduce the concept of BIM from the origins and then analyze its spread in the world, evaluating the benefits and related issues. Afterwards, we define a derivative of BIM, lighter and capable of supporting the security analysis. From BIM, we carry on with its implementation, the technology of the Industry Foundation Classes starting, also in this case, from its origins to then define the standard, the architecture and finally, continuing with analysis of the relevant entities for the fulfillment of our purpose.

In Chapter 4 we describe our objectives, the assumptions made to approach the problem and the exposure of the solution strategy that involves the use of a transformation between models. We introduce the concept of "model-to-model transformation" and its mechanism by making use of an example. At this point, we reach the heart of the transformation that we are going to deepen by defining mappings between classes in the object model and the resulting constraints.

In Chapter 5 we talk about the analysis applied to the resulting model from the transformation considered as a bigraph materializing, in fact, the solution to the problem. We show the difference between the analysis of the static type and the dynamic by discussion of two prototypes.

In Chapter 6 we discuss about the documentation related to other work and approaches in this field of study, starting with a research study conducted at the Technical University of Monaco of Bavaria to reach the description of other approaches to international practices.

In Chapter 7 we present our conclusions regarding the work done and the possible developments that lie ahead in the future.

Chapter 2

Security Requirements in a Courthouse

The collection of security requirements is a basic thematic in the engineering of software, it requires an effort in terms of time and resources to obtain information that represent the base of work for the implementation. The delicacy of this phase of the project is given by the risk of wasting excessive resources, to avoid or limit this danger, a systematic approach supported by a good cooperation in the team and a good interfacement with the customer is required to understand the needs.[6] Among the common causes of lost of time we have:

- The project is in an early phase;
- The recruiting of new members of the team is ongoing;
- The team doesn't understand very well the client's objectives and therefore may not be able to meet expectations;
- An effective procedure for requirements gathering is not available or used.

In addition to the issues related to the team cooperation and contact with the customer, there are other factors which, if considered, can promote the success of the project. During the planning stages of assessment of the problems, it is good practice to be updated about the latest technologies on the market, as it could be of help in their resolution. The definition of a development path, from the early stages, can lead the team during the project and encourage an optimized use of resources. All aspects concerning communication are very important, including the most significant ones: effective change management in project requirements, facilitation of discussions and a good mediation in conflicts.

2.1 Problem domain

At the beginning of the development of a project, the first issue to be dealt with, is the application domain of the system to build. In fact, the domain is identified in the operational environment and, as consequence, the requirement collection must be carried out from a practical point of view oriented in such an environment.

Often the customer has only a basic idea, or it has none, concerning what the system must be able to do, or sometimes the presence of a predecessor can offer promising hints to develop, in all other cases it will be necessary to understand what will be the purpose of the system. [7] The definition of what the customer wants the system to be able to do, can be done regardless of any implementation or an existing solution, and this leaves completely open the space of solutions for engineers and architects; finally, you will need to define what actually is possible to realize as a function of the available resources.

Once the domain has been identified, it is necessary to define what will be the level of abstraction required to operate effectively his analysis, that is, what will be the level of detail to achieve in order to design the desired system. For example, if we have to make an assessment on a car to buy in a showroom, we will hardly be interested in the engine management of the system state transition diagram, rather, data regarding the acceleration, the maximum speed, the consumption fuel, the vehicle's capacity and the overall size will be relevant. Finally, the choice will have to comply with certain present constraints such as the maximum budget available for the purchase or the needs related to the delivery timing.

2.2 Use cases

The definition of a set of assumptions on which to base the establishment of a dialogue is an essential structural element to achieve a good level of productivity. These assumptions can be interpreted as a model of mutual understanding between the interlocutors. A basic tool, useful for the discussion of the requirements, is the use scenario, it can be used to define a framework in which meaningful dialogue can take place. The scenario encourages individuals and work teams in defining requirements, leading them to think what they would like to be able to do and how, in a situation like the one described. A scenario can be composed of a hierarchy of objectives desired by the parties, it represents the features offered by the system without specifying how they will be provided; so, this structure allows to better define the problem and avoid dwelling too much on possible solutions. During the definition of a scenario, it is important to analyze in depth the range of possibilities that may occur including the exceptions. Often, in several systems, the functionalities to manage this type of events are more complex than those that provide the basic functionality, through dialogue with the parties it is easier to identify actions that will be undertaken in order to return to one of the desired states. During the preparation of a scenario is best to set the boundary conditions a little larger than the expected system limits, which ensures that the agreed position will not be "blind" and it is used to set the system in its context. Only at the end, taking into account the costs that will accrue, the final scope of the system will be set.

2.3 Requirements in early design stages

Experienced users can propose changes which may be valuable, although, often, they are not considered so important. In fact, agreeing too easily to make changes to the project can cause failure; a good strategy may be to understand the importance level and choose accordingly if you opt for an immediate intervention or for a future update. The realization of a prototype that can be tested by users, it offers an idea of the possibilities of the planned system, and it allows a verification on the collected requirements and it highlights any issues which had been missed in the design phase. When planning the construction of a prototype is important to avoid an excessive waste of resources, to identify a small portion of the overall project to implement, so as to manifest a behaviour similar to that of the final system, keeping in mind that, at best, it is only an approximation.

The model of a system may be composed of several submodels that may potentially overlap, from each of them it is possible to derive the requirements, but the final goal is to have a structure which contains all of them. Each requirement must be formulated as a single testable statement that will then be placed in the model identified as the one with the widest reach. The placement of the requirements allows their verification, in a physical system such as a public building, it is preferable to use a model derived from the physical structure; in our case, the starting point will be the IFC model.

2.4 Drawing from other jurisdictions

To make a fair analysis of the security requirements, it is necessary to characterize the state of the standards defined by the most advanced countries in the security sector. The "Court Design Guide" [8] of the United States provides a good starting point to focus on the aspects of greatest relevance to the security analysis in a courthouse.

2.4.1 Courthouse actors

A key role is played by the importance of identifying which are the actors who may be involved in the scenarios that lead to operational changes of the environment and understand what their needs may be related to the ability to carry out their activities within the building. In the US standards, the following actors are identified:

- **Judicial Officer:** Generally, a single judge presides over civil and criminal proceedings, whereas a limited number of district cases are heard by panels of three judges.
- **Courtroom personnel:** In district judges' courtrooms, a courtroom deputy clerk and court reporter are present. One or two law clerks may also be present at the discretion of the judge.
- **Courtroom Deputy Clerk:** The courtroom deputy clerk, typically, is responsible for tracking all court activity during the proceedings, including the order of cases called, as well as documenting decisions by the court and tagging and caring for exhibits. The deputy clerk often confers with the judge during proceedings and can swear in the jury and witnesses.
- **Law Clerk:** They may attend courtroom proceedings and provide research assistance to the judge.
- **Reporter:** The court reporter records court proceedings.
- **Security Personnel:** US Marshall Service Personnel are responsible for the custody and movement of criminal defendants, the security of the courtroom and its occupants; and maintaining order in the courtroom. The total number of personnel depends on the nature of the matter being heard. In criminal cases, two personnel may be present for each in-custody defendant in the courtroom.

- **Attorney:** At least one attorney is present on each side of the case, trial attorneys can be assisted by paralegals and other personnel, who are positioned in the courtroom at the discretion of the presiding judge. In criminal cases, government attorneys are usually assisted in court by case agents from the investigative agency involved.
- **Litigant:** The number of litigants depends upon the number of parties and the nature and complexity of the case.
- **Witness and Interpreter:** Witnesses are heard during the proceeding and an interpreter is furnished if required by a witness or defendant
- **Juror:** In courtrooms the jury for criminal trials consists of up to 16 jurors which may become to 18 jurors for special proceedings courtrooms. The jury for civil trials consists of 6 to 12 jurors.
- **Spectator:** The right to a public trial necessitates a certain volume of general public seating which varies depending on the public interest of the argument.

2.4.2 Courthouse adjacency Relationships

Another important aspect on which is defined a standard, is related to the relationships of adjacency of the rooms present in a court, in fact, according to the use destination of an environment, we can identify the logical relationships that will make the conduct of activities possible and easy. In order to illustrate the relationship between the spaces, there are the adjacency diagrams; these diagrams are not intended for floor plans that reflect actual configurations of rooms, but are intended to present the functional relationships as operational objectives.

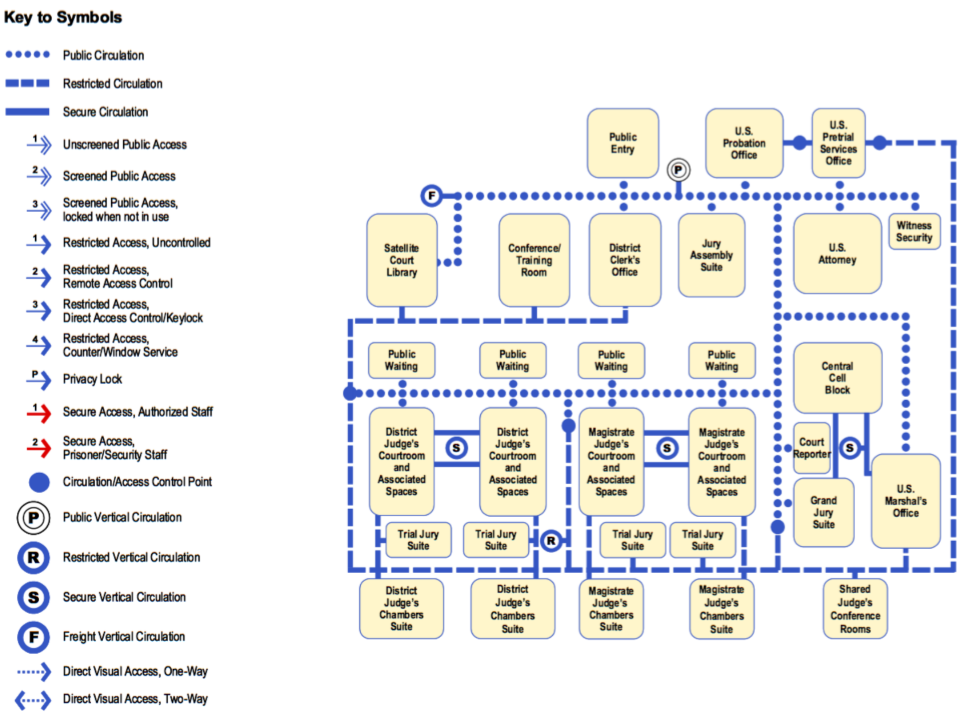

Figure 2.1. U.S. District court adjacency relationships

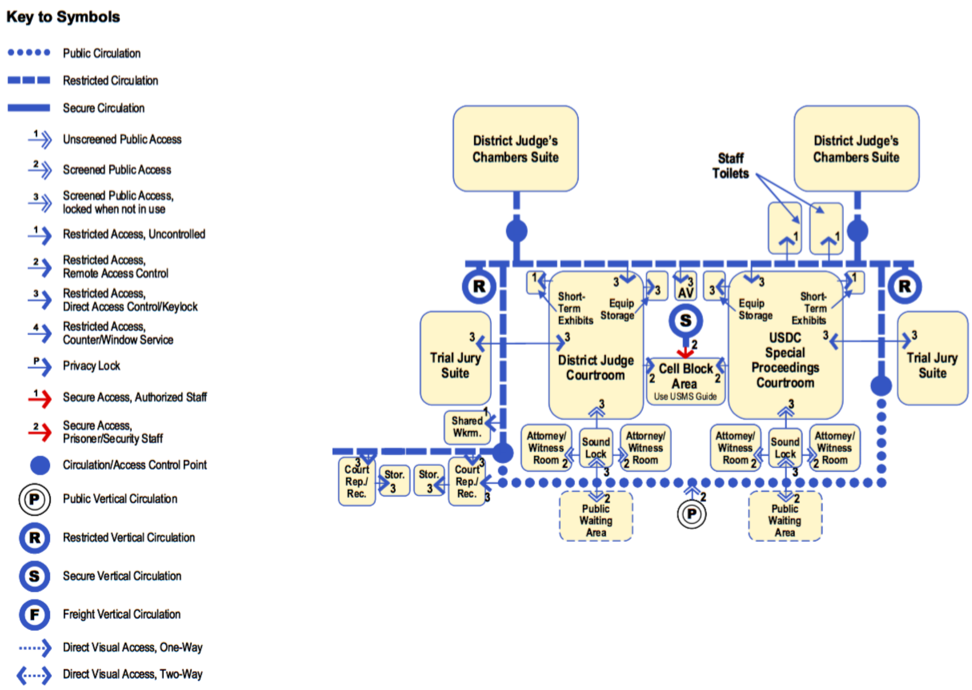

Figure 2.2. Court adjacency relationships detail

2.4.3 Courthouse circulation flows

The identification of circulating flows used for the public, judges, court staff and prisoners who are held in custody and their parting is an essential feature for achieving an efficient and secure movement of people throughout the building. To comply with the requirements related to the circulation flows, the Palace of Justice is to be divided into areas that meet operational needs, physical characteristics and similar access requirements. This issue is extremely important for the proper functioning of federal buildings, in particular, of the courts. The correlated validation process is one aspect that must be addressed during the design of the building, in that, the identification of a problem of this type after it will lead to a significant increase in costs and, in some cases, the impairment of the whole project.

As consequence, the identification of a circulation problem must be done during the early planning stages so that the organization of space within the Palace of Justice are still in a flexible state.

We therefore have three conceptual stages for the development of an architectural project:

- Identification of all spaces in the proposed design.
- Modeling geometry and functions.
- Organization of circulation spaces and levels of security.

Checking the validity of the conditions for the circulation is done by software that performs the comparison of two hundred standards, in most cases, they refer to high-level functional spaces such as the courtroom; the control is done for each identified by the rule spatial instance.

For example, in chapter 7, page 21, line 17 of the "U.S. Courts Design Guide" there is the following rule: **The jury must not pass through the area of public places to sit in the courtroom when moving to or from the trial by jury.**

This rule during the verification of the validation tool is composed of three rules:

1. A-1 The jury does not have to go through the public area of the district court when moving to and from the following jury trial.
2. A-2 The jury does not have to go through the public area of the court of bankruptcy when moving to and from the following jury trial.
3. A-3 The jury does not have to go through the public area of the court of magistrates during moving to and from the following jury trial.

To be able to support interoperability between multiple applications validated by 3D-4D BIM programs, the checking of the validation conditions uses IFC as a neutral file format for the modeling of a building. All instruments validated by the National 3D-4D BIM program have the ability to export its model of a building in IFC format.

Space modelling

An appropriate use of BIM in the definition of space is fundamental, this aspect appears from the attribution of a proper name by a text string and a unique identification number. In a courthouse, as in other public buildings, it is reasonable to have more spaces with the same function, for example the presence of more courtrooms. This common factor necessitates to distinguish the level of types and instances spaces, because depending on the destination use of a given space, it is possible to define a type or class and the presence of more environments of this same class is possible with the definition of multiple instances of the same. Once a type level space has been defined, you can then identify the actors and the amenities of the room with the resulting classification in terms of security. In relation to the type of movement which will take place in that environment we have defined three levels of security:

- **Public Circulation:** refers to all those areas used by the general public inside the building, public circulation is subject to security checks located at the main entrance.
- **Reserved Circulation:** access to areas affected by this type of circulation is via a controlled internal access, the use of these areas is restricted to judges, court staff and official visitors.
- **Safe Circulation:** affects all areas associated with the movement of prisoners; the control of the "Marshals Service" US (USMS) is present.

Modeling Requirements in a Courthouse

For the definition of the spaces, the used standard is IFC 2x2 or 2x3. One of the basic methods of identification of a particular space is the definition of its border conditions, practically, this is done by placing the walls along the perimeter of the area concerned; in it you can identify subdivisions into functional sub-areas.

The definition of a single level of security and a limited circulation may not be trivial in the multifunction areas subject to different types of use, these case studies are still treated as spaces characterized by a single level of security also regarding any movement limited. All other spaces containing more than one security level are considered functionally as a set of subspaces therefore the modeling of these types intended for a multifunction movement do not require special techniques.

In cases where the building has a number of levels, you must also deal with the aspect relating to the vertical circulation via stairs, ramps and elevators.

Security in a courthouse requires coordination of architectural design, allocation of a dedicated staff and appropriate security systems. Basic solutions of architectural type security issues are an integral part of the planning and building design. The use of design patterns regarding separation in areas in public circulation, restricted and safe is essential for security, also in the creation of a court is required to consider the possibility of future expansion and satisfaction of housing needs short and long term. The cost of the architectural solutions integrated during the planning, design and the first embodiment along with the operating and maintenance costs of the security systems is certainly lower than that of the dedicated staff projected on the entire life of the building.

Number Key

Public 1: Unscreened access (uncontrolled entrance to the building and spaces)

Public 2: Screened access (public passes through a security checkpoint)

Public 3: Screened access (locked under certain conditions)

Restricted 1: Uncontrolled access

Restricted 2: Remote access control (as in CCTV and electric door strikes)

Restricted 3: Direct access control

Restricted 4: Counter window service

P: Privacy lock

Secure 1: Authorized staff only

Secure 2: Security staff and prisoners

	ADJACENCY RELATIONSHIPS (For explanation of numbers, see Number Key)				SECURITY & ALARM SYSTEMS											
SPACES	Public	Restricted	Secure	REMARKS	CCTV Camera	CCTV Monitor	Duress Alarm	Security Intercom	Door Contact	Card Reader	Walk-Through Metal Detector	X-Ray Machine	Remote Door-Release Mech.	Emergency Power	Emergency Lighting	REMARKS
Assembly Area		1														
Equipment Storage		3		Adjacent to jury assembly area												
Supplies Storage		3		Adjacent to jury clerk's office												
TRIAL JURY SUITE																
Soundlock Entry		3														
Jurors' Toilets		P		Off soundlock												
Coat Closet		1		In soundlock												
Trial Jury Deliberation Room		1														
Service Unit		1		In jury room												
Exhibits Display				In jury room												
Video Equipment				In jury room												
Sworn-Jury Custodian Station		1		Off entry area			■									
GRAND JURY SUITE																
Entry Area	3	3														
Jurors' Toilets		P		Off entry area												
Service Unit																
Security Workstation		1		Off entry area			■									
Witness Room		1														
Hearing Room		3	1													
Exhibits Display				In hearing room												
Video Equipment				In hearing room												
Closet		1														

Figure 2.3. Table extract on security and access controls

2.5 Requirements elicitation and analysis

The approach that we propose will feature a formalization of the analysis on the nature of the requirements, the choice of the most correct way of application of the courthouse, the study of the dynamics of a general court until finally arriving, identification some requirements and assets.

When you wish to delve into the issue of the requirements analysis in the context of operational environments, is soon evident that there are two different types of approach. [9] Static analysis focuses on the latent characteristics of a given project of the building, it is characterized by the definition of invariants based on topological relationships between entities such as direct access from one room to another or the distance between two rooms. The focus on those static properties of structural models is an appropriate formalism for expressing structural constraints and invariants. In contrast, the dynamic analysis focuses on changes in the environment as a result of the interactions between people, goods and environments that characterize a building and that can lead to the violation of certain requirements. A good example of a violation of a constraint subject to the dynamic analysis, is the limited access control to a computer containing data characterized by a confidential content.

Performing requirements gathering in the courthouse of Pavia, an area characterized by a multitude of more or less complex dynamics such as those present may not be an easy task, so we follow a contextual approach to investigation.

2.5.1 Contextual Inquiry

The contextual inquiry [10] is a semi-structured interview method aimed at obtaining information on the context of use; users are first asked a series of standard questions and then observed and interrogated while they work in their environment. Since users are interviewed in their environment, requirements analysis is more realistic than a derived analysis from pre-designed deductions. The contextual inquiry is based on a set of principles that enables an application in various fields. This technique is generally used at the beginning of the design process and it is a good tool to get rich information on employment practices, social relationships, physical environments and user tools. The four principles of this methodology are:

1. **Focus:** Inquiry planning based on a clear understanding of the purpose.
2. **Contest:** Direct observation of the dynamics related to the client's activities.

3. **Interaction:** Dialog with the clients about their activities with the aim of snatching the most hidden issues.

4. **Interpretation:** Development of a shared understanding with the client about the most important aspects of their activities.

2.5.2 Requirements domain analysis

Initially, we focused on the acquisition of the knowledge necessary for an approach in this field, the understanding of the dynamics linked to court proceedings was one of the first issues we tackled. Processes can be of two types depending on the area of the arguments discussed, in less serious cases related to problems between people, we proceed with the civil trial whereas in severe cases in which the crime is recognized as a criminal one, then we proceed with this type of procedure. Based on the number of judges employed in the conduct of the proceedings we can talk about monocratic processes or collegial. In addition, there is a further classification that divides the proceedings in open court when people can attend, and council chambers where the process is restricted to stakeholders.

In a similar way as we did in the presentation of the US standard, we identify the actors who take part in such proceedings. In criminal proceedings we can detect the presence of a chancellor who will have the task of creating a report of everything that happens during the proceedings, the accusation personified by the prosecutor who acts as the State's voice defining the charge against the accused, the defense identified by the lawyer defender, the accused present inside a cage, the civil party composed of an additional attorney and the injured party may be present, and finally we have the police who have the task to supervise and ensure order in the courtroom. In a civil trial, however, the prosecutor is not present whereas both parties attend next to their lawyers. The distinction between a monocratic and a collegial process is due to the fact that in the first there is a single judge, while in the second there are three judges. In the event of a council chamber, only persons authorized by the court may attend, whereas, in case of a public hearing whoever wants to attend, simply present his ID card to gain access to the classroom.

2.5.3 The identification of stakeholders

Proceeding with an analysis of the organizational structure of the Palace of Justice, we were able to identify the parties involved in the project with relation to their role; these subjects were identified in court summits. As responsible for order during the proceedings, the transport of prisoners and general security inside the building, we looked at the police as a stakeholder. Finally, as responsible of the renovation and expansion of the Palace, we take into account the Superintendency who also collaborated with us by promoting the gathering of requirements.

The court vertex

We have identified three figures inside the Palace of Justice as the principal organs of the organizational structure:

- **The Court President** is the magistrate in charge at the head of a particular court that, pursuant to Legislative Decree No. 160/2006 takes his judgmental function of First Instance. He has organizational tasks and other functions given to him by the law, he is present in both the civil and criminal procedural law. In the criminal context, first of all the President has organizational tasks, during the process he manages and coordinates the running, in some cases he may intervene by suggesting questions to the parts or by asking them directly. The term of position runs from a minimum of four years to a maximum of six through a renewal request that requiring approval.

- The Procura della Repubblica represent the offices of Pubblico Ministero at a courthouse, this office have a vertex figure personified in **The Chief Prosecutor**. He is the promotor of the penal action, he manages his office with organization of activities and he exercises the functions attributed from the law to Pubblico Ministero. The Pubblico Ministero leads the investigation using the police and he must undertakes a criminal proceedings everytime he has knowledge of a crime.

- **The Supervisory Magistrate** "head" is a monocratic organ with powers related to the execution of the sentence. He performs supervisory activities on Penitentiary Institutes and monitors that the implementation of the treatment of the prisoner is in conformity with the principles enshrined in the constitution and penitentiary law, through visits and hearing of the prisoners. The Supervisory Magistrate has wide powers to intervene when a prisoner reports a complaint. For this reason, the law compel the magistrate to frequently visit the jails and hear all inmates who ask to talk.

Police

There are two types of Police: **Judical Police** and **Security Police**. The Judical Police is the set of deputies police offices to conduct judicial investigations. He has the administrative function to ensure the conditions for the prosecution, identifying crimes, by searching for the perpetrators and collecting evidence to be used in criminal proceedings. The judicial police are supporting the Pubblico Ministero in the conduct of investigations, it has law- enforcement purposes, in fact, it intervenes when a crime has already happened. Instead the Security Police tends to prevent the fulfillment of crimes, she is responsible for defending a civil society which are public order and the security of individuals.

Ministero Infrastrutture e Trasporti - Provveditorato OOPP

In this case, the Provveditorato has the role to support the activities related to development of the infrastructure, with the aim of doing modification for the increment of security standards; its jurisdiction is interregional.

2.5.4 Composition of contextual interviews

Once the stakeholders have been identified, we have created a series of questions for each of them, the idea was to get answers and investigate any issues that would emerge as a result during the interview.

Questions for the court vertexs

The questions forwarded to the court members were more related to the functions performed by the actors involved in the dynamics of the Palace of Justice and the functioning of the latter environments. The topics were focused on the functions of each court room, on the identification of the most important assets and their location. With regard to the conduct of the proceedings, we have explored the possible movements of the actors involved, among which we find judges, clerks, lawyers, defendants and members of the public; furthermore, we also sought to understand how to handle the procedural material.

Questions for the Police

With regards to the police, we discuss about issues related to security monitoring and which actions to perform for its maintenance. Our questions were focused on the agent's reaction during a proceeding of natural or intentional disasters, such as, a fire, an earthquake or a shooting; an interesting aspect of one of these disasters concerns the need to deal with the accused person in an appropriate housing. Continuing focusing on the court proceedings, other important aspects are related to the controls carried out on the courts, on lawyers, on the employees involved and the public. We posed the questions regarding the transport of prisoners trying to understand how many police cars are involved in the transportation, how many agents are in charge of guarding and escort the inmate, the checks carried out before introducing the prisoner inside the court, the nature of the trips made to gain access to the courtroom, the relative timing and possible reactions to the occurrence of a disaster during this flow of operations.

Questions for the Provveditorato

To the Provveditorato, consistent with the role played in this project, we posed questions about the building in its structure and on equipment supplied that can be used to ensure security. Inherently to the access control, we discussed which devices were present in the courthouse in order to carry out checks on the people entering the courthouse, with particular attention to the prisoners. Then we treated more generally the issue of the security devices relating to the distribution inside the building and the ability to monitor and prevent the occurrence of dangerous situations.

2.5.5 Security Scenarios

During the composition of the questions, we tried to focus on what could be a valid use scenario on which to contextualize the questions in the process of interaction with stakeholders and focus, in general, our analysis of the security requirements. The use case would have to present, in addition, characteristics which should enable it to demonstrate the effectiveness of the application of a specific methodology with the purpose to monitor and prevent the concretization of dangerous situations. The scenario that we have identified to be the right one for this type of requirement is the transport of prisoners.

Prisoner transport

One or more prisoners have to access the courtroom, so they will need to be checked before entering in the building to verify that they don't possess weapons of any kind, on the way to avoid dangerous and malicious acts towards people and objects and in the courtroom during the proceedings. Moreover, one or more spectators, relatives of the accused or prisoner may be present, they will have to be kept separate from any prisoner and controlled at the entrance, along the way and in the courtroom.

The Actors involved

In the transportation use case of the prisoners we can identify a number of stakeholders, among which we find:

- **Prisoner** as the protagonist of the scenario since the entire flow of operations and the figures of other actors involved concern around him.
- **prisoner's relative** he constitutes a factor to not neglect as he represents an external contact for the prisoner and could interact with him in respect of his interests.
- **policeman** is the actor who aims to monitor the prisoner transport operations, and prevent the realization of a dangerous situation.

2.5.6 Adjacency diagrams

Inspired by ideas identified in U.S. standards, we have also decided to apply the concept of adjacency diagram to the courthouse slightly reformulating the breakdown of security degrees that, we remind the reader, in the American case it was composed of public, private and secure circulation. In this aspect we have received great cooperation from the Provveditorato, the degrees of security defined in the end are specialized based on the type of user interested in a given area:

- Restricted access to the internal personnel administration
- Restricted access to Prosecutor staff
- Restricted access to internal and external staff (lawyers)
- Restricted access to the public

For security reasons, the diagrams are not included in this paper.

2.5.7 Assets and security requirements

From the collection of information it has emerged a set of assets and security requirements that constitutes a necessity for a courthouse.

Assets and requirements relating to their custody

The important assets to a court are IT tools and the files of cases, the former are kept in the Registry administrative, civil and criminal but also in the rooms of the magistrates that contain sensitive data. The files are stored in the chancellor offices in closed cabinets, during proceedings in the judges rooms in the case of civil proceedings and in court in case of the criminal; this applies, in general, to all the procedural material. Once the judgment has been defined, the files reside in the magistrates rooms until the judge formalizes his sentence. In addition, another asset of highest relevance, is interception tapes that are kept in the vaults.

Requirements related to actor movements

At the entrance of the Palace of Justice, there is a check through a metal detector operated by the supervisory personnel who work, also, to identify the individual. In each hearing, many cases are scheduled with a variable duration. The total duration of each hearing is different, usually, civil hearings without the presence of a registrar ends before 2pm but there are cases in which the appearance of the parties is expected in the afternoon. In criminal hearings the presence of a chancellor is required for the preparation of the verbal, so the timing will respect service hours contemplated by law and in labour contracts. In principle, criminal hearings last from 9:30am to 5pm, but if necessary, they can proceed even beyond. Throughout the hearing, the judge or the court, in case of collegial hearings, sit at their reserved desk in the classroom. To deliberate during a civil trial and GIP hearings, the judge dismisses the audience, or he withdraws into the adjoining chambers courtroom located near the rooms of hearings of criminal trials. During any breaks the judge or the court may leave the room or classroom venue of the hearing; this also applies to the figure of the chancellor. In the civil sector and in GIP hearings, lawyers access to the court room only when they have to participate in the proceedings in which they conduct their defensive mandate; they wait in the corridor that the procedure is called. During the audience they sit in front of the judge. During the development process, the parties sit at the desk reserved for them next to the one intended for P.M. and they stand up when they have to carry out activities which fall within their mandate such as the examination of texts and/or defendants, speeches etc. They approach the judge's bench to produce court filings. They get to talk to their clients in cases where, for reasons of space, these are not sitting next to them. The defendants are seated next to their lawyer both in the GIP room and a public hearing room and in order to exit, they have to inform the judge. They can be sent away if, with their behaviour, they disturb the smooth conduct of the hearing. The public is only present at hearings that take place in the classroom and takes place behind a fence separating the area to the Parties from the public. The public can not intervene in any way, and if disturb arises, the judge may order their removal by the police. An agent who is monitoring a room where a hearing is held, in case of alarm, will follow the procedures of the evacuation in the courthouse. In the event that he has in custody or is accompanying a prisoner to attend a hearing, he shall comply with the provisions of current evacuation procedures in the courthouse, and he also has to care for the security of the detainee and his interactions with other people present building.

2.5.8 Prototypical static and dynamic security requirements

To structure the information gained from the interviews, we classify the security requirements gathered according to their nature. With regard to static requirements, that are closely linked to the structural design of the building, it was found that in the operations flow that occur during the process, you can locate the functional need to have a direct and secure link between the courtroom and the hearing room of counsel in criminal proceedings where the judge retires to deliberate. Thus, we define the security requirement belonging to the static class:

R1: *In the design phase, it is necessary to arrange the courtroom and the council chamber in such a way as to ensure a secure connection between the two rooms.*

From the dynamic point of view, that is related to the interactions between people, property and the environment in general, take for example a requirement inherent in the use scenario of the prisoners transport. In the first place, it is necessary to consider that inside the building there are some important assets, during the movement of the prisoner, suitably escorted to the courtroom, there is the need to maintain the flow of circulation isolated from any assets in order to ensure the absence of any potentially dangerous contact. Our aim is to calculate a path where there is no interaction between the transport of the prisoner and the assets, then we ask what is the probability associated with that event. Thus we can define the dynamic nature of security requirements:

R2: *During prisoner transport to the court proceedings hall, no critical assets must be encountered on the way; the asset is a generalized entity that can be a physical asset or a person that must not be encountered, such as a relative.*

R1 and **R2** are two prototype security requirements that reflect more general cases that occur in practice.

Chapter 3

BIM as the domain model

BIM is a relatively young technology in an industry typically slow to adopt the changes, and yet, among those who have recently adopted this tool is widely thought that BIM will grow until it will play an increasingly crucial role in the documentation of the construction of a building. Proponents of this technology claim that BIM offers various advantages such as the display improvement, the increase of productivity due to easy retrieval of information, a better coordination of the construction documents, the increase of speed delivery, the reduction of the costs and moreover the embedding and linking of important information such as suppliers for specific materials, the location of details and the estimate of required quantities. Furthermore, BIM contains most of the necessary data to the realization of an analysis of energy performance, in fact, building properties can be used to automatically perform energy simulations useful to verify the adequacy of the building to the charge standard and cost savings.

3.1 Building Information Modeling

You can define the Building Information Modeling [11] as the process of development, growth and analysis of multi-dimensional models of the physical and functional characteristics of a building that are generated digitally through the use of special programs. Currently, BIM software is used by professionals, companies and government agencies to plan, design, build and maintain several physical infrastructures such as schools, offices, factories, bridges ...

The concept of BIM exists since the 70s, and for 30 years was the subject of several academic papers, including that of G.A. van Nederveen and F.P. Tolman who in 1992 used for the first time the term "Building Information Model", however, only in 2002 it reached some interest thanks to the release from Autodesk of a paper entitled "Building Information Modeling" and also other companies began to get involved in this field. The transition from Model to Modelling is crucial as BIM is no longer proposed only for the modeling of a building but also as a project sharing instrument with the aim to promote interoperability, a semantic enrichment of content and the reduction of errors design. In 2005, the General Services Administration of the US government announced that all provider companies would have to include a BIM model as part of their design services.

The role of BIM in the construction industry through its actors such as architects, engineers, surveyors, builders or clients, is to support communication, cooperation, simulation and optimal improvement of a project along the full cycle of life of the built work; the latter is defined from the design stage through the implementation phase, up to the use and maintenance. The BIM model offers a view that goes beyond the two-dimensional and three-dimensional building, aiming to ensure the elimination of project inconsistencies present. A BIM project can contain any information regarding the building or its parts, including the most commonly collected we find the geographical location, geometry, material properties/components/systems and the technical elements, stages of implementation, the maintenance and disposal at the end of the life cycle.

Due to the complexity of the information that is taken into account when working on a construction project, some companies have developed software designed specifically to offer a BIM framework that facilitates the design and sharing of information. Among the different software packages available for this purpose, there are, for example, Bentley AECOsim Building Designer, ArchiCAD, MagiCAD, Autodesk Revit that differ from the architectural design tools such as AutoCAD as they allow the addition of further information about production details, cost, sustainability beyond the mere representation of the parts of the model. ArchiCAD was considered the first implementation of BIM as the first CAD product available on a personal computer can create drawings in 2D and 3D.

3.1.1 BIM adoption

The adoption of BIM is definitely on the rise, the use of digital models for virtual design, construction and collaboration is becoming a standard and key markets around the world supported by governments are moving more and more towards this new technology.

United States

In the US, BIM has been used since the beginning of the millennium, in 2003, the General Services Administration (GSA) has established the National Programme for the 3D-4D BIM [12] publishing guides describing the methodology of work in the construction industry. The basic idea was that the power of visualization, coordination, simulation and optimization of 3D, 4D and BIM information technologies would allow GSA to meet more effectively the customer, the design, construction, resource management, simplifying the management and program requirements. The GSA also requested that as of 2007, BIM should be used for the "Spatial Program Validation" before presenting the project to tender, this allows design teams to simplify the validation of space requirements by encouraging the use of this technology in all phases of the project.

United Kingdom

The British government has identified the importance of an industrial policy in the construction industry as a key to the UK growth. [13] The adoption of BIM was encouraged by the development of the National BIM Library, a digital library of free objects and easily accessible online by all professionals in the construction sector. The message for companies implies that those with the confidence of investing resources for the adoption of this technology will end up with an edge over others in the future. The Construction Industry Council (CIC) has immediately made its contribution to the development and conducting this program with the government, the plan is to assist companies to adapt to the new standards and make BIM required for the realization of public works from 2016.

Europe

In northern Europe, in Norway, BIM has been used increasingly since 2008. Several of the largest public clients require the use of this standard in open formats in most of their projects. [11] At the center, in France, there are examples of organizations that promote the use of BIM such as the Fédération Française du Bâtiment while in Germany in December of 2015, the German Minister of Transport announced a timetable for the introduction of mandatory BIM for road and rail projects in Germany since the end of 2020.

Italy

In Italy, BIM is spreading but not in a systematic manner as it lacks today a uniform methodology and open-minded. The spread of BIM, in fact, has been more successful in realities where there are major engineering companies or integrated design firms that have a greater advantage in the entire project management. The lack of widespread dissemination of BIM in Italy may be due to the fact that the design of the market is composed of small businesses, so if they are not constrained by specific laws, they have difficulty in investing more resources in a new technology. In Italy there is currently no legislation that regulates or promotes the use of BIM although the new Procurement Code [14] Article 23 comma 13, contains the transposition of European directives which encourage the use of BIM in the planning for public works.

3.1.2 Issues related to interoperability

The fact that this technology offers an opportunity to foster interoperability between designers and engineers lays the foundation for the presentation of a problem related to the allocation of responsibilities for the choices made during the design and construction of a building. In this context, in fact, the work product liability mingles among the employees, as consequence, it is a non-trivial task the association of a product part to a single person. There are no laws that guide the risk breakdown [15] in a BIM project, this concept of shared design is still too new and its supporters argue that such a model is the best evidence of an appropriate design when it is used to defend the work of architects, engineers and others who have contributed. Some argue that a shared fee necessarily implies the sharing of risks, this aspect should be well settled from the beginning and agreed on by architects, engineers and anyone else who contribute to the project, otherwise, if you ever have a problem, the change of a dispute arising is very high. A further problem may develop related to copyright as some of the choices, solutions and techniques used could be reused by one of the members who participated in another project. In 1990 the US Congress approved the Architectural Works Copyright Protection Act (AWCPA) that provides protection to architectural works from the point of view of copyright linked to the overall shape, design solutions, the composition of space and design elements. In some states, the right to the protection of designs is recognized in the absence of a formal procedure linked to copyright. A good part of the contractual documentation is intended to protect the professional design from the reuse of the owner of the work in the absence of an approved agreement. Apply all these considerations to the BIM is complex and involves a good preliminary stage with the aim of defining all these delicate aspects, it is important that each project participant maintains a track that allows being able to isolate his contribution; there is not a normative guide for that yet. The attribution of guilt regarding a possible design error is definitely not an easy task, despite this, the advocates of BIM believe that it improves communication between designers and customer by promoting the progress of the project and its fulfillment.

3.1.3 BIM and security

The construction security is an issue at the national and global level that has not yet found a permanent solution. [16] There are several contributions offering solutions that involve the application of automated security rules on BIM models. The proposed algorithms analyze the model of a building in order to detect security risks and suggest possible preventive measures to designers in different cases of danger related to the collapse. The spread of BIM in the AEC/FM industry [17] is changing the mode of approach to security, with the introduction of this methodology you can pinpoint problem during the preliminary stages of design and planning that can then be removed easily ensuring higher standards of security and health. In Finland, the VTT Technical Research Centre has developed a manual procedure for using BIM in the security, planning, management and communication fields. In 2010 the Institute of Technology of Georgia introduced the first preliminary results of an automated controller for the security rules for BIM models. The literature shows that BIM allows virtual controls that can be used to identify potential security risks, the idea is to exploit the formalization of security rules together with the spatial representation; this will allow efficient identification of problems at the design stage.

Control rules approach

One of the most promising applications of BIM in the AEC industry is facilitating the monitoring of security rules and carry out various simulations to evaluate construction projects in their initial stages. This process consists of five stages:

1. **Rule interpretation**: For interpretation of the rules, we mean the process of translating a given standard from human speech to a format understandable by the machine. The two main aspects of this procedure are generally summarized in identifying the conditions for the application and definition of the properties on which the rule applies; for example, the first aspect may identify a room as the target object and the second one identifies rules about its size and position.

2. **Model preparation**: In object-based modeling, all construction objects should be associated with a specific type of object and its attributes; this concept is the basis of geometric elements in BIM. A key requirement for a control system based on rules is that each object construction must possess the information as a unique identifier code, name, type, attributes, relationships, date and the author of creation element model. In addition, the spatial structure of the objects should be well organized, for example, via a subdivision for floors or sections, this helps to classify the spatial constraints more easily.

3. **Rule execution**: The execution phase of the rule set comprises the use of rules with the building model. Thanks to the fact that the rules have been transformed into code understandable by the machine, their execution is simple. This procedure has two phases, the first consists in the automatic control of the model by applying the security measures while the second has the goal of providing all the possible solutions available to the implementation on the basis of another set of contextual rules. The execution of the rules is repeated to identify potential new risks after implementing one of the possible solutions to the first threat identified.

4. **Report on the rule control**: The results of the security checks are reported in two different forms, one from the point of view of the protection devices used whereas the other uses a table to show in detail the applied solution.

5. **Corrective actions**: Human decision makers may use a three-dimensional representation of the environment to help applying preventive methods. BIM improves decision making and increases awareness of the project

participants. The main corrective action that will take place on construction sites is related to planning and tracking of logistics movements of the material used.

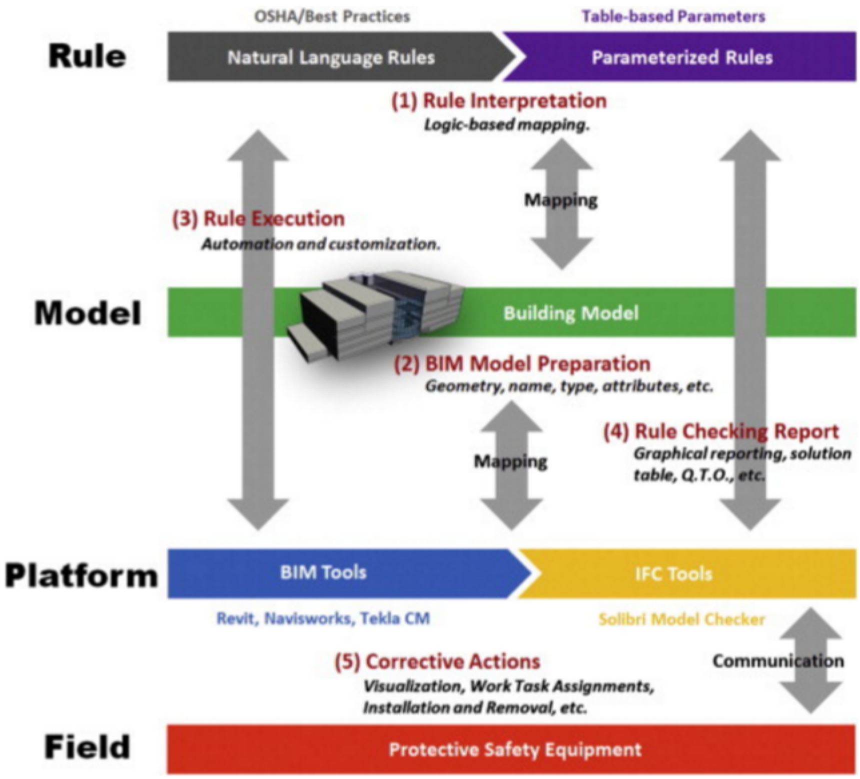

Figure 3.1. Process of control rule application

Changes in the operational environment as analysis object

The security issue is widely spread as a function of the structural soundness of the building, although this is an aspect of fundamental importance, beyond this stage, a different dynamic comes into play. The first aspect relates to the definition of the environment in terms of the set of rooms in relation to their intended use, in fact, a suitable distribution of the spaces encourages the correct functioning of the building both from the operational point of view and from that of the functional security. The second aspect is related to the need to have a system capable of supporting the security officers in building control. Changes in the operational environment determine the transition from one state to another and hence which set of control rules must be active to allow the verification of the security conditions; moreover, in cases where there is a violation, it determines which corrective actions will be necessary in order to deal with the exception.

3.1.4 BIMLight as a BIM subset

BIM provides a rich representation of the construction elements both in terms of geometry and of attributes and relationships. Despite its widespread adoption, BIM is not expressive enough to represent the entities and properties of a building that might be relevant for security analysis, for example, BIM models do not include an explicit representation of digital assets and materials as confidential file or server. In addition, BIM is perceived as too rich and heavy for security analysts, because, detailed structural properties that might be relevant to the architects could not be obtained from the analysis of security support. Due to these limitations, BIMLight[2] is used, a lightweight version of BIM standard which is compatible with it and supports the security review. To achieve this goal, the entities and topological properties that appear to be relevant for the security analysis and in particular for access control have been analyzed.

Relevant entities

BIMLight offers the representation of entities such as floors, rooms, walls, doors and windows, which were already present in BIM standards but it also includes additional entities, relevant from the point of view of security, that are not fully supported by BIM. In the approach to the problem of the security in the Palace of Justice, we focused on some of these other entities:

- When **agents** are considered, we must take into account possible malicious intentions towards physical or digital resources.

- The **access control management** allows us to identify areas or resources to which an agent has a right to access. This system is based on the association of certain functions to an agent and it has the aim to prevent potential malicious agents from getting access to valuable assets.

Relevant properties

To support the security analysis there are three fundamental properties:

- **Containment**: You may take into consideration the containment in terms of placement of a physical asset within a room or area or as the positioning of IT assets, such as files, in a calculating device; this last case is the subject of the extension of the containment concept within BIMLight.

- **Proximity**: The interest in the concept of proximity lies in the need to detect when one or more agents are in the same environment where an important resource is placed. This property can be inferred by the analysis of the containment relationships, therefore it does not need an explicit representation.

- **Reachability**: Similarly to the containment properties also reachability has a dual nature in this area. In physical terms, a room is reachable from another if there is a passage which establishes the connection, whereas, from a software point of view, two devices can be reached if there is a network connection to which both are connected. For this reason, in BIMLight network connections are also represented.

3.1.5 BIM Software

The list of certified software from buildingSMART IAI contains a fair number of programs which offer several BIM implementations able to support architects and engineers during the design. Among the most significant companies: Autodesk, Nemetschek and other members of the group of the latter as Solibri and Graphisoft appear.

Autodesk Revit

Revit[18] is a single application that supports a BIM workflow, starting from concept up to the actual construction. This program can be used to accurately model the projects, optimize performance and promote efficient interoperability. This software offers several features including:

- **Parametric Modeling**: In Revit, parametric components are the basis for the design of all components of the building, they favor the design intent at increasingly higher levels of detail forming an open, graphical system for design and creating forms.

- **Real-time sharing**: Worksharing enables multiple team members to work on the same project simultaneously in a shared central model, this represents an important aspect of interoperability between different disciplines team. Sharing in Revit provides a full range of modes ranging from access simultaneously to the shared model to formal division of the project into distinct and separate shared units, to complete separation of project elements into individually managed linked models.
- **Coordination uninterrupted**: The ability to store all information in a single coordinated database allows you to make revisions and modifications which are then propagated throughout the model. The Revit parametric change engine automatically coordinates changes executed anywhere in the model views, drawing sheets, sections and plants.

Solibri Model Checker

Solibri Model Checker[19] is a software solution for monitoring the quality of BIM, it undertakes analysis from the architectural and engineering point of view to provide integrity, quality and physical security. In addition to this, Solibri Model Checker includes the functionality for the extrapolation of information, analyzing and verifying the information available within BIM models. Solibri Model Checker aims to reduce design errors, actively contributing to saving on costs of construction projects, in a more effective modeling and quality assurance. The outcome of the analysis carried out on BIM models reveals potential flaws and weaknesses in the design, it highlights the contrasting components and checks that the model complies with the requirements. Also, an add-on is available which allows interoperability between Solibri Model Checker and ArchiCAD ensuring that these two software are fully integrated. The building models can be easily transferred from ArchiCAD to Solibri Model Checker and thus be controlled in several ways. During the analysis, Solibri Model Checker automatically assigns a classification of the severity of the reported design problem, finally, the problems encountered in the project are directly returned to ArchiCAD for their clarification and correction; all these features make the fluid workflow, easy and productive.

Graphisoft ArchiCAD

ArchiCAD[20][21] is an architectural BIM program developed by the Hungarian company Graphisoft initially until 1995 under the name of Radar/Ch. Development began in 1982 for the Apple Macintosh, and it became a famous product for that platform, it is recognized as the first CAD product on a personal computer able to create both 2D and 3D drawings. In 2011, about 150.000 professionals use it in industry of building design, ArchiCAD is now the leading software solution in the Building Information Modeling (BIM), used by architects, designers, engineers and builders to be able to personally design, document and collaborate on architectural projects.

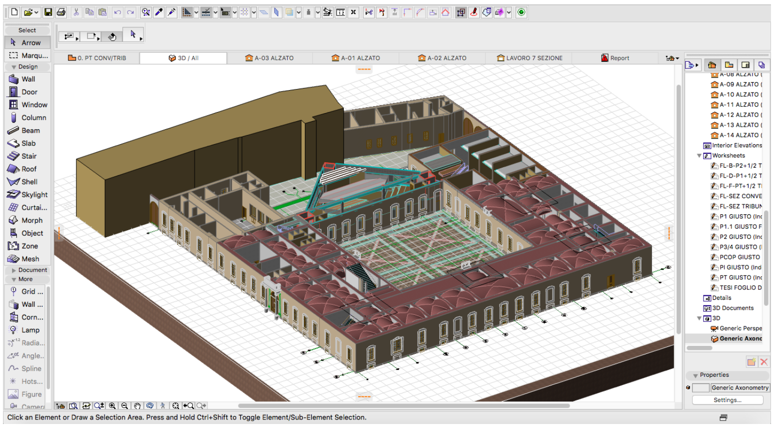

Figure 3.2. Section of the Palace of Justice in Pavia in an ArchiCAD project.

ArchiCAD allows the user to work with objects embedded with parametric data to create a virtual building using real structural elements such as walls, floors, roofs, doors, windows and furniture. The program comes with a wide variety of pre-packaged customizable items that the user can create autonomously, either by using the primitive elements of the program or by using the GDL language. Plans, sections, elevations, lists of materials and other processed are generated directly from the program based on the three-dimensional model of the building, and are updated in real time. ArchiCAD saves projects in PLN and can export their models in various formats, including DWG, DXF IFC and SketchUp. Among the new features offered by ArchiCAD there are:

- **Background Processing**: In order to optimize response time, ArchiCAD manages to exploit the unused computer resources to anticipate user actions, providing views in the background on which it is not operating. This new generation technology sets it apart from a traditional BIM software.

- **OpenGL 3D Navigation**: The OpenGL engine optimization allows ArchiCAD to make a 3D navigation project, smoother and faster project.

- **Support for Point Cloud**: ArchiCAD unables to import all common disc formats for point clouds measured with 3D laser scanning instruments and to generate the corresponding 3D geometry instantly. This speeds up the time required to reconstruct the state of a building and the project stages.

3.2 Industry Foundation Classes

Requirements related to interoperability in the AEC/FM industry has necessitated the definition of standard formats based upon the implementation of BIM, for the exchange of information concerning the design of a building.

3.2.1 Origins and development

In 1994 the development of Industry Foundation Classes[22][23] gave officially the start to the definition work of a standard for an open data model, favoring the interoperability for using BIM. In the same year, the originator company of this work, Autodesk, formed an industry consortium to steer the company towards the development of a set of classes `C++` that could support the development of integrated applications; twelve companies joined it. In September 1995, the activities began, and the consortium, initially called "Industry Alliance for Interoperability", extended the possibility to adhere to all interested parties. IFC 1.0 was released in January of 1997, this version had a very limited scope and was basically focused on the architectural model of a building. Spread globally, the association of companies changed its name to "International Alliance for Interoperability" established as an organization driven by the non-profit organization, with the aim to publish IFC in the AEC industry as a carrier neutral model of a building's life cycle. In the same year the first version was improved up to the 1.5 and in July 1998 the first commercial implementations of BIM software were released supporting IFC 1.5.1. Simultaneously to the development of the first version of IFC, the 2.0 was launched with the release in 1999 by incorporating new information schemes regarding building services, cost estimating and construction planning. IFC 2.0 had greatly increased the flow of information supported by the standard, therefore, the IFC 2x version was primarily released for increasing the stability which occurred in 2000. In 2003 IFC 2x2 was released which wore a further increase of information supporting the standard, a 2D spatial geometric model, the analysis of structural details and support for verification of building codes, etc. In 2005 the Association changed again its name, the IFC specification is now developed and maintained by buildingSMART IAI [24]. The 2x2 version was followed by the 2x3 one, released in 2006 and it made substantially stable the changes introduced, in 2013 the latest IFC version 4.0 came out, whose development had two central goals:

introducing quality to the speed and gaining full international standardization.

3.2.2 The technology

The exchange of BIM data is dominated by proprietary solutions as many construction projects are based on a solution in which all employees have software from the same vendor or derive from any compatible way.[23] Proprietary solutions predominate despite the fact that, since the beginning, the industry has been working on specifications for an open data format open with regards to the technological maturity of BIM software. IFC could potentially bridge the connections between stakeholders and project phases in a typically fragmented project environment of the construction industry, however, it is important to highlight that IFC does not provide intrinsic interoperability, it depends on how the software packages are interfaced with it. Most modern BIM platforms support the import and export of IFC data models while buildingSMART certifies applications complying with the standard.

The STEP stardard, the EXPRESS language and the BCCM model

The need for a standard for the exchange of data models was recognized since 1950, the turning point that brought at the start of the development of STEP (Standard for the Exchange of Product Model Data)[23] was in 1984, when the ISO TC184/SC4 ISO subcommittee declared that none of the existing formats could be extended to meet the need of an open modeling standards to the computer for various industrial and manufacturing sectors. To get an idea of how much such development had spread, it can be noted that AEC/FM was only one of several areas included for the standardization within STEP. Thus began an effort geared to the development of a language that later became known as EXPRESS[25], this language was originally developed in conjunction with STEP to define standard data models; relationships, attributes, constraints, and inheritance defined for a domain of interest are fundamental concepts of EXPRESS. An Express data model can be defined in two ways, textually and graphically. For formal verification and as input to tools such as SDAI (Standard Data Access Interface), the textual representation of an ASCII file it is the most important. The graphical representation, however, is often more suitable for human usage, for example for an explanation or a tutorial, defined EXPRESS-G, it is not able, however, to represent all the details that can be formulated in textual form.

The data model of the example defines an EXPRESS schema called Family, it contains the supertype entity Person and two subtypes Male and Female, Person is declared as abstract (ABS), so only occurrences of subtypes Male

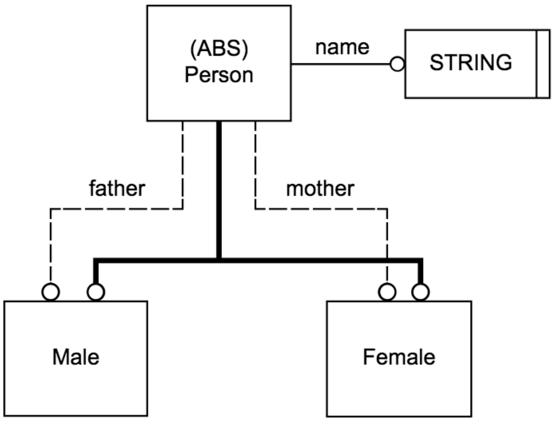

Figure 3.3. An EXPRESS-G diagram for the schema of a family

or Female may exist- Each occurrence of a person has a compulsory name attribute and two optional father and mother attributes, on these last there is a well-defined reading style: a Female can play the role of a mother for a person and, similarly, a Male can play a father role.

```
SCHEMA Family;

ENTITY Person
   ABSTRACT SUPERTYPE OF (ONEOF (Male, Female));
     name: STRING;
     mother: OPTIONAL Female;
     father: OPTIONAL Male;
END_ENTITY;

ENTITY Female
   SUBTYPE OF (Person);
END_ENTITY;

ENTITY Male
   SUBTYPE of (Person);
END_ENTITY;

END_SCHEMA;
```

Figure 3.4. Code for the family EXPRESS data model

Within a schema with various data types, structural constraints and algorithmic rules can be defined. A main feature of the EXPRESS language is the possibility to formally validate a population rich of data types by verifying the presence of all the structural and algorithmic rules. In December 1994, the initial STEP version became an international standard: "ISO10303:1994, Industrial Automation Systems and Integration - Product data representation and exchange". In the AEC/FM industrial sector, the ambitious effort to aggregate and harmonize the most important aspects of the existing work, the development of a high-level data model with universal concepts, became the central data model taking the name of Building Construction Core Model (BCCM).

IFC

The requirements that were identified as a study subject for a standard model for the open construction, such as the containment of all the information of the building, meeting the needs of all interested parties, the entire cycle support life of the product, the ability to have multiple levels of abstraction, the need to have an independent software and in an independent format, were integrated into IFC that is in fact the culmination of this research area.[23] Putting together parts of the standard STEP, by incorporating the BCCM model concepts, basing on the modeling language EXPRESS and definitions for geometric representation was a good starting point for the implementation of IFC although the task of composing a strict but flexible data model, able to contain and represent products and process data that would meet the needs of an entire sector was not an easy task. It provides a standardized data structure for storing information on the buildings but not force or ability in any way specific implementation in software; many ways are possible and it is up to software developers make a decision about it. EXPRESS schemas containing IFC data can be encapsulated into files to be exchanged or the IFC data structure can be represented in an object database and be updated remotely via Internet. Essentially most of the BIM software provides the end user interface via which the project can be saved or exported and IFC is one of the possible standards that can be selected to store the template data in parallel with the data format owner.

Supported formats

IFC data files are exchanged among applications in one of the three available formats[22]:

- **IFC-SPF** is a textual format defined by ISO 10303-21 ("STEP-File"), where each line, typically, consists in a single object and it has ".ifc" as extension. This is the most spread format as it has the advantage of a compact size as well as a text easy to read.
- **IFC-XML** is an xml format established by ISO 10303-28 ("STEP-XML"), it has ".ifcxml" extension and it is suitable to interoperate with xml

instruments and for the exchange of templates that define portions of a building. Because of the large size it is the least used.

- **IFC-ZIP** is a compressed zip format containing an IFC-SPF file with ".ifcZIP" extension.

3.2.3 IFC Structure

The structure of a IFC data model is divided into four software layers: domain, interoperability, nucleus and resources.[23]

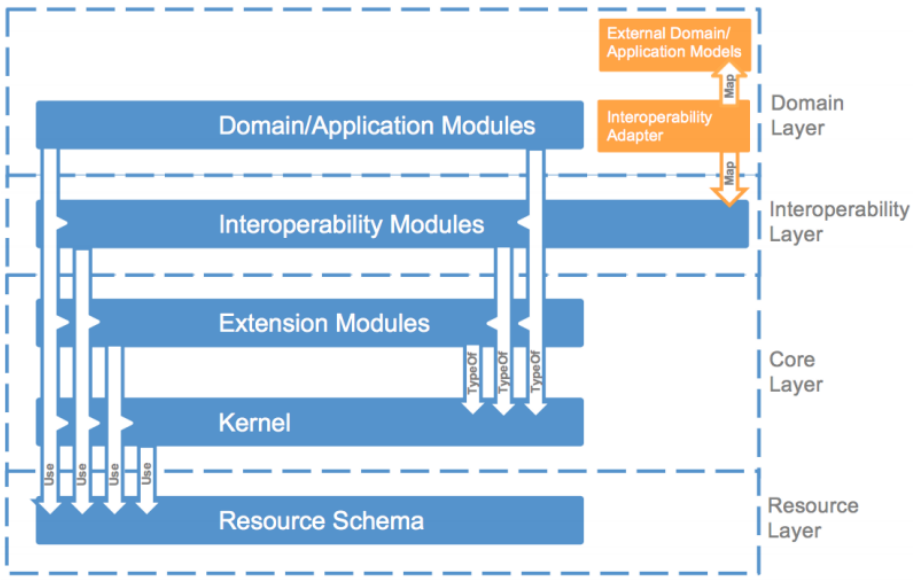

Figure 3.5. IFC data model structure

- The domain layer contains domain models for processes in specific AEC domains or types of applications, such as architecture, structural engineering, and HVAC, among others.

- The interoperability layer provides the interface for the domain models by implementing an exchange mechanism to allow the interoperability among domains.

- The nucleus is made up by the kernel and extension modules, the first one determines the model structure and decomposition, providing basic concepts about objects, relations, type definitions, attributes and roles, whereas the extension modules are specialization of classes defined in the kernel.

- The resource layer represents the schema of resources containing the basic definitions which are used to describe objects in the higher layers.

These levels represent hierarchies closely referenced, the main rule says that a reference can only be directed down the hierarchy. This means that data in the resource layer must be independent and have no references towards higher classes, instead, the other layers may have references in the lower ones; references within the same layer are allowed only within the resource layer. From a syntactical point of view, an IFC file defines entities such as objects with a set of attributes in which a unique identifier is assigned to each object that is used when referring to it. Each row in the data section defines a new object, wherein the identifier is given in the left side of the equal sign and the right side defines the type of object followed by a set of unordered attributes. Each attribute can be a string, a reference to an object, such as ”#10”, or a null value ”$”.

```
#13= IFCORGANIZATION('GS','Graphisoft','Graphisoft',$,$);
#14= IFCAPPLICATION(#13,'19.0.0','ArchiCAD-64','IFC2x3 add-on version: 3003 INT FULL');
#15= IFCOWNERHISTORY(#10,#14,$,.ADDED.,$,$,$,1461437109);
```

Listing 3.1: IFC Language Example

Any entity in the kernel or in a subsequent sub-level of the IFC model inherits the entity **IfcRoot** which is the abstract class, it assigns a globally unique identifier (GUID), the ownership information and history for the entities, optionally even a name can be stored for ease of use by users or software systems and a description provided for the exchange of remarks. [24]

There are three types of fundamental entities in the IFC model, all of which are subtypes of IfcRoot, they form the first level of specialization within the IFC hierarchy of the classes:

- **IfcObjectDefinition** is the generalization of any thing or process, both in the case of a type and in the case of an occurrence. The definitions of objects inherit the name attribute that should be a label recognizable by users, further details of the object can be specified in the attribute description inherited.

- **IfcPropertyDefinition** defines the generalization of all characteristics (a grouping of single property) that can be assigned to objects, property definitions define information shared among object instances.
- **IfcRelationship** is the generalization of all possible relations among objects in IFC. This allows to keep as generic the properties directly inherited leaving open the possibility to specify afterwards a different behaviour. There are two different types of relations, 1 to 1 and 1 to many which are exploited in the subtypes of IfcRelationship.

For each of the fundamental entities, we are going to focus on some subtypes which are particularly significant for the definition of the model of a building.

IfcObject Entities

IfcProduct: It can represent any object or support useful to define, organize and annotate an object that refers to a geometric or spatial context. The subtypes of IfcProduct usually have a representative form and a local position within the project structure.
In addition to physical products and spatial elements, IfcProduct also includes non-physical elements that relate to geometric or spatial contexts, such as the grid, annotation, structural measures, etc.

IfcSpace: Subtype of IfcSpatialStructureElement, a IfcSpace, represents an area or a volume concretely delimited or delimited at a theoretical layer on the basis of features within a building.
A space, if specified, is associated with the floor of a building and can span multiple connected spaces, therefore, a space group provides a collection of spaces included in a plan. A space can also be broken down into parts, where each part defines a partial space. Among the various fields of a IfcSpace we find the unique name, a description containing further information that the user may have specified, an identifier that contains the full name of the space, and a field that contains the type of space, that is typically the functional class of space.

```
#16586= IFCSPACE('34hq_Bcev00f8ZXGGrldWq',#15,'1',
   $,$,#16466,#16583,'UFFICIO 5',.ELEMENT.,.
   INTERNAL.,$);
```

Listing 3.2: IfcSpace Example

IfcElement: Generalization of all components that constitute an AEC product, IfcElement is a subtype of IfcProduct. These elements can be logically contained by an element of the spatial structure and they constitutes a certain level in a structural hierarchy of the project. The elements may be physically existing objects but also empty elements, such as holes; they can be assembled on site or pre-manufactured and constructed on site. Examples of elements in a building of buildings context are the walls, floors, windows and doors.

IfcWall: The wall represents a construction that delimits or divides the spaces, it is generally vertical or almost, it is a planar element and often designed to withstand structural loads but it is not required to be a load-bearing wall. IfcWall is a subtype of IfcBuildingElement and the IFC specification also provides another entity for walls, IfcWallStandardCase which is in turn a subtype of IfcWall. IfcWallStandardCase is usually used for all occurrences of walls that have a fixed thickness along the entire route and where this thickness can be described by a set of layers of materials, while IfcWall is used for all other occurrences in particular for the walls in which the thickness changes or that are not vertical.

```
#13494=IFCWALLSTANDARDCASE('07vZ1G03rFL9wlij362dTQ
   ',#15,'MURO 423',$,$,#13322,#13490,'07E63050
   -003D-4F54-9EAF-B2D0C60A775A');
```

Listing 3.3: IfcWall Example

IfcSlab: A slab is a construction component which encloses one normally vertical space, it can provide lower support such as the floor or upper floor covering as in any of the spaces present in a building. It should be noted that only the core or constructive part of this construction is regarded as a slab. A particular type of slab is the landing, described as a section of the floor in which stairs or ramps act as a link.

```
#18648= IFCSLAB('0pKr6vSgTB2vkpAR8w_518',#15,'
   SOLAIO 174',$,$,#18620,#18645,'335351B9-72A7-4
   B0B-9BB3-29B23AF85048',.FLOOR.);
```

Listing 3.4: IfcSlab Example used as floor

IfcDoor: The door can be defined as a construction element for closing an opening, mainly destined for the access with a zipper, pivoted or sliding operation. Its entity, IfcDoor, defines a particular instance of a door inserted in the spatial context of a project.

```
#13267= IFCDOOR('0YtizV9ej7hxlfzuycIgnu',#15,'
   PORTA 046',$,$,#13255,#13264,'22DECF5F-268B-47
   AF-BBE9-F78F264AAC78',2100.,900.);
```

Listing 3.5: IfcDoor Example

IfcWindow: Construction for the closure of a vertical opening in the vicinity of a wall or roof which admits the passage of light and allows the entry of fresh air. The IfcWindow entity defines a particular instance of a window placed in the spatial context of a project.

```
#10893= IFCWINDOW('1bi09LIhH0CgeJhdZRBgKM',#15,'
   FINESTRA 002',$,$,#8689,#10890,'65B00255-4AB4
   -4032-AA13-AE78DB2EA516',2530.,1160.);
```

Listing 3.6: IfcWindow Example

IfcFeatureElement: Generalization of all the elements that change the shape and aspect of the main element to which they are associated. IfcFeatureElement offers the ability to manage the shape modifiers such as semantic objects within the model to IFC objects. The concept of functionality introduced by this entity defines the primary element with subordinate parts as additions or with empty/clippings as spatial subtraction.

IfcOpeningElement: It represents a vacuum inside an element that owns physical manifestation. Openings must be managed by all sectors and disciplines in the industry AEC FM, therefore interoperability for the opening elements is provided at a high level. IfcOpeningElement is a specialization of IfcFeatureElementSubtraction that offers the ability to manage the shape modifiers within the IFC model in order to steal a space portion of the main element.

```
#8643= IFCOPENINGELEMENT('38nkoEQwU$cRwyeT7abfrF'
   ,#15,'FINESTRA 002',$,$,#8508,#8640,'C8C6EC8E-6
   BA7-BF99-BEBC-A1D1E4969D4F');
```

Listing 3.7: IfcOpeningElement Example

IfcPropertySetDefinition Entity

IfcElementQuantity: It is defined as a series of measurements derived from the physical properties of an element. The elements may form part of the spatial structure such as buildings, floors and spaces or be construction objects such as walls, floors, and finishes. The optional MethodOfMeasurement attribute defines the code, for example by a standard measurement method, which was used to calculate the amount of the element. IfcElementQuantity has as an attribute a set of IfcPhysicalQuantity subtypes that contain the extracted measures, in the case of IfcPhysicalSimpleQuantity subtype we have a set of entities related to measurements of length, area and volume, for example IfcQuantityArea and IfcQuantityVolume.

```
#16600=IFCQUANTITYAREA('GrossFloorArea',$,$
    ,23.7278431108);
#16609=IFCQUANTITYLENGTH('NetPerimeter',$,$
    ,20682.0841402);
#16610=IFCQUANTITYVOLUME('GrossVolume',$,$
    ,156.841042963);
#16612=IFCELEMENTQUANTITY('1WxSgn5Vmv_b0Jjx$sYOOM'
    ,#15,'BaseQuantities',$,'ArchiCAD BIM Base
    Quantities'(#16600,#16601,#16607,#16609,#16610)
    );
```

Listing 3.8: Example of IfcElementQuantity and of some correlated IfcPhysicalQuantity.

The physical quantity, IfcPhysicalSimpleQuantity, is an entity that contains a single quantity value together with a semantic definition of use for the value of the magnitude. The attribute name must characterize the individual quantities, and the attribute of the measure unit, if specified, indicates that the data value is a quantity in that unit. Alternatively the global measurement of the unit definition will be used, IfcUnitAssignment, which contains a set of IfcUnit. Each definition of units must be unique, so there will not redundant be definitions of the units for those of the same type, such as unit of length, unit of surface area, etc.

```
#16= IFCSIUNIT(*,.LENGTHUNIT.,.MILLI.,.METRE.);
#17= IFCSIUNIT(*,.AREAUNIT.,$,.SQUARE_METRE.);
#18= IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
#32= IFCUNITASSIGNMENT((#16,#17,#18,#22,#23));
```

Listing 3.9: Exampe of IfcUnitAssignment and of some contained IfcUnit

IfcRelationship Entities

IfcRelSpaceBoundary: The border defines the physical or virtual demarcation of a space as a relationship with the surrounding elements. IfcRelSpaceBoundary is a subtype of IfcRelConnects and it is defined as an object that represents the relationship between an item and the space it delimits. It is given as a one to one correlation but allows each constructive element to define many relationships and to each space it allows to be defined by as many reports of this type. In the case of a physical border, where the IfcPhysicalOrVirtualEnum attribute assumes the "PHYSICAL" value, the positioning and shape of the contour can also be given providing the reference to the present construction element whereas in the virtual case the attribute is "VIRTUAL" and there is no reference to a border element. The field InternalOrExternalBoundary defines whether the border is inside (INTERNAL) or external (EXTERNAL), that is adjacent to an open space that can be partially closed, such as a terrace.

```
#32037=IFCRELSPACEBOUNDARY('0bwCn75PMk_fdAr$kV0C1Q
    ',#15,'2ndLevel','2a',#16586,#13494,#32036,.
    PHYSICAL.,.INTERNAL.);
```

Listing 3.10: IfcRelSpaceBoundary example

In the example 3.10 #13494 the wall 423 defines a physical and internal border on the office 5, #16586, #32036 is the IfcConnectionGeometry's id representing a 3D area of connections between two objects.

IfcRelDefinesByProperties: Defined as a specialization of IfcRelDefines, it represents the relationships between a set of properties and objects. Properties are aggregated into a set, sets of properties may be in turn be grouped to define an object type. IfcRelDefinesByProperties is a one to many relationship as it allows the assignment of a set of properties to a single object or to many, then these objects share the same property definition. The following sample is an example of the set of properties assignment 3.8 with id #16612 all'IfcSpace example 3.2 with id #16586.

```
#16614=IFCRELDEFINESBYPROPERTIES('3
    A8fb3uKGihndrVCqWCFu',#15,$,$,(#16586),#16612);
```

Listing 3.11: IfcRelDefinesByProperties example

IfcRelVoidsElement: A relationship is defined between one element of a building and an opening that creates, therefore, an empty element.
As it is a subtype of IfcRelConnects that, in general, link objects according to some criteria, IfcRelVoidsElement is a relation of one-to-one that implies a boolean subtraction between the geometric bodies of the element and the opening. In the sample that follows we find the relation between the item with id #8488 and the opening of the code example 3.7.

```
#8646=IFCRELVOIDSELEMENT('3tlrwrThM8OUWIqt0R4Mqk'
    ,#15,$,$,#8488,#8643);
```

Listing 3.12: IfcRelVoidsElement example

IfcRelFillsElement: Another IfcRelConnects's specialization is IfcRelFillsElement that defines a relationship between openness and the constructive element that fills it, it is one-to-one relationship. In the sample we find a relationship between the opening of the example 3.7 and the code window 3.6

```
#10896=IFCRELFILLSELEMENT('3ND1Nw1n_0SDbhGf0r$FnE'
    ,#15,$,$,#8643,#10893);
```

Listing 3.13: IfcRelFillsElement example

3.2.4 IFC implementations

IfcOpenShell

IfcOpenShell[26] is an open source library that helps users and software developers to work with the IFC. IfcOpenShell internally uses the library oce (acronym of opencascade community edition) to convert the implicit geometry in the IFC file in an explicit geometry, in order to make it compatible with any CAD software or modeling package.
Oce [27] is a library `C++` for 3D modeling, it can be used to develop CAD / CAM programs such as FreeCAD or IfcOpenShell. IfcOpenShell is still under development, it currently supports only files with the extension SPF-IFC, in the future the support for XML-IFC and IFC-ZIP will be added.

IfcPlusPlus

IfcPlusPlus[28] is an open source model classes in language `C++` which can also be used to read and write files in STEP IFC format. Among the main features, it is worth mentioning an easy and efficient memory management through the use of smart pointers and reading skills, exploiting parallelism, guarantees fast analysis on multi-core processors. There is also an application for displaying IFC models based on the use of cross-platform graphics libraries such as Qt [29] and OpenSceneGraph [30]. It can be used as a starting point for all kinds of applications that implement a construction model IFC.

IFC TOOLS Project

IFC TOOLS Project[31] provides a framework for access and viewing of BIM models based on IFC, developers can easily integrate libraries into their products and end users can use the viewer to analyze their models. This project is a division derived from the no longer supported OPEN IFC TOOLS, this separation took place in 2010 and since then has resulted in a complete overhaul of the goals. The framework provides the ability to quickly develop applications based on IFC in order to support complex business processes in the construction industry.

xBIM

The Toolkit xBIM[32] is an open source tool for the development of BIM software that supports the IFC data model, because it allows developers to read, create, and view information on BIM models in IFC format; this tool provides comprehensive support for geometric and topological operations as well as for display. In other words xBIM can be used to create BIM middleware tailored for applications based on IFC. xBIM is extensible because the source code is available for anyone who wants to participate in the project in order to extend and add functionality to the code but also because it exploits the characteristic extension methods of the C#, a native programming language xBIM, which allows to extend existing classes without having to change their source code. In xBIM solution there are eight `C#` projects and one in `C++`, the implementation was done on the .NET platform and is compatible with ASP.NET. The xBIM potential is expressed in the implementation of xBIM Xplorer which allows users to:

- Open and view IFC files.
- Navigate through a model to display the detailed information of the building, such as the roof thickness, the number of ports, etc.
- Filter the view of certain parts of the building.
- Display a 360° model.

- Export IFC files in .ifcXML and .ifcZIP formats.

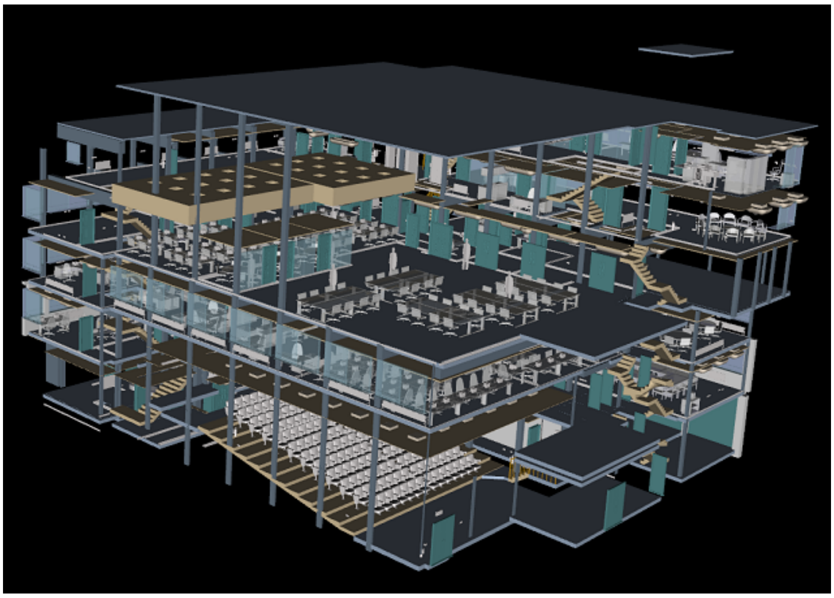

Figure 3.6. Displaying a 360° model in xBIM Xplorer.

IFC Engine DLL

IFC Engine DLL[33] is a Toolbox STEP able to generate 3D geometry for the most popular versions of the IFC model, the component is able to create, upload and modify files in IFC and IFC-SPF-XML; It is compatible with IFC2x3 and IFC4. Among the most important features of IFC Engine DLL, we find the excellent performance in terms of speed and scalability, compatibility with the three most widely used operating systems in the world in both the 32-bit version and in the 64-bit and reliability of the merged geometry kernel.

Chapter 4

Enabling Analysis – Model-to-Model Transformation

We need to obtain analyzable models from IFC descriptions – this is our overall goal. To do this, we exploit the structure inherent in BIM/IFC to generate a graph representation of relationships inherent in the design. This is in contrast to taking into account geometry in IFC descriptions, an approach adopted by other works. We formalize the process as a model-to-model transformation, by defining mappings between the classes of the source and target models and related constraints; to fulfill the first we will make use of some class diagram while for the second will exploit OCL as a formal specification language. All this will be accompanied by an running example to allow an easier understanding of the process of abstraction by the use of metamodels and transformation that we will implement.

4.1 Transformation Overview

The concept of transformation of a model in another will be useful since, to take advantage of the implementation of BIM, we must obtain an export via the IFC model from ArchiCAD project. The obtained model should be transformed into another one, useful in order to perform security analysis. To make the transformation, we will abstract the IFC model, thus obtaining a metamodel that through the use of a mapping will create a new relational metamodel containing the various components of the project and their connections; finally, by reducing the level of abstraction we will define the final model that will have the aim of allowing a simpler exploration.

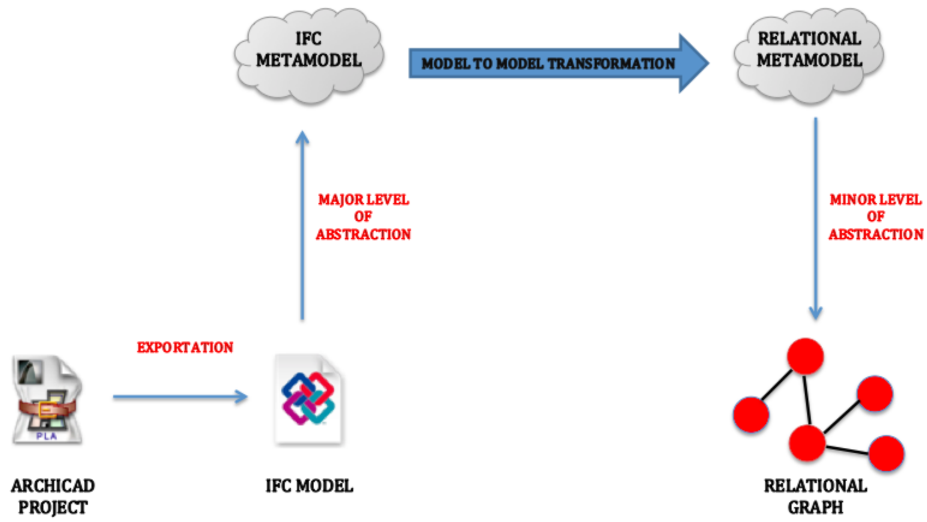

Figure 4.1. Graphical representation of the entire transformation process.

4.2 Transformation Assumptions

Our initial goal is deriving an IFC model from an ArchiCAD project, keeping in mind that this model has to be useful for the type of analysis we are interested in, we assume that in the modeling done by the architects, the concept of space is present as it will be useful to exploit border relations with the constituent parts so as to have a complete view of an environment in three-dimensional terms. Other assumptions of fundamental importance refer to the production of a careful design in geometric terms in order to ensure that incorrect connections between the architectural components of the building in the final model are not generated and to the enrichment of the project with as much information as possible as they prove to be helpful in order to achieve a good analysis. The idea is to obtain a model which allows to identify a room, such as a polyhedron with volume, delimited by walls that has doors or windows, from the floor and from the ceiling. The relationships with respect to other rooms will be consequently calculated, as the existence of an object of common border will define the eventual adjacency and reachability.

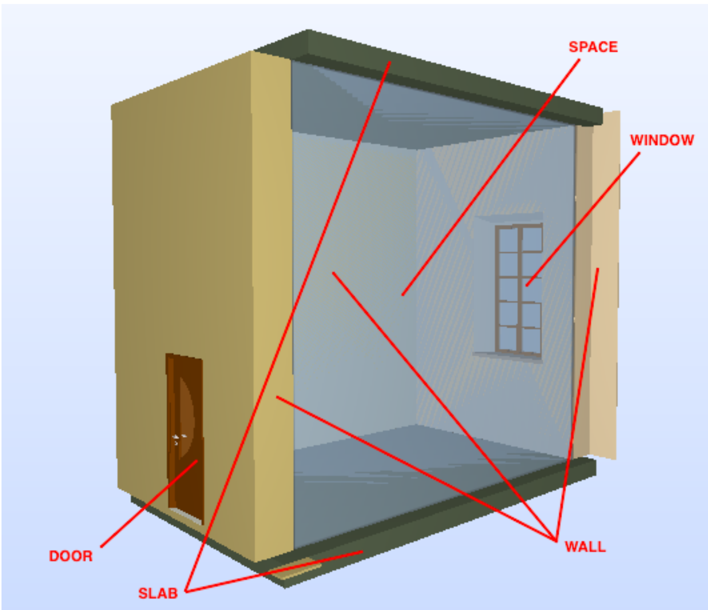

Figure 4.2. Model of a room where border elements have been identified

4.3 Model to model transformation

Basically a model is a simplified abstract view of a complex reality. The abstraction is a reduction process of the information content of a concept in order to maintain only the relevant information for a particular purpose. The transformations that have the goal of getting a model from another model are a useful tool to promote interoperability; we can identify two categories:

- **Horizontal transformation:** in this case the level of abstraction of the source model and the destination is the same.
- **Vertical transformation:** the source model is at a different level of abstraction than that of destination.

To create transformations of this kind is necessary to specify the models or metamodels of origin and destination and to create the statements of one or more mappings that define the correspondence relationships of the belonging elements. The statements contain matching rules that define the relationships between the characteristics of the elements, working on this level of abstraction allows to focus on the problem domain.[34]

4.3.1 Partial model of an office design

For practical reasons, given the considerable size of the building, we decided to use a partial model as running example throughout the chapter to showcase how the m2m transformation works. In this model there is a room described by its volumetric space, the floor and a wall containing a window.

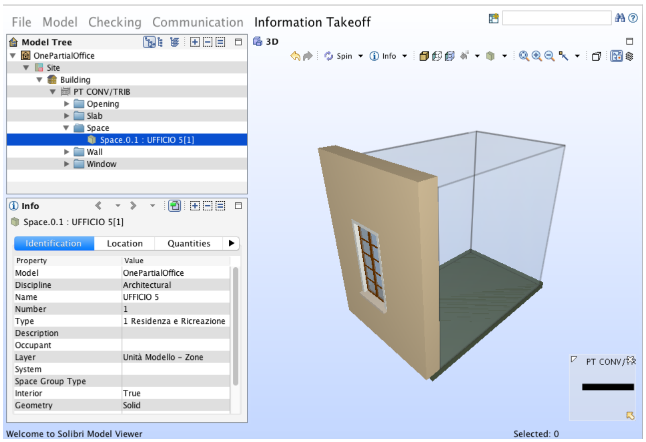

Figure 4.3. Partial model of an office viewed with Solibri Model Viewer.

4.3.2 From the ArchiCAD project to the IFC model

ArchiCAD provides an implementation of BIM as each object which is instantiated, along with other compound or described geometrically, corresponds to an entity IFC. There is thus the possibility of obtaining an IFC model starting from the ArchiCAD project but to do this you need to build a suitable export configuration.

In ArchiCAD the import and export of IFC models occurs according to the settings of a particular translator, some are already predefined, but you can configure your own. [35]

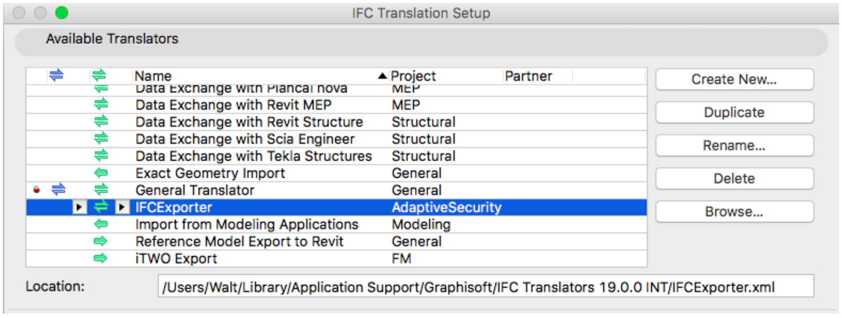

Figure 4.4. Configuration of a translator for the export of a IFC model

The configuration options cover various conceptual aspects useful in order to obtain a model that can transmit the information required for security analysis and at the same time guaranteeing a certain optimization. Among the most significant ones we find:

- The definition of a **domain** of interest for the IFC model, can be extended to all the elements, reducing it to only those structural and mechanical or choose one personalized. Our configuration focuses mainly on IfcSpace as spatial structural elements, the IfcWall, the IfcOpening necessary for the presence of IfcDoor and IfcWindow and finally, as vertical separating elements we consider IfcSlab and IfcRoof. You can also use the structural function to decide whether to consider all elements, or only those bearing or only non-bearing.

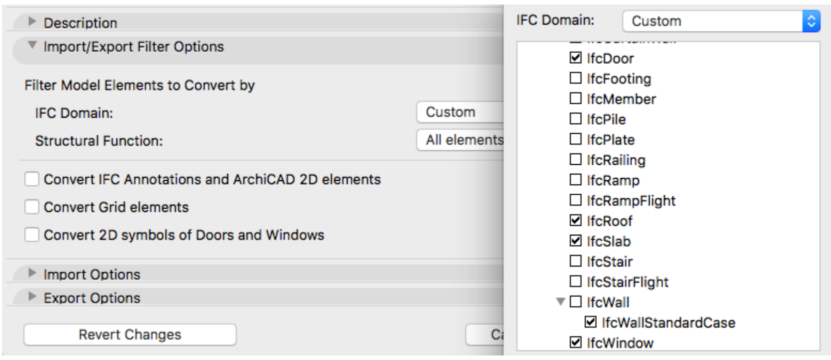

Figure 4.5. Definition of a personalized domanin

- The definition of appropriate **geometric conversion options**, in detail we configured the one on the multi-layer coatings such that each element is logically turned into an IFC entity with assigned its construction material. The exported geometric representation defines the elements as solid delimited by their boundary surface (BREP), the possible alternatives consider only the upper surface or a description with contours and points.

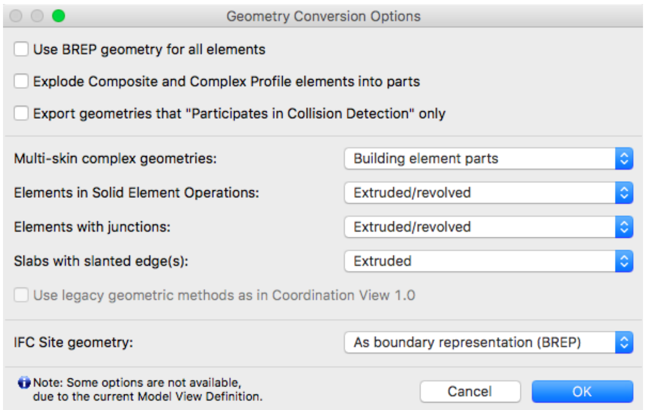

Figure 4.6. Options of geometric conversion

As regards the elements involved in geometric operations between solids, the elements with joints and the flooring with inclined edges, the "extruded/revolved" configuration is the most suitable, as compared to the "BREP" it allows to export the elements while keeping the values of the fundamental parameters such as, for example, the thickness or height without play in an exact manner some special geometries which would be of little significance to our analysis.

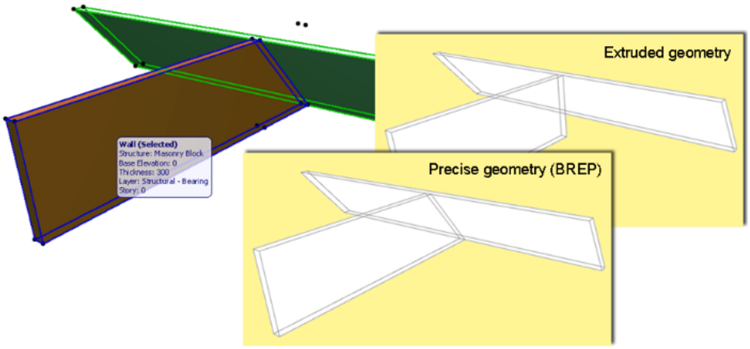

Figure 4.7. Comparison between the configurations with a precise geometry and the approximate.

- The ability to export **additional data** to the elements, which can be useful for exchanging data with energy analysis programs or cost estimate. We have considered interesting to export information regarding the containment relationships of spaces compared to objects of various kinds, such as HVAC or assets of another kind; also data about the geometric dimensions in terms of length, area and volume can be useful to have a clear view of the thickness of the walls and the capacity of the rooms.

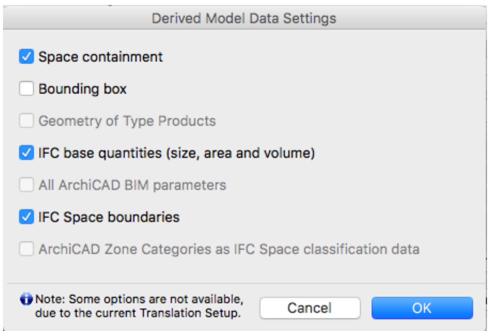

Figure 4.8. Configuration mask for the additional data.

The option on the "IFC Space boundaries" plays a fundamental role as it allows us to model the rooms as three-dimensional solid whereas in normal cases, rooms are delimited by two plates and four walls. With this configuration, ArchiCAD exports the spatial boundaries and their relationships (IfcRelSpaceBoundary) together with the spaces (IfcSpace), that it is, ArchiCAD calculates the position, size and adjacency of the elements that surround each room and divides the rooms according to areas defined and cut by elements connected and openings.

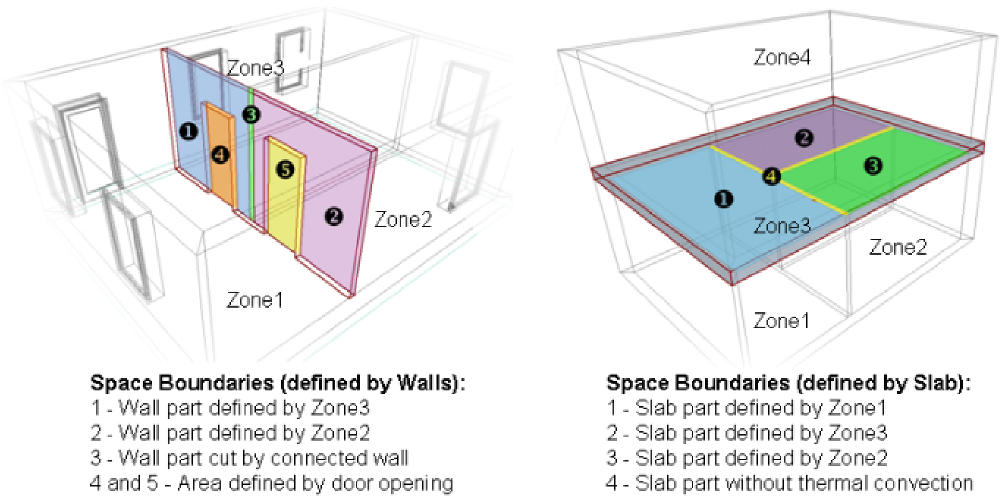

Figure 4.9. Border relationships defined by walls or from the floor slabs.

- Finally you can set the **measure unit** globally for the export of all the geometric properties such as length, area and volume. The option of maintaining **the IFC unique identifier of ArchiCAD** can be a good practice because it allows you to compare two versions of the model through the use of other programs.

4.3.3 IFC model abstraction

Once we obtain the IFC model containing all the necessary information, the idea is to transform it in a different metamodel that is more oriented to our analytical needs.

Towards an IFC metamodel

It aims to achieve a high-level representation that contains all the elements present in the IFC model, highlighting its logical processes in order to draw closer to the domain concepts ignoring, for the moment, the computational and algorithmic parts in a strict sense. We then proceed with an explanatory example which allows to illustrate the process of abstraction.

```
#2944=IFCSPACE('34hq_Bcev00f8ZXGGrldWq',#15,'1',$,$,#2822,#2941,'UFFICIO 5',.ELEMENT.,.INTERNAL.,$);
#2953=IFCPROPERTYSINGLEVALUE('Var',$,IFCLABEL('X'),$);
#2954=IFCPROPERTYSET('01Bdj5P7uKBR8bzOLGFle8',#15,'Pset_SpaceCommon',$,(#2953));
#2956=IFCRELDEFINESBYPROPERTIES('3mgv4qHBd$PBerl8LnD',#15,$,$,(#2944),#2954);
#2974=IFCQUANTITYVOLUME('NetVolume',$,$,137.8085);
#2975=IFCELEMENTQUANTITY('1WxSgn5Vmv_b0Jjx$sYOOM',#15,'BaseQuantities',$,'ArchiCADBIMBaseQuantities',(#2973,#2974));
#2977=IFCRELDEFINESBYPROPERTIES('3iA8uKGihndrVCqWCFu',#15,$,$,(#2944),#2975);
#2765=IFCSLAB('1lpx5eOXUiD99',#15,'SOLAIO 174',$,$,#2735,#2762,'6B168-61B7-46B7-A1DE-8DD8D249',.FLOOR.);
#3012=IFCRELSPACEBOUNDARY('0bMk_fdAr$C1Q',#15,'2ndLevel','2a',#2944,#2765,#3011,.PHYSICAL.,.EXTERNAL.);
#170=IFCWALLSTANDARDCASE('0DTmDtHOfE3BdON4FIN',#15,'MURO289',$,$,#118,#164,'0D770377-458A-4E0C-BB9B-AE761710F497');
#3154=IFCRELSPACEBOUNDARY('0U71yG_Ia5VH6',#15,'2ndLevel','2a',#2944,#170,#3153,.PHYSICAL.,.EXTERNAL.);
#391=IFCOPENINGELEMENT('38nkoEQwU$cRwyeT7abfrF',#15,'FINESTRA002',$,$,#256,#388,'C8C6EC8E-6BA7-BF99-BEBC-A1D1E4969D4F');
#396=IFCRELVOIDSELEMENT('3tlrwrThM8OUWIqt0R4Mqk',#15,$,$,#170,#391);
#2661=IFCWINDOW('1bi09LdZRBgKM',#15,'FINESTRA 002',$,$,#451,#2658,'6555-4AB4-4032-AA13-A516',2.53,1.16);
#2664=IFCRELFILLSELEMENT('3ND1Nw1n_OSDbhGfOr$FnE',#15,$,$,#391,#2661);
```

Listing 4.1: Partial IFC model of an office.

In the extract of IFC code a room is analyzed in volume terms, the room is associated with an attribute whose value can be freely chosen by a user and a geometric quantity that indicates the net volume. Furthermore, we have the floor of the room and one of the walls having an opening that contains a window. The defect of this partial model obtained from an IFC file, is that although it contains the information regarding the room, its attributes, the wall and the floor, it does not allow to effectively display the links between these components.

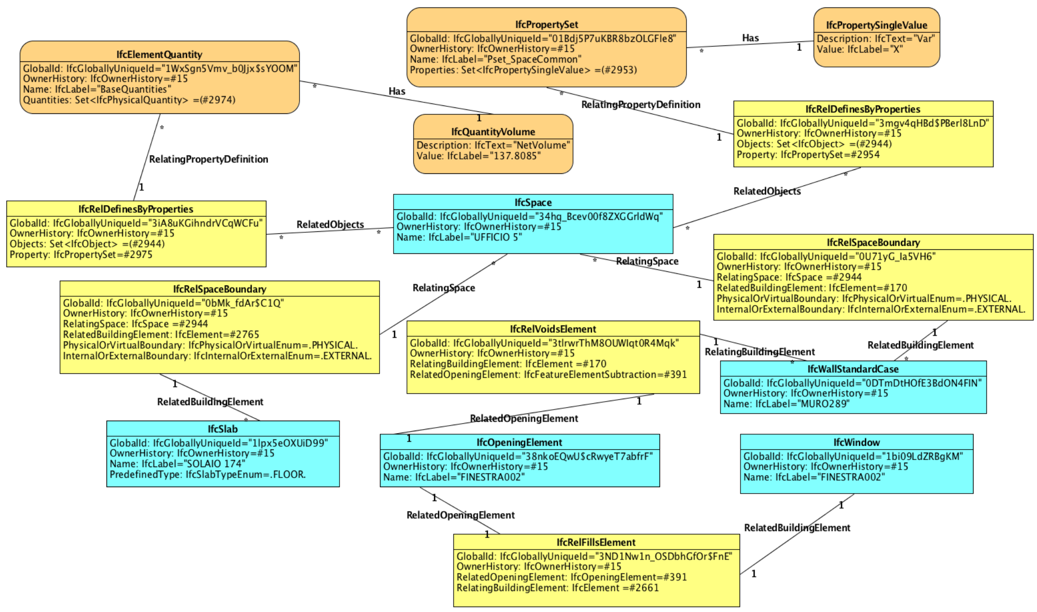

Figure 4.10. IFC metamodel of the example.

For this reason, we further abstract the model: each row containing an IFC element is represented by a graphical component of the metamodel, for example, the IfcSpace with id #2944 is matched to the block center which preserves the most important attributes as GlobalId, OwnerHistory and Name, for IfcRelSpaceBoundary with id #3154 we proceed in a similar manner so that the two blocks are connected with a bidirectional arrow "RelatingSpace" where the multiplicity of connection are indicated; in particular, the space can have many border relations while a border report refers to a single space. In the block metamodel, the obtained objects are highlighted in cyan, yellow relations and orange properties. The metamodel exactly mirrors the logic of the IFC model, note that despite being considered a very low partial model the number of components is already quite high.

4.3.4 From the IFC metamodel to a relational metamodel

The relational metamodel

For the identification of the most appropriate target metamodel, we focus on the characteristics that it should possess, our interest is aimed at obtaining a topological representation of the courthouse and to do so you will need to define a mapping of the elements belonging to the metamodel IFC with those of the relational metamodel. Also, once you get the topology, the relational metamodel must allow exploration in order to perform security analysis using an automated system. In summary we have some main features that the target metamodel must have:

- An effective topological representation of the building where there will be all entities of the IFC model and their connections.
- The possibility of being able to be explored to allow an analysis by an automatic system.
- The information content of interest present in the IFC model must be preserved.

Defining a transformation contract

To represent a transformation it is important to define a contract to express what it will do and what are the constraints for its use without going into details of implementative character. [36] In general what we're going to do is apply the transformation to a metamodel of origin to obtain one target, in practice, this process can be seen as an operation. The contract of a transformation must contain two types of constraints, the first must specify constraints on the source and metamodel of the target while the latter must define constraints on the relationships between the elements of the source and destination. We can then define the contract of a transformation as a tuple of 3 sets of constraints:

- A set of constraints to match the metamodel candidate to be the source of transformation.
- A set of constraints to be matched to a metamodel for it to be considered a viable product of the transformation.
- A set of constraints on the relations and developments of the elements from the source to the target metamodel.

Intuitively, what we want to do is the transformation of the metamodel IFC in a relational metamodel that reflects a graph.

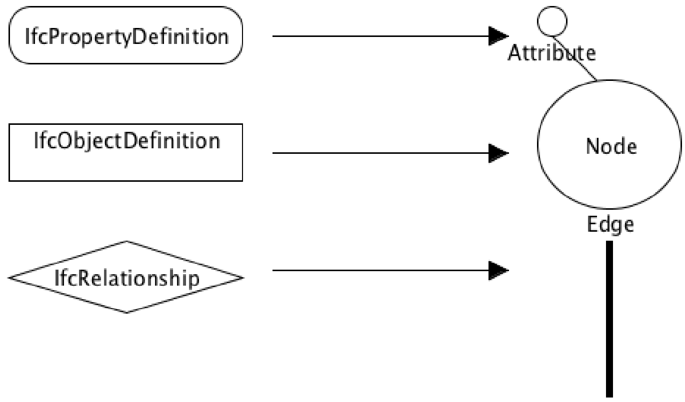

Figure 4.11. Logic representation of the transformation.

We, therefore, define an appropriate transformation by the aid of some class diagram representing the mapping of the components of the metamodel IFC with a set of classes representing the abstraction of a graph. To support this transformation we will also have to define the contract by defining the constraints expressed by a formal specification language appropriate.

Property mapping

The first class diagram that we are going to deal with, concerns the mapping of IfcSimpleProperty class, belonging to the IFC metamodel used as the source, in the destination class Property consisting of two attributes representating the name and value. The relation is a one to one type because our interest focuses on instances of IfcPropertySingleValue whereas IfcSimpleProperty is its generalization. In practice, what interests us is to be able to associate the attributes information with certain IFC entities, such as IfcSpace, in the target metamodel.

In case the properties taken into account are of a geometric nature magnitudes, such as lengths, areas and volumes, then we will go to map the IfcElementQuantity class in that GeometricQuantity destination class. The information content we are interested in, consists of the name of magnitude, by its numeric value

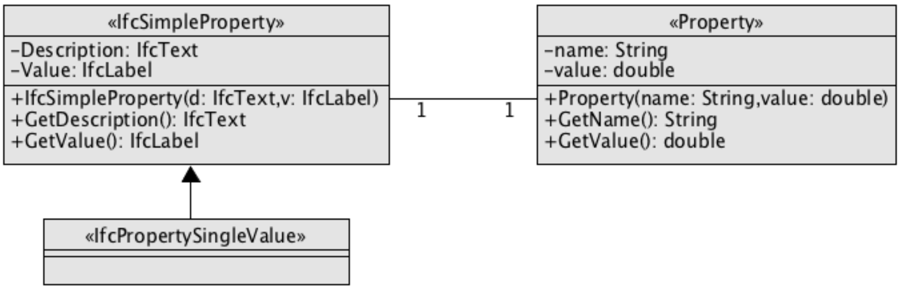

Figure 4.12. Class diagram representing the mapping from IfcSimpleProperty to the Property class.

and its unit of measurement; these fields include the GeometricQuantity class. The relationship is one-to-many, because, IfcElementQuantity may contain the definition of different geometric quantities, such IfcQuantityLength, IfcQuantityArea and IfcQuantityVolume, each of which can be mapped to an instance of the class GeometricQuantity.

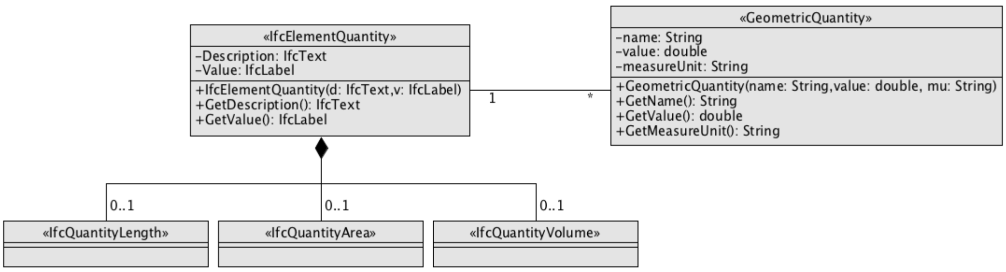

Figure 4.13. Class diagram representing the mapping of a geometric property.

Mapping of IfcSpace

One of the assumptions made during the design phase for the IFC model which is so projected in its abstraction, is linked to the presence of instances of IfcSpace class, this class is useful in identifying the presence of a room. We proceed with a one to one mapping of IfcSpace class in a node, the latter has two attributes that are reflected in the lists, geometricProperties contains all geometric quantities associated with the node and properties is a list of properties; the latter have the purpose to store any quantity associated to the space, such as the number of people that can contain. The nature of the two lists is derived from relationships of type IfcRelDefinesByProperties that associate IfcSpace class to classes that describe its properties.

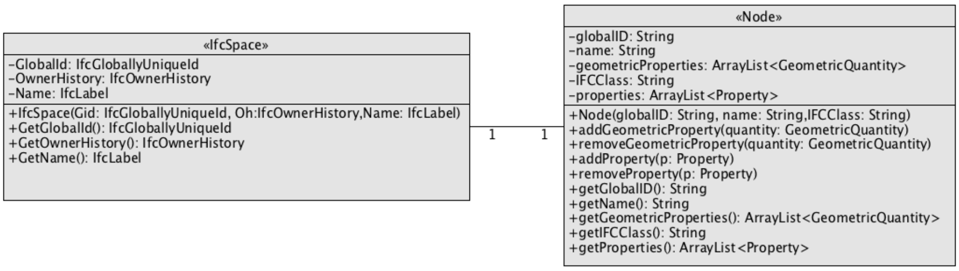

Figure 4.14. Class diagram representing the mapping of a IfcSpace to a node.

Mapping of IfcBuildingElement

IfcBuildingElement is the generalization of entities like IfcWall and IfcWindow, even this type of components belonging to the metamodel IFC will be mapped with an instance of the Node class. The relationship between a IfcBuildingElement and the Node class is of type one by one, because each element representing the floor, a wall, a window or a door will be transformed into a node with the exception of IfcOpeningElement that we will analyze later.

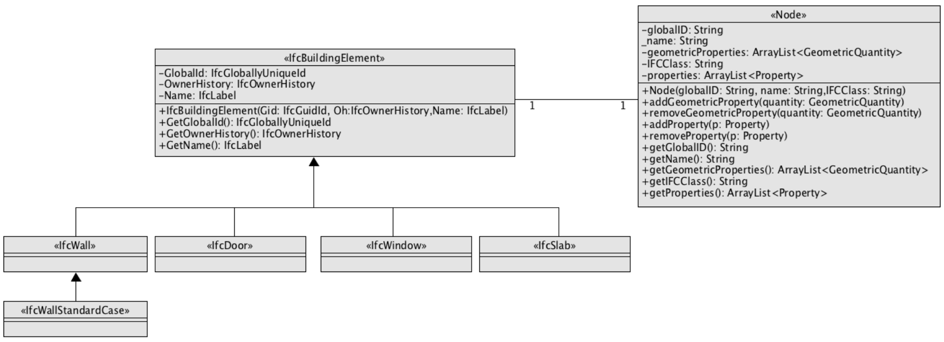

Figure 4.15. Class diagram representing the mapping of a IfcBuildingElement to a node.

Mapping of IfcRelSpaceBoundary

IfcRelSpaceBoundary defines a relationship between a IfcSpace and IfcElement, such as IfcWall or IfcSlab, so let's define the mapping with a relation of one-to-one with the Edge class. Edge Node has two attributes that identify entities derived from mapping of the IfcSpace considered and of the IfcElement present in IfcRelSpaceBoundary. The PhysicalOrVirtual and InternalOrExternal attributes are semantically similar to those present in the border relationship of the IFC metamodel, in the Edge class the appropriate methods to assign values of the corresponding enumerations are present.

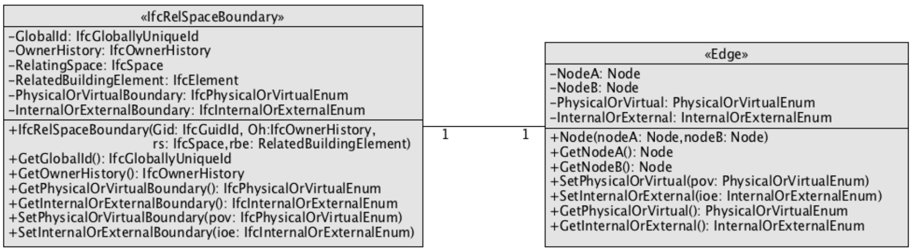

Figure 4.16. Class diagram representing the mapping of a IfcRelSpaceBoundary to an edge.

Mapping of IfcOpeningElement

In the IFC metamodel to identify whether in a wall there is a door or a window, it is necessary to go though its corresponding opening. In the mapping towards the relational target metamodel, the intention is to simplify this aspect, as not significant for the topological description. Therefore, we consider these three components of the IFC metamodel:

1. IfcRelVoidsElement which describes the relationship between the wall and the opening and it has, therefore, as RelatingBuildingElement and RelatedOpeningElement attributes.
2. IfcRelFillsElement which describes the relationship between the opening and the corresponding door or window using the RelatedOpeningElement and RelatingBuildingElement attributes.
3. IfcOpeningElement which is the opening present in both reports.

Basically, every time the above components are present in this configuration, we will map them to an instance of the Edge class by matching the NodeA and NodeB attributes of Node type with nodes generated from RelatingBuildingElement of IfcRelVoidsElement and homonymous attribute in IfcRelFillsElement; this would result in the avoidance of IfcOpeningElement.

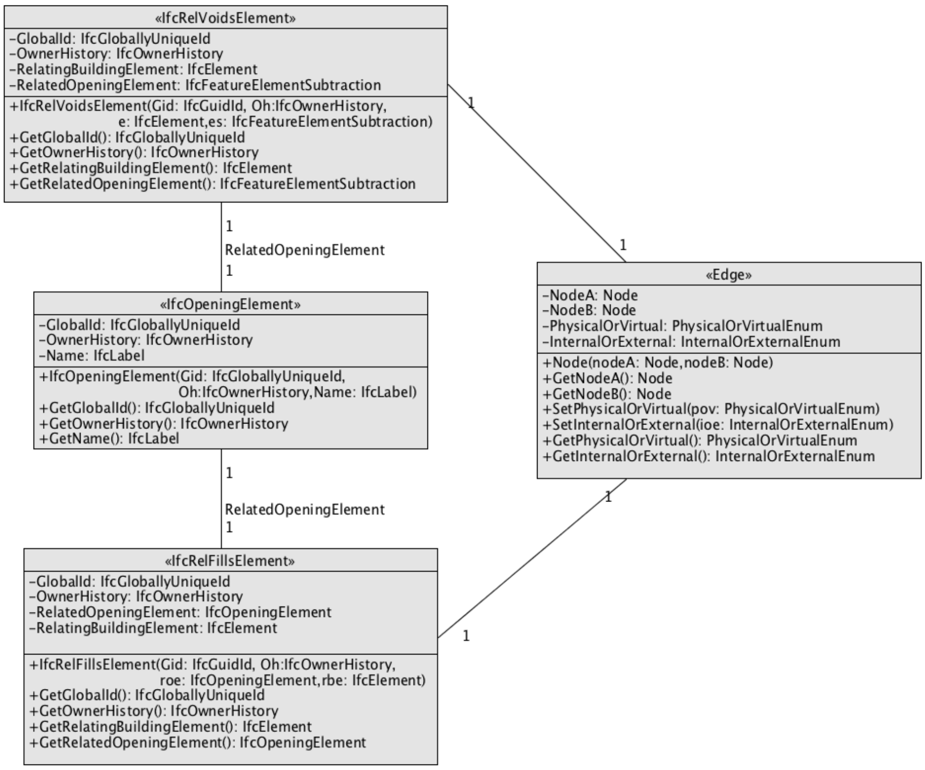

Figure 4.17. Class diagram representing the mapping of the relationship IfcRelVoidsElement and IfcRelFillsElement to an edge.

The transformation contract

A contract for a transformation between metamodels requires three sets of constraints, for each of the mappings described by the class diagram, we will define the constraints for the source metamodel, to the destination one and, finally, for the evolution of the component before and after the transformation; the union of all these constraints will make up the contract. The Object Constraint Language (or OCL)[37] is the formal specification language that we use to express the constraints, it is particularly useful, as it is an extension which allows to define constraints and queries on UML models. Mainly, the aspects that we are going to express with the constraints are:

- The **invariant** is a constraint that must be true for the entire life cycle of an object.
- The **precondition**, in the context of the definition of an operation, is a constraint that must be true before execution.
- The **postcondition** is a constraint that must be true after the execution of an operation, in other words, describes the effects of an operation in OCL.

Constraints on properties

With regard to the mapping of the properties, it is necessary to define the constraints needed to ensure that the instance of IfcSimpleProperty is valid as the source of the transformation, a constraint existing for the entire life cycle of this object is that the Value attribute of the property will have to take sound magnitudes and not void. The same constraint defined for the source class can be defined in a similar way to the destination class. The processing operation will require that the IfcSimpleProperty class belonging to the metamodel IFC will be mapped in the property class belonging to the relational metamodel, the content of the attributes description and value of the first will be assigned to the name and value of the second attribute.

```

package IfcMetamodel
context IfcSimpleProperty inv:
not (self.Value -> oclIsInvalid()) and self.Value ->
   notEmpty()
endpackage

```

```
package RelationalMetamodel
context Property inv:
not (self.value -> oclIsInvalid()) and self.value -> notEmpty()
endpackage

context IfcSimpleProperty::PropertiesTransformation(IfcSimpleProperty source) : Property
pre: source.oclIsKindOf(IfcSimplePropertySingleValue)
post: result.oclIsTypeOf(Property) and source.Description -> ToString() == result.name and source -> getValue().value == result.value
```

Listing 4.2: OCL constraints for the mapping of properties.

Similarly to what was done for the generic properties we can proceed to also define the constraints for the classes related to the geometrical quantities. The IfcElementQuantity class is mapped with the GeometricQuantity class, as before invariants discuss if the value attribute is valid for the source to the destination, the attribute values of the source class are transported in the target class; as regards the unit of measure, it is assigned based on the nature of the geometric magnitude respecting the international system.

```

package IfcMetamodel
context IfcElementQuantity inv:
not (self.Value -> oclIsInvalid()) and self.Value -> notEmpty()
endpackage

package RelationalMetamodel
context GeometricQuantity inv:
not (self.value -> oclIsInvalid()) and self.value -> notEmpty()
endpackage

context IfcElementQuantity::GeometricPropertiesTransformation(IfcElementQuantity source) : GeometricQuantity
pre: source.oclIsKindOf(IfcElementQuantity)
post: result.oclIsTypeOf(GeometricQuantity) and source.Description -> ToString() == result.name and source ->
```

```
    getValue().value == result.value and
( let mu = result.measureUnit in
if(source.oclIsTypeOf(IfcQuantityLength) then
mu = "METRE"
else
if(source.oclIsTypeOf(IfcQuantityArea) then
mu = "SQUARE_METRE"
else
if(source.oclIsTypeOf(IfcQuantityVolume) then
mu = "CUBIC_METRE"
else
mu=""
endif
endif
endif)
```

Listing 4.3: OCL constraints for the mapping of geometric properties.

Constraints for the mapping of IfcSpace

As regards the mapping of a IfcSpace in a node, we proceed by defining as invariant of the source class, the global id which must be valid and non zero, this must be true also for the destination class. In the transformation we have also to take into consideration all the properties in relation with IfcSpace, whether geometric or not, these properties are added to the collections belonging to the corresponding node. In particular, we verify that for each property in connection with the IfcSpace, one exists in the corresponding reference collection of the node and vice versa.

```

package IfcMetamodel
context IfcSpace inv:
not (self.GlobalId -> oclIsInvalid()) and self.GlobalId
    -> notEmpty()
endpackage

package RelationalMetamodel
context Node inv:
not (self.globalID -> oclIsInvalid()) and self.globalID
    -> notEmpty()
```

```
endpackage

context IfcSpace::IfcSpaceTransformation(IfcSpace source
    ) : Node
pre: source.oclIsKindOf(IfcSpace)
post: result.oclIsTypeOf(Node) and source.GlobalId ->
    ToString()==result.globalID and source.Name ->
    ToString()==result.name and result.IFCClass==IfcSpace
    and source.ElementQuantities ->size() == result.
    geometricQuantities ->size() and result.
    geometricQuantities -> forAll(q | source.
    ElementQuantities -> exists(e |  q =
    GeometricPropertiesTransformation(e))) and source.
    ElementQuantities -> forAll(e | result.
    geometricQuantities -> exists(q  | q =
    GeometricPropertiesTransformation(e))) and source.
    Properties ->size() == result.properties ->size() and
    result.properties -> forAll(p | source.Properties ->
    exists(pp |  p = GeometricPropertiesTransformation(pp)
    )) and source.Properties -> forAll(pp | result.
    properties -> exists(p  | p =
    GeometricPropertiesTransformation(pp)))
```

Listing 4.4: OCL constaints for the mapping of IfcSpace into nodes.

Constraints for the mapping of IfcBuildingElement

A IfcBuildingElement can be treated in a similar manner to a IfcSpace, in fact, they are also transformed into nodes in the destination model. Following, we will check the validity of the global identifiers for both classes of meta-models and, as for the postconditions of the transformation, we will check the content of simple attributes and collections exactly as we did for the IfcSpace. IfcBuildingElement is a parent entity, therefore it will be necessary to discuss the value of the IFCClass attribute of the resulting class based on the nature of the source entity.

```

package IfcMetamodel
context IfcBuildingElement inv:
not (self.GlobalId  -> oclIsInvalid()) and self.GlobalId
    -> notEmpty()
endpackage

package RelationalMetamodel
context Node inv:
not (self.globalID ->  oclIsInvalid()) and self.globalID
    -> notEmpty()
endpackage

context IfcBuildingElement::
   IfcBuildingElementTransformation(IfcBuildingElement
   source) : Node
pre: source.oclIsKindOf(IfcBuildingElement)
post: result.oclIsTypeOf(Node) and source.GlobalId ->
   ToString()==result.globalID and source.Name ->
   ToString()==result.name and ( let ifcClass = result.
   IFCClass in
if(source.oclIsTypeOf(IfcDoor) then
ifcClass = "IfcDoor"
else
if(source.oclIsTypeOf(IfcWallStandardCase) then
ifcClass = "IfcWallStandardCase"
else
if(source.oclIsTypeOf(IfcWindow) then
ifcClass = "IfcWindow"
else
if(source.oclIsTypeOf(IfcSlab) then
ifcClass = "IfcSlab"
else
ifcClass=""
endif
endif
endif
endif)
and source.ElementQuantities ->size() == result.
   geometricQuantities ->size() and result.
```

```
    geometricQuantities -> forAll(q | source.
    ElementQuantities -> exists(e |  q =
    GeometricPropertiesTransformation(e))) and source.
    ElementQuantities -> forAll(e | result.
    geometricQuantities -> exists(q  | q =
    GeometricPropertiesTransformation(e))) and source.
    Properties ->size() == result.properties ->size() and
    result.properties -> forAll(p | source.Properties ->
    exists(pp |  p = PropertiesTransformation(pp))) and
    source.Properties -> forAll(pp | result.properties ->
    exists(p  | p = PropertiesTransformation(pp)))
```

Listing 4.5: OCL constraints for the mapping of IfcBuildingElement into nodes.

Constraints for the mapping of a boundary relationship

For the mapping of a IfcRelSpaceBoundary we will define a constraint on the attributes GlobalId, RelatingSpace and RelatedBuildingElement which must have valid and not null values. The source class will be transformed into an Edge class, for this type of class we must verify that NodeA and NodeB attributes are valid and not void. Finally, in relation to the transformation constraints, it will be important to ensure the mapping of the attributes of the source and target classes, in particular the two node attributes of the Edge class must match the spatial mapping and of the building element of the border relationship.

```

package IfcMetamodel
context IfcRelSpaceBoundary inv:
not (self.GlobalId  -> oclIsInvalid()) and self.GlobalId
     -> notEmpty() and not (self.RelatingSpace  ->
    oclIsInvalid()) and self.RelatingSpace -> notEmpty()
    and not (self.RelatedBuildingElement  -> oclIsInvalid
    ()) and self.RelatedBuildingElement -> notEmpty()
endpackage

package RelationalMetamodel
context Edge inv:
not (self.NodeA ->  oclIsInvalid()) and self.NodeA ->
    notEmpty() and not (self.NodeB ->  oclIsInvalid()) and
      self.NodeB -> notEmpty()
endpackage

```

```
context IfcRelSpaceBoundary::
    IfcRelSpaceBoundaryTransformation(IfcRelSpaceBoundary
    source) : Edge
pre: source.oclIsKindOf(IfcRelSpaceBoundary)
post: result.oclIsTypeOf(Edge) and source.
    PhysicalOrVirtualBoundary -> ToString()==result.
    PhysicalOrVirtual -> ToString() and source.
    InternalOrExternalBoundary -> ToString()==result.
    InternalOrExternal -> ToString() and result.NodeA ==
    IfcSpaceTransformation(source.RelatingSpace) and
    result.NodeB == IfcBuildingElementTransformation(
    source.RelatedBuildingElement)
```

Listing 4.6: OCL constraints for the mapping of IfcRelSpaceBoundary in an edge.

Constraints of the mapping of openings

In this case, things get slightly more complicated, because, we should define consistency constraints of three classes of the source model, in particular, for IfcRelVoidsElement and IfcRelFillsElement classes we will verify that the attributes GlobalId, RelatingBuildingElement and RelatedOpeningElement are valid and then also the GlobalId of IfcOpeningElement is valid and not null. For the transformation operation a precondition ensures that the three classes discussed in the invariants are actually related to each other, so that the RelatedOpeningElement attribute present in the relationships would refer the opening considered. The postcondition will check that the two RelatingBuildingElement attributes present in the relationships are mapped in NodeA and NodeB attributes of the resulting edge.

```

package IfcMetamodel
context IfcRelVoidsElement inv:
not (self.GlobalId -> oclIsInvalid()) and self.GlobalId
    -> notEmpty() and not (self.RelatingBuildingElement
    -> oclIsInvalid()) and self.RelatingBuildingElement ->
    notEmpty() and not (self.RelatedOpeningElement ->
    oclIsInvalid()) and self.RelatedOpeningElement ->
    notEmpty()

context IfcRelFillsElement inv:
```

```
not (self.GlobalId -> oclIsInvalid()) and self.GlobalId
    -> notEmpty() and not (self.RelatedOpeningElement ->
    oclIsInvalid()) and self.RelatedOpeningElement ->
    notEmpty() and not (self.RelatedBuildingElement ->
    oclIsInvalid()) and self.RelatedBuildingElement ->
    notEmpty()

context IfcOpeningElement inv:
not (self.GlobalId -> oclIsInvalid()) and self.GlobalId
    -> notEmpty()
endpackage

package RelationalMetamodel
context Edge inv:
not (self.NodeA -> oclIsInvalid()) and self.NodeA ->
    notEmpty() and not (self.NodeB -> oclIsInvalid()) and
    self.NodeB -> notEmpty()
endpackage

context IfcOpeningElement::
    IfcOpeningElementTransformation(IfcRelFillsElement
    relFills, IfcRelVoidsElement relVoids,
    IfcOpeningElement opening) : Edge

pre: relFills.RelatedOpeningElement == relVoids.
    RelatedOpeningElement == opening
post: result.oclIsTypeOf(Edge) and result.NodeA ==
    IfcBuildingElementTransformation(relVoids.
    RelatingBuildingElement) and result.NodeB ==
    IfcBuildingElementTransformation(relFills.
    RelatingBuildingElement)
```

Listing 4.7: OCL constraints for the mapping of IfcOpeningElement.

4.3.5 Resulting metamodel and the level of abstraction

After defining the transformation in terms of mappings between classes and in terms of contract by specifying constraints in OCL, we can appreciate the relational metamodel resulting in reference to the example considered. In transforming the IfcSpace, the wall, the window and the floor have been mapped with node classes. The opening together with the two IfcRelVoidsElement and IfcRelVoidsElement relationships have been mapped with an edge representing the direct connection between the wall node and window node. The border relationships have been mapped as edges showing the link between the space and its construction element. The generic properties and geometric quantities were mapped respectively as classes Property and GeometricQuantity.

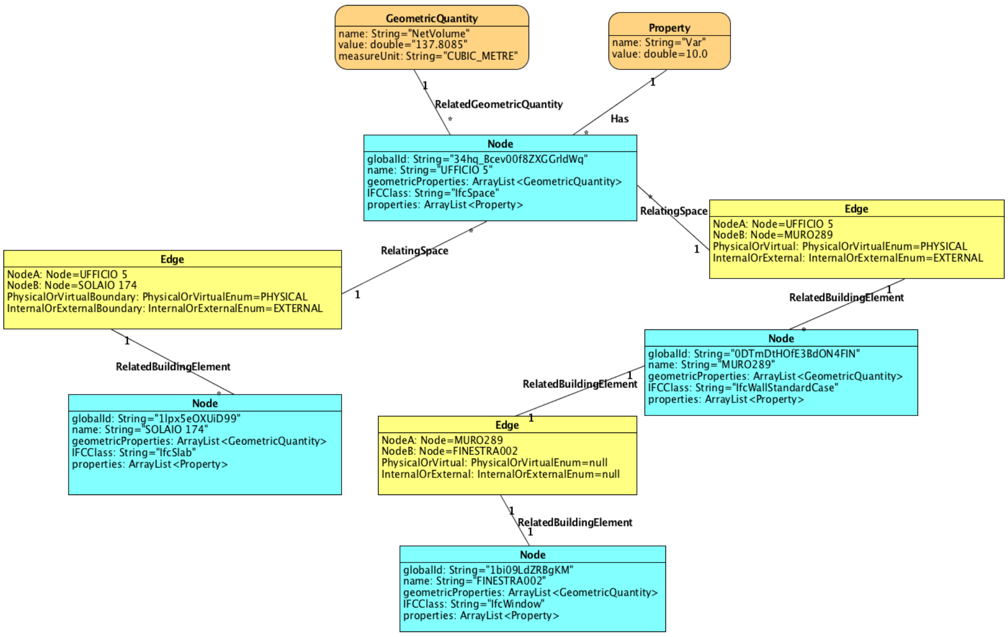

Figure 4.18. Relationship metamodel result of the example

Through an appropriate mapping, the elements of the metamodel IFC have been converted to nodes whose link is described by one edge, to each component of the relational metamodel you can assign properties that will guarantee to keep the information of interest. The model that meets our needs is definitely the graph, a model obtained with a reduction of the level of abstraction from the relational metamodel; the model thus obtained is certainly explored via known algorithms.

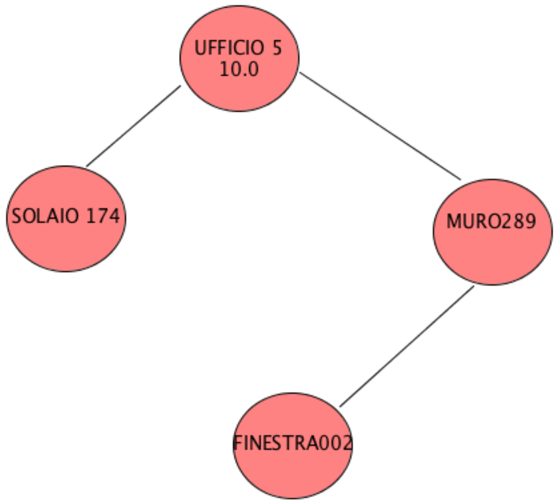

Figure 4.19. The graph obtained from the relational metamodel.

4.4 Transformation implementation

For the implementation of the transformation we chose a development based on Java 8 using the Eclipse IDE version Luna. Once the metamodel classes target were defined, it was necessary to implement a parser for the IFC language. To fulfill this purpose, we used a library derived from IFC Tools project that we have already seen in the section 3.2.4, this library has enabled us to solve the relationships defined in IFC and then instantiate objects belonging to our metamodel. After obtaining the relational metamodel, there has been the need to export it to formats suitable for data exchange, the chosen formats are json and gml.

4.4.1 The JSON format

The JSON format [38] stands for JavaScript Object Notation, is an open standard which uses text for the transmission of data objects made up by attribute-value pairs. This data format is the most commonly used for asynchronous communication between browser and server, it has been replacing XML and it is used by AJAX. JSON is a language-independent format in which objects to be transferred are represented, it derives from JavaScript, but since 2017 most programming languages include portions of code to generate and analyze data in JSON format. The type of official support on the Internet for this format is application/json, the extension for a JSON file is .json. To export in this type of format, we have exploited the implementation of Google called gson.

```
{
   "globalID": "07ijQdgcvFSgXZOYw16W7J",
   "name": "FINESTRA 003",
   "geometricProperties": [],
   "properties": [
      {
         "name": "PeopleWeight",
         "value": 0.0
      }
   ],
   "IFCClass": "IfcWindow
}
```

Listing 4.8: Example of a window node of the graph in JSON format.

4.4.2 The Gml format

Graph Modeling Language [39] is a viable option for a portable file format suitable for the graph representation. A GML file consists of a hierarchical structure of lists containing key-value pairs based on ASCII, it allows the description of graphs containing arbitrary data structures; its main features are portability, simple syntax, extensibility and flexibility. Several software are compatible with GML, among them we find Graphlet, Pajek, yEd, Leda and NetworkX. The export in this format was obtained by exploiting the implementation tinkerpop/blueprints.

```
node [
        id 2
        blueprintsId "0rpDoVkmL4$gjM022kZC2y"
        geometricProperties []
        IFCClass "IfcWindow"
        label "FINESTRA 003"
        properties [
            Property [ name PeopleWeight value 0.0 ]
          ]
      ]
```

Listing 4.9: Example of a window node of a graph in gml format.

4.4.3 Visualizing the generated graph

Finally, after implementing the two possibilities to export the graph we decided to create an applet to display the graph. The idea was to build this small component software to meet the essential needs of the display of the graph and ensure easy integration in any larger software; to do this we used the implementation JGraphT. JGraphT [40] is a free Java graphics library that provides mathematical objects to be applied to graph theory and algorithms. This library supports various types of graphs including oriented and non-oriented, weighed and non-weighed, immutable graphs etc. Although powerful, this library is designed to be simple and type-safe thanks to the use of java generics, the vertices of a graph can represent any type of object and you can also create graphs of graphs.

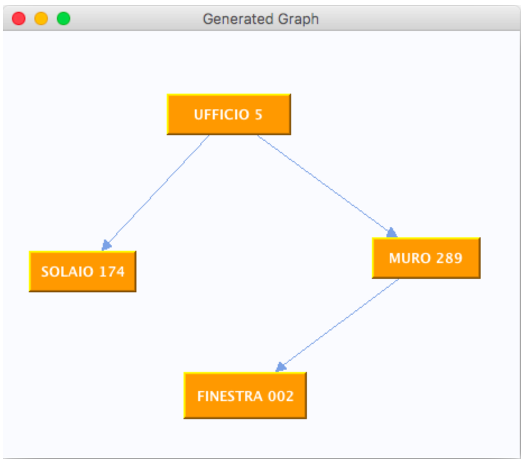

Figure 4.20. Screenshot of the example graph visualized with the applet.

Chapter 5

Analysis

In this chapter, we report on the analysis of example requirements elicited in Section 2.5.8. Firstly, we recall basic modelling principles of evolving spaces [3]. Secondly, to perform analysis we follow the approach of [9, 41]. We use the toolchain of [3] for the checking component. We treat the graph generated in Chapter 4 representing the topology of the courthouse as a *bigraph* [42], capturing static semantics of BIM/IFC specifications. Subsequently, we proceed to consider dynamics; how the space may change, giving rise to dynamic behaviour, enabling checking of complex probabilistic temporal properties. Finally, we report on experimental results obtained.

5.1 Bigraphs as the Semantic Domain

Bigraphs are an emerging formalism for structures in ubiquitous computing, consisting of two graphs. A *place graph* is a forest, a set of rooted trees defined over a set of nodes. A *link graph* is a hypergraph over the same set of nodes and a set of edges, each linking any number of nodes to names; this graph represents generic many-to-many relationships. Connections of an edge with nodes are called ports. Place and link graphs are orthogonal, and edges between nodes can cross locality boundaries. What follows is an informal presentation of the bigraphical theory.

$P.Q$	*Nesting* (P *contains* Q)	(5.1a)
$P \mid Q$	*Juxtaposition of nodes*	(5.1b)
$-_i$	*Site numbered* i	(5.1c)
$K_w.(U)$	*Node with control* K *having ports with names in* w. K *contains* U	(5.1d)
$W \parallel R$	*Juxtaposition of bigraphs*	(5.1e)

Bigraphs can be described through concise algebraic expressions (Formulae 5.1a-5.1e). The containment relationship is expressed in Formula 5.1a. Bigraphs can contain sites (Formula 5.1c) that can be used to denote place-holders; sites can be used to indicate presence of unspecified nodes. Controls are names that define a node's type; each node control can be associated with a number of named ports. P, Q, and U are controls of bigraph nodes. In Formula 5.1d the node identified by control K and port name w also contains U. Ports that appear in a formula with the same name are connected, forming a hyperedge with that name, called *link* in the sequel. We use the notation @*var* to refer to variables scoped in a formula. Bigraphs can be contained in roots that delimit different hierarchical structures; in Formula 5.1e, W and R are different roots.

Our objective is expressing topological information through containment and linking relations, mapping it to a bigraph placing and linking structure. IFC entities are mapped to bigraph nodes. The entity type (e.g. IfcDoor, IfcWall, etc.) is used to identify node controls while the entity name corresponds to a port name uniquely identifying it. The placing structure of the physical space is obtained by juxtaposing all the rooms in a building and subsequently nesting in each room nodes corresponding to entities contained in that room. For a wall, more than one node is created; each is nested inside nodes representing the rooms that wall bounds. Similarly, e.g. two nodes are created for each door;

these are nested inside the nodes representing the rooms that door connects. For example, partially shown in Formula 5.2 is a fragment of the bigraph representation of the courthouse.

$$\begin{aligned}
&IfcSpace_{UFFICIO10}.(IfcWallStandardCase_{MURO307,4} \mid \\
&IfcSlab_{SOLAIO8,8} \mid IfcSlab_{SOLAIO174,11} \mid -_0) \\
&\mid IfcSpace_{UFFICIO11}.(IfcWallStandardCase_{MURO307,4} \mid \\
&IfcSlab_{SOLAIO174,11} \mid -_1)
\end{aligned} \tag{5.2}$$

In Formula 5.2, controls $IfcSpace$ denote rooms, of which the port name denotes the name of the room as extracted from the IFC representation; similarly with $IfcWallStandardCase$ and $IfcSlab$ elements. Ports may link controls, representing connectivity; for instance, $UFFICIO10$ and $UFFICIO11$ share a wall $MURO307$. Sites imply presence of other, unspecified nodes in the bigraph formula.

5.2 Analysis: Courthouse design static requirements

Having specified an analyzable model of the courthouse design, we can formally express a property as a *pattern*, expressing the requirement R1 of Section 2.5.8. Our intuition is that a property of a given space can be expressed as a bigraph. A configuration described by a bigraph (in turn describing the courthouse design) satisfies a property if the bigraph specifying the property can be matched against it. The property which formally specifies the example's security requirement R1 has the following form:

$$\begin{aligned}
&IfcSpace_{AULADIBATTIMENTO}.(-_0 \mid IfcWallStandardCase_{@a,@b}.(IfcDoor_{@c})) \mid \\
&IfcSpace_{CORRIDOIO4PT}.(-_1 \mid IfcWallStandardCase_{@a,@b}.(IfcDoor_{@c})) \mid -_4 \\
\wedge &IfcSpace_{CAMERADICONSIGLIO}.(-_2 \mid IfcWallStandardCase_{@d,@e}.(IfcDoor_{@f}) \mid \\
&IfcSpace_{CORRIDOIO4PT}.(-_3 \mid IfcWallStandardCase_{@d,@e}.(IfcDoor@f)) \mid -_4
\end{aligned} \tag{5.3}$$

Formula 5.3 states that there is a common wall ($IfcWallStandardCase$) between AULA DI DIBATTIMENTO and CORRIDOIO 4 and in this wall there is a door ($IfcDoor$). This is signified by sharing the same port names – $@a$, $@b$ and $@c$. Additionally, that there is a common wall between CORRIDOIO 4 and CAMERA DI CONSIGLIO and in this wall there is also a door.

Due to presence of sites in the property specification, other entities that may be contained in rooms do not affect satisfaction of the property. Satisfaction of such a property is checked automatically through bigraph matching using the toolchain developed in our group [3]. We abstain from the details of this procedure as it is outside the scope of this thesis.

5.3 Analysis: Courthouse design dynamic requirements

We proceed to consider how the space of the courthouse may change, giving rise to dynamic behaviour. This is reflected by Bigraphical Reactive Systems (BRS) [42], which extend bigraphs by adding reaction rules defining possible reconfigurations. Reaction rules are parametric and specify how a bigraph can be modified by selectively rewriting some of its portions. Reaction rules have the general form of $R \rightarrow R'$, where R is a redex and R' is a reactum; both the redex and reactum are bigraphs. In particular, if a part of a bigraph that matches the redex is identified, it can be replaced with the reactum, in a fashion similar to graph rewriting. A BRS allows us to describe possible ways spaces can evolve through reaction rules. For instance, a fundamental reaction from the scenario presented in Section 2.5.5 is the ability of a prisoner transport to move within the courthouse. Formula 5.4 represents the movement of the prisoner transport from a $IfcSpace$ to another with which they share a connection on the bigraphical level. Note that when executing the movement, other elements contained in the $IfcSpace$s are not affected. Variables scoped in the reaction rule specification generalize the reaction rule for any names of $IfcSpace$s.

$$\begin{aligned} &IfcSpace_{@a}.(-_0 \mid IfcSlab_{@b,@c} \mid Prisoner) \mid \\ &IfcSpace_{@name3}.(-_1 \mid IfcSlab_{@b,@c}) \mid -_2 \rightarrow \\ &IfcSpace_{@a}.(-_0 \mid IfcSlab_{@b,@c}) \mid \\ &IfcSpace_{@name3}.(-_1 \mid IfcSlab_{@b,@c} \mid Prisoner) \mid -_2 \end{aligned} \tag{5.4}$$

Having defined how dynamics can be expressed with BRS, we proceed to perform analysis of the courthouse space regarding the requirement R2, given the assumptions outlined in Section 2.5.8.
More precisely, analysis will be performed through an interpretation of the BRS description over a Discrete-Time Markov Chain [43] (DTMC) model enabling reasoning with a probabilistic branching temporal logic. The DTMC formalism enables automated analysis of a wide range of quantitative properties specified through a probabilistic temporal logic. Probabilistic Computation Tree Logic (PCTL) [44] is such a branching time logic which extends CTL [43] with a probabilistic operator, manifested as quantitative extensions of CTL's *all* (A) and *exists* (E) operators. Model checking for PCTL involves determining states of an DTMC satisfying a PCTL formula. We abstain from the details of this procedure as it is outside the scope of this thesis; the interested reader can refer to [41].

$$\begin{aligned}&\mathbf{P{=}?}\ [\mathbf{F}\ \neg IfcSpace_{@room}.(-_0 \mid Asset \mid Prisoner) \mid -_2\\&\mathbf{U}\ IfcSpace_{AULADIBATTIMENTO}.(-_0 \mid Prisoner) \mid -_2]\end{aligned} \tag{5.5}$$

Formula 5.5 specifies requirement R2, to be evaluated over a DTMC describing the probabilistic evolution movements of the prisoner transport over the courthouse space.

5.4 Experimental Results

We implement the scenario presented in the toolchain of [3]. To obtain suitable models for our experiments, we place randomly a prisoner and assets in the bigraphical model of the courthouse. Subsequently, we generate the BRS obtaining a transition system (Figure 5.1), on transitions of which we randomly assign probability distributions on the moves of the prisoner transport.
Datasets for our experiments and models can be found at[1]. Subsequently, we proceed to verify the properties of Formula 5.4 and Formula 5.5, utilizing the PRISM model checker for probabilistic model checking. For our example and experimental model setup, checking the property of Formula 5.5 results to a probability of 0.5930 that the prisoner transport will not meet any assets in the way to the courtroom. The probability distributions used for the experiments can be found online.

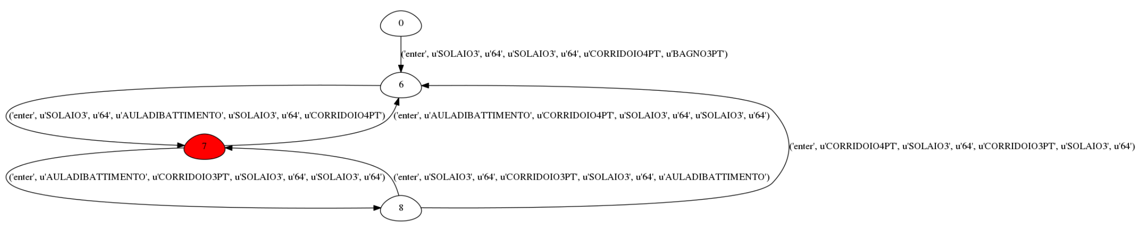

Figure 5.1. Fragment of a transition system representing movements of prisoner transport in the courthouse. The shaded state represents the configuration where the prisoner transport is finally located in the AULADI-BATTIMENTO.

[1] home.deib.polimi.it/tsigkanos/palazzodigiustizia/.

Chapter 6

Related Work

In this chapter we discuss about the documentation related to other work and approaches in this field of study, starting with a research study conducted at the Technical University of Monaco of Bavaria to reach the description of other approaches to international practices.

6.1 BIM database and queries formalization

Between research environments in the European landscape it is particularly noticeable that the polytechnic in Monaco of Bavaria, the Technische Universität München (TUM), has a research group dedicated to BIM in the design of buildings.

The research team KSD[45] based his research on the importance of the dissemination of knowledge organized as a source of relevant information in economic and social terms. During the design, architects often study the floor plans of existing buildings, however, the accessibility to these sources is hampered by un inconsistent use of terminology and lack of objective structured metadata. Their goal is to support the early stages of the design process by using information technology, in order to find similar architectural construction solutions as a source of inspiration, an explicit solution or a means to better understand the problems of the current design. The iterative nature of the design process results leads to a continuous exchange between the various creative phases, analytical and evaluation to select the most promising design variants. For digital formalization of architectural solutions, it proposes an approach based on the semantic footprint of a building that allows the identification of a building; it is the same approach used to recognize ad individual on the base of his fingerprints.

The purpose of research project metis is the development of innovative research methods to simplify the design in its conceptual stage. In the early design phases, people have only a general understanding of the construction and requirements of the design parameters that are taken into account. The use of references to drawings of existing constructions is an established method for the development of the ideas in practical solutions, to illustrate the design parameters or to demonstrate new possibilities. The existing projects serve as a source of knowledge as well as an example of spatial arrangement which shows architectural solutions based on specific forms, this methodology is effective, both in the early stages of design that in the subsequent ones. The main objectives of the metis project are:

- The development of a database of specific project components that can be used as references.
- The formalization of knowledge in BIM.
- The definition of methods and models for the query based on formal structures.
- The description and definition of a query language for spatial configurations.

The spatial arrangements of objects can be retrieved as a pictorial representation of the isomorphous, in such a way that the understanding of the distances and of the spaces can be accessed from the objects. The concept of semantic fingerprint is proposed as a useful method for the specific description of objects in a spatial configuration.

6.1.1 Graph Extractor/BIM2Graph

Especially in the early design stages, it can be useful to rely on the lessons learned from existing buildings, it would be useful to send a request to a database with the plans, in order to find similar. In this project a program for the development of an IFC file database has been developed. The defined process primarily involves the user who has to send, with the help of the client, a building model as IFC files to a webservice which save the model on a BIM open source server, the server will calculate the geometry of the building. The export service comes from a graph of the rooms and it offers a view to the user who can then edit the items and add new ones. The three-dimensional model of BIM server is useful in order to be able to understand the programmatic connections.

The webservice saves the plan in the database ar:searchbox and the graph generated on neo4j database:

- **ar:searchbox**[46] is a project created by the joint cooperation of the IT architecture department and of the TUM university library. It is proposed as an extension of the features offered to students, it provides a customized interface that allows the presentation of design data and the creation of relationships between architectural projects. To achieve this purpose, a scheme of quite accurate metadata has been developed to describe the construction projects in detail but it is also flexible enough to cover a wide spectrum of projects. Metadata includes attributes that let you apply different sorting and categorization criteria. Through the concept of decentralized management of the media server, the content can be inserted in parallel from multiple locations and it is immediately available to others by way of a reference to ar: searchbox. A sophisticated user access control mechanism ensures that the ability to view and change is available only to certain groups of users. The media server is fully accessible through a browser and requires no additional software.

- **Neo4j**[47] is the only enterprise-wide database based on a graph on the market, it has been developed entirely in Java by Neo Technology. Traditional databases are designed to scan paper forms, and automate highly structured business processes. RDBMS cannot model or store data and its relationship without an increase in complexity, as consequence, performance degrades with the increase in data size and the number of levels of relations. The addition of new data types and relationships requires a schema redesign and this increases the update times. For these reasons, RDBMS are inappropriate when there is a need to define relationships between data in real time. NoSQL databases are inadequate too, because they possess neither data structures for modeling relationships nor structured queries that support them. A database based on graphs instead, naturally allows to store, manage, analyze and use the data in the context of the connections. Understanding the relationships between data is the key to understanding dependencies, so a graph database is an essential tool to achieve this result with extremely high performance.

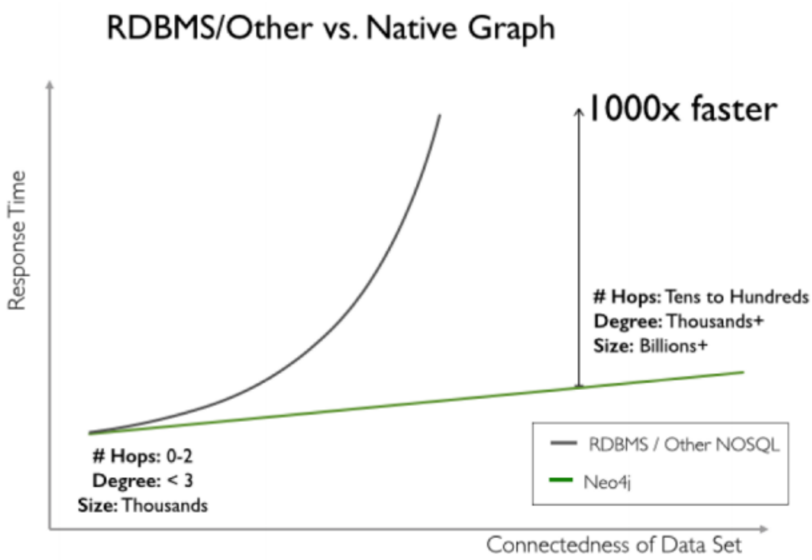

Figure 6.1. Performance comparison between neo4j and other databases

The graph associates to each item a unique identification and displays the information on the basis of the history and construction of the nodes. The AGraphML download service allows downloading the files existing in the database as AGraphML file. The performance of the Graph Extractor client has been tested for the loading and in any subsequent changes made manually with over 200 IFC files. So far almost 70% of the nodes and the edges are automatically recognized, for the remaining a certain degree of "post-editing" is required. In the future the manual of "post-editing" can be reduced by improving the attributes for the automatic identification of rooms and connections.

6.1.2 Dolphin

The goal of Dolphin[48] for Grasshopper is to support the early stages of the design process to find a source of inspiration, an explicit solution or a means to better understand a design problem. The Dolphin plugin enables the use of digital semantic information in the parametric work environment of Rhinoceros and Grasshopper using the format of AGraphML exchange, XML-based, developed by the research group KSD. The AGraphML format stores topological spatial information which can be accessed through local sources and web-based sources. The information can be downloaded in AgraphML file or read directly via a BIM cloud that manages spatial information in a graph database, the geometrical information on a BIM server and meta-information in a content management system. To encode spatial information in AgraphML format, automatic and semi-automatic methods have been developed. To formalize digitally architectural configurations, an approach based on semantic footprints of the buildings has been proposed. In order to map the spatial structure of an architectural arrangement some approaches based on graphs are explored to mathematically represent the fingerprints made by the information content regarding spaces, connections, alignment, orientation or integration at urban level. The nodes of the graph are labeled by following a fixed taxonomy concerning the spaces, whereas, for edges, labels will represent the relationships that exist between the spaces. The Dolphin components are divided into the following categories:

- **Analyzer** represents a type of component of which three components were implemented, with the aim of analyzing the construction information and display them in "Rhino". The first component is the "Node Analyzer", it retrieves data from all nodes, the second, called "Edge Analyzer", imports all data about edges and the third component, the "Graph Analyzer" imports the graph coordinates. The output content can be used by standard components "Grasshopper" to display the information.

- **BIM Tools** four of these components eases the interaction with BIM models during the design process. Once you download an IFC model from a BIM server, the "Grasshopper" component help in checking the result by enabling further processing in order to view the models built with the plugin Geometry Gym. The "BIM Visualiser" allows the user to select the model of construction of a series of models in the cloud.

- **Filter**: two filter components have been developed to define the basic requirements and to automatically check whether such conditions are met and in which part of the plan. The system distinguishes between two types of conditions, simple ones that have direct access to AgraphML data and complex ones that have to derive the values from AgraphML data to make a decision about the fulfillment of a certain condition. The "File Filter" and the "Room Filter" can be used to filter and view the remaining files from a set of files, or the rooms by a set of rooms in function of a given parameter; for example, the type of room or the presence of windows.

- **Source**: the local AgraphML Reader loads different AgraphML files from a folder and the component File Checker tests if a single file has all the attributes required by a AgraphML; finally, a single file or a list of them will be produced.

- **Statistics**: Three components compute the statistical data into AgraphML format as regards the type of room, the size and the number of connections with other rooms. The first component "Room Area Post" considers the size of the room by performing a calculation based on the comparison with a calculated mean, for example, a comparison is made between the bath dimensions with those of an average bathroom. The second component "Room to Room Edge Occurrences" calculates and shows the percentage of probability that there is a connection between rooms such as between the bedroom and the bathroom. The third component "Average Edge Occurrences" calculates the average number of connections.

- **Web Source**: four components of this type integrate the database Neo4j graph and "mediaTUM" data in "Grasshopper" and "Rhino" to allow further processing. As result, a list of AgraphML files will be returned which could also be reduced to a reduced number of results on the basis of defined criteria. The "Textual Filter" allows the definition of search parameters for requests to the "mediaTUM", for example, the definition of a certain architect or a project date. The component "Spatial Filter" allows the definition of search parameters for the request to Neo4j database, for example, based on the number of rooms or to a certain type. The third component, called "Building Searcher," has a central role and manages this Search "mediaTUM" and Neo4j taking into account the component "Search Credentials" whose content is specified by other components. In addition, with the component "Building Searcher" the number of results may be limited.

- **Coordinator** This type of Dolphin components allow the creation of AGraphML documents from a Rhino3D design as well as querying the online repository regarding existing plans through a new central server, the coordinator. The new components are perfectly integrated with the existing ones, and most of them can be combined freely.

6.1.3 Semantic fingerprint of buildings

Given a problem where a query is formalized, represented by a graph where we want to find a possible match in a database of graph models, it is significant to understand which techniques exist to handle some interesting situations that emerge from this issue; for example, the presence of a partial match, or the total absence of it.[49] This problem of graph matching becomes more complex with the presence of large databases, accordingly, the methods for indexing and pre-processing are becoming essential to maintain performance at a good level. Among the ideas that follow this direction, we have the filtering applied to graphs which aims to reduce the number of possible candidates using simple features and the cluster concept that is used to group similar graphs. In 1999, Messmer and Bunke propose a technique for recognizing the correspondence at the level of subgraphs based on the decision tree approach. They exploit an adjacency matrix to commute from a graph to a decision tree, this technique is very effective at run time but requires the calculation of all permutations of adjacency matrices and for computational reasons remains limited to graphics with up to 19 vertices.

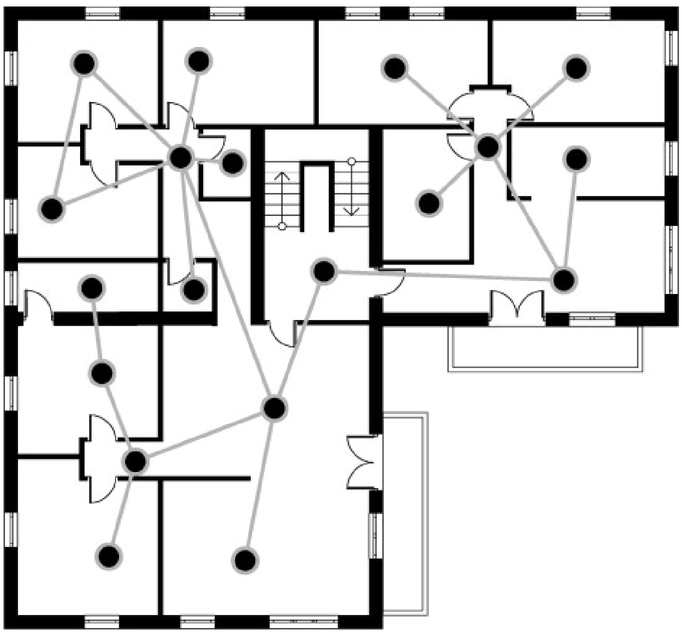

Figure 6.2. Semantic fingerprint of a floor in a building

The spatial configuration is a key feature, it is extracted and stored as a fingerprint in a semantic graph and, at last, a corresponding decision tree is obtained. The fingerprint is derived from IFC models and it represents the accessibility and nearness of the spaces in the structure. [50] This interconnection is encoded in a graph whose nodes are labeled with respect to the spaces whereas, the edges are labeled with regard to the presence of connections between them. Finally, to identify the correspondence between the graphs, a pattern matching is applied between subgraphs of the graph subject to analysis with those in a database of semantic imprints of other buildings. The isomorphism of subgraphs is a generalization of two issues, the first issue dealing with maximal clique and the second one tries to determine whether a graph contains a Hamiltonian cycle, so it is NP-complete. To reduce the complexity, an indexed structure for the spatial structure is calculated, this implies that the computationally intensive part is processed in advance and the resulting index is then used to facilitate a more rapid identification process.

6.1.4 Comparison with our approach

This methodology proposes the production of a database containing the fingerprints of semantic produced buildings using BIM methodology with an approach based on graphs. The work of the KSD research group aims to support early stages of the design process with a view oriented to the needs of architects to find effective architectural solutions to build solid buildings and respect the legal constraints. Our goals go beyond the concept of building seen as a mere construction and aim to analyze the mix of people, physical and virtual goods which, by establishing interconnections between them, could realize the risk scenarios. As regards the various technical aspects, however, there is much in common, because the realization of models representing the buildings and offering the possibility of being interrogated is a need preliminary to any type of analysis. In our approach, we exploit the structure inherent in BIM/IFC to generate a graph representation of relationships inherent in the design. This is in contrast to taking into account geometry in IFC descriptions. We formalize the process as a model-to-model transformation, by defining mappings between the classes of the source and target models and related constraints.

6.2 Analysis of building designs

Research in several areas has considered analysis of building designs with respect to requirements, thus we discuss other key approaches to position our work on cyber-physical spaces with respect to relevant lines of research followed by larger research communities.[9]

6.2.1 Rule-based checking

We have already seen in the section 3.1.3 in detail what is the approach to the control rules and appreciated the potential of software as Solibri, in the section 3.1.5, that offers an implementation providing capabilities for checking a model from simple checks such as shape overlappings, existence of specific objects as well as more detailed checks based on ISO accessibility regulations or fire code path distances. Our approach differs from current rule-based approaches in two directions. First, it views rule-based checks in the general context of formal (static and dynamic) semantics of the modeling formalism. Second, it also accounts for CPSp dynamics, which includes cyber aspects critical to complex requirements that span both spaces.

6.2.2 Analysis of dynamic scenarios

Different Model-based reasoning techniques have been utilized for studying and simulating certain dynamic aspects of building information models. However, these type of analisys are focused on very specific scenarios and cannot be adopted to generally reason about dynamic properties of a CPSp as it is possible using our approach. A more generic approach has been presented by Isikdag[51], it describes a BIM-oriented modeling methodology which extends standard BIM/IFC by providing detailed semantic information intended to support the analysis of general indoor navigation requirements. The approach allows designers of spatial environments to formulate SQL-like queries to extract semantic information considered relevant w.r.t. an indoor navigation requirement.

6.2.3 Extending BIM to smart built environments

Concerning the conceptual modeling of smart built environments, among the proposed approaches we find the extension of BIM by smart objects[52]. The modeling approach is conceptually at a lower level and it focuses on the modeling of different kinds of sensors and actuators and their integration into the physical space, while we are mostly interested in high-level conceptual entities along with their topological relationships and connections in the physical space. Although analisys of cyber-phisics spaces is considered, it has as specific target scenarios instead of providing a methodology of general application.

6.2.4 Spatial assistance systems

Spatial assistance systems have been developed oriented in particular to a general and architectural design.[53] With respect to our approach, a formal modeling approach for architectural design[54] has been proposed; based on an ontological model of architectural domain knowledge, architects may specify instances and employ reasoning services provided by the assistance system.[55] The project focuses on qualities of people-centered architectural design which can be characterized by navigational experience of users and subjective lighting influences.

6.2.5 Graphs as formal building models and case-based design

Different forms of graphs as formal models of static representations of buildings have been proposed by several approaches in different fields such as architecture informatics[50] or computer graphics[56] with different objectives. Focusing on security reasoning, the aim in early design phases is to propose a method and heuristics to discover security threats on building specifications via simulation, utilizing BIM.[57] Analyses such as similarity checking are performed based on graph matching techniques.[58] The overall goal is to build a comprehensive case base which can be queried to retrieve previously designed and stored building designs serving as reference examples in early design stages. Case-based design is largely complementary to our approach, in fact, static and dynamic models used within our approach, notably bigraphs, can be considered as another source of information which may be integrated and analyzed in existing case bases.

6.2.6 Bigraphs and BRS

The adoption of BRS for modeling physical spaces is a formal modeling approach to represent indoor spaces and mobility of objects and agents in those spaces[59], the idea is a reasoning about path-based navigation tasks, for example, the reachability of specific locations by agents. BiAgents[60] are a formalism who utilizes bigraphs for modeling the physical space and abstract algebraic structures for the cyber space and they have been used to identify strategies to prevent ill-defined concurrency situations that can emerge from cyber agents operating in shared physical structures.

Chapter 7

Conclusions & Outlook

In this chapter, we present our conclusions regarding the work done by analyzing the experimental results and the possible developments that lie ahead in the future

7.1 Conclusions

As in the construction industry, Building Information Models (BIM) are the standard for specifying complex information about building infrastructures, analyzing designs in a precise manner is crucial to engineering dependable spaces. Using the courthouse of Pavia as a case study, and considering security concerns of this particular domain, we followed software engineering principles to concretely enable analysis of security requirements of the building design.

In our approach, we initially considered the problem of security requirements elicitation. Subsequently we analyzed our findings, in the context of a requirements analysis study. Requirements in such a domain can predicate about static structures inherent in the courthouse physical space design, or reason about complex scenarios that involve dynamic behaviour of assets or people inhabiting it. We formulated two generic requirements; one, capturing static concerns to be verified upon a design, and one dynamic, capturing complex behaviour of a prisoner transport scenario. Since Building Information Model (BIM) descriptions of the courthouse physical space were available for the domain due to the collaboration of our group with the architecture-engineering-construction research group at Politecnico di Milano led by prof. Arlati, we utilized those as the foundation to obtain models amenable for analysis through model-to-model transformation techniques. In particular, in our approach we exploited the structure inherent in BIM/IFC to generate a graph representation of relationships inherent in the design. We formalized the process as a model-to-model transformation, by defining mappings between the classes of the source and target models and related constraints. Finally, by showcasing analysis over obtained models of generic example requirements of both static and dynamic nature, through external tools. Our preliminary experimental results are promising in the sense that models we obtained enabled verification of both static-structural and complex probabilistic properties predicating on behaviour inherent in the evolution of the space.

7.2 Future work

The computational dimension was not considered in this thesis; how existing security measures, like surveillance cameras and access control play a part in reasoning about security in the courthouse setting. Regarding the design phase of the engineering of dynamic spaces, a high-level mechanism along with graphical interfaces to specify dynamics at the BIM/IFC level would be desired to lower the technological barrier of entry. Moreover, a transformation mechanism from analyzable models back to a BIM representation would be useful to facilitate use by practitioners.

Bibliography

[1] *Milan court shooting: Man kills lawyer, judge and co-defendant.* http://www.bbc.com/news/world-europe-32231079.

[2] Liliana Pasquale, Carlo Ghezzi, Edoardo Pasi, Christos Tsigkanos, Menouer Boubekeur, Blanca Florentino, Tarik Hadzic, Bashar Nuseibeh. "Topology Aware Access Control of Cyber-Physical Spaces". In: ().

[3] Christos Tsigkanos, Liliana Pasquale, Carlo Ghezzi, IEEE Fellow, Bashar Nuseibeh. "On the Interplay Between Cyber and Physical Spaces for Adaptive Security". In: (2016).

[4] Christos Tsigkanos, Liliana Pasquale, Claudio Menghi, Carlo Ghezzi, Bashar Nuseibeh. "Engineering topology aware adaptive security: Preventing requirements violations at runtime". In: *2014 IEEE 22nd International Requirements Engineering Conference (RE)*. Aug. 2014, pp. 203–212.

[5] Sofia Bigoni, Gabriele Mansueto. "Modellazione digitale per un edificio storico: Applicazione di metodologie BIM per il palazzo di Giustizia di Pavia". Thesis. Politecnico di Milano, 2015/2016.

[6] Ralph R. Young. *The Requirements Engineering Handbook*. 2004.

[7] Elizabeth Hull, Ken Jackson, Jeremy Dick. *Requirements Engineering*. 2011.

[8] Mark Fleming, Sheila Gobes-Ryan, Janet Goodman, Ellen Keable, Jim Kessler, Karen McCown, Tim Scarbrough, Tammy Thomas. *U.S. COURTS DESIGN GUIDE*. 2007.

[9] Christos Tsigkanos, Timo Kehrer, Carlo Ghezzi. "Architecting dynamic cyber-physical spaces". In: *Computing* 98.10 (2016), pp. 1011–1040.

[10] *Contextual Inquiry*. URL: http://www.usabilitybok.org/contextual-inquiry.

[11] *Building Information Modeling*. URL: https://en.wikipedia.org/wiki/Building_information_modeling.

[12] *3D-4D Building Information Modeling*. URL: http://www.gsa.gov/portal/category/21062.

[13] *Industrial strategy: government and industry in partnership*. URL: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/34710/12-1327-building-information-modelling.pdf.

[14] *Nuovo Codice Appalti*. URL: http://www.ediltecnico.it/wp-content/uploads/2016/04/Nuovo-Codice-Appalti-18-aprile-2016-Bollinato-dalla-Ragioneria-dello-Stato_definitivo.pdf.

[15] Thomas L. Rosenberg. *Building Information Modeling*. URL: http://s3.amazonaws.com/academia.edu.documents/7868726/building%20information%20modeling%20-%20rosenberg.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1466358142&Signature=lPYhS%2Bu8rxyTYa%2Blj7Odyl10y0Q%3D&response-content-disposition=inline%3B%20filename%3DBuilding_Information_Modeling.pdf.

[16] Sijie Zhang, Jochen Teizer, Jin-Kook Lee, Charles M. Eastman, Manu Venugopal. "Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules". In: *ScienceDirect* 29 (2013), pp. 183–195. URL: http://www.sciencedirect.com/science/article/pii/S0926580512000799.

[17] *Architecture/Engineering/Construction (AEC) and Facilities Management (FM) Markets*. URL: http://www.opengeospatial.org/ogc/markets-technologies/aec.

[18] *Autodesk Revit*. URL: http://www.autodesk.com/products/revit-family/overview.

[19] *Solibri Model Checker*. URL: http://www.solibri.com/products/solibri-model-checker/.

[20] *Graphisoft ArchiCAD*. URL: https://www.graphisoft.com/archicad/.

[21] *ArchiCAD*. URL: https://en.wikipedia.org/wiki/ArchiCAD.

[22] *Industry Foundation Classes*. URL: https://en.wikipedia.org/wiki/Industry_Foundation_Classes.

[23] Mikael Laakso, Arto Kiviniemi. "THE IFC STANDARD - A REVIEW OF HISTORY, DEVELOPMENT, AND STANDARDIZATION". In: *www.itcon.org - Journal of Information Technology in Construction* 17 (May 2012). URL: http://usir.salford.ac.uk/28373/1/History_of_IFC_standard.pdf.

[24] *buildingSMART International Alliance for Interoperability*. URL: http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/.

[25] *EXPRESS - Data Modeling Language*. URL: https://en.wikipedia.org/wiki/EXPRESS_(data_modeling_language).

[26] *IfcOpenShell - the open source ifc toolkit and geometry engine*. URL: http://ifcopenshell.org/.

[27] *Open CASCADE Community Edition*. URL: https://github.com/tpaviot/oce/.

[28] *IfcPlusPlus*. URL: http://www.ifcplusplus.com/.

[29] *Qt*. URL: https://www.qt.io/.

[30] *OpenSceneGraph*. URL: http://www.openscenegraph.org/index.php/about/features.

[31] *IFC TOOLS Project - visualize your digital knowledge*. URL: http://www.ifctoolsproject.com/.

[32] *XBIM - The xBIM Toolkit*. URL: https://xbim.codeplex.com/.

[33] *IFC ENGINE DLL*. URL: http://www.ifcbrowser.com/.

[34] *IBM Knowledge Center - Creating model-to-model transformations*. URL: http://www.ibm.com/support/knowledgecenter/SS8PJ7_9.5.0/com.ibm.xtools.transform.authoring.doc/topics/ttransm2mover.html.

[35] *ArchiCAD 18 Reference Guide - Import/Export Settings*. URL: http://helpcenter.graphisoft.com/guides/archicad-18/archicad-18-int-reference-guide/interoperability/file-handling-and-exchange/working-with-ifc/importexport-settings/.

[36] Eric Cariou, Raphaël Marvie, Lionel Seinturier, Laurence Duchien. "OCL for the specification of model transformation contracts". In: *OCL and Model Driven Engineering, UML 2004 Conference Workshop*. Vol. 12. 2004, pp. 69–83.

[37] *OCL - Object Constraint Language*. URL: https://st.inf.tu-dresden.de/files/general/OCLByExampleLecture.pdf.

[38] *JSON - JavaScript Object Notation*. URL: https://en.wikipedia.org/wiki/JSON.

[39] Michael Himsolt. *GML: A portable Graph File Format*. URL: http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-technical-report.pdf.

[40] *JGraphT*. URL: http://jgrapht.org/.

[41] Christos Tsigkanos et al. "Adding static and dynamic semantics to building information models". In: *Proceedings of the 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems*. ACM. 2016, pp. 1–7.

[42] Robin Milner. *The Space and Motion of Communicating Agents*. Cambridge University Press, 2009.

[43] Edmund M Clarke, Orna Grumberg, and Doron A Peled. *Model checking*. MIT press, 1999.

[44] Hans Hansson and Bengt Jonsson. "A logic for reasoning about time and reliability". In: *Formal aspects of computing* 6.5 (1994), pp. 512–535.

[45] *KSD research GROUP*. URL: http://ksd.ai.ar.tum.de/?lang=en.

[46] Christoph Langenhan, Arne Seifert, Astrid Teichert, Frank Petzold. *ar:searchbox - Knowledge Management for Architecture Students*. Vol. 1. 2012, pp. 639–645. URL: https://mediatum.ub.tum.de/doc/1120687/1120687.pdf.

[47] Kamille Nixon. *Sustainable Competitive Advantage: Creating Business Value through Data Relationships*. 2015. URL: http://info.neo4j.com/rs/neotechnology/images/wp_sca_neo4j.pdf.

[48] *Dolphin*. URL: http://www.grasshopper3d.com/group/dolphin.

[49] Sheraz Ahmed, Markus Weber, Marcus Liwicki, Christoph Langenhan, Andreas Dengel, Frank Petzold. "Automatic analysis and sketch-based retrieval of architectural floor plans". In: *Pattern Recognition Letters* 35 (2014), pp. 91–100. URL: http://www.sciencedirect.com/science/article/pii/S0167865513001554.

[50] Christoph Langenhan, Markus Weber, Marcus Liwicki, Frank Petzold,Andreas Dengel. "Graph-based retrieval of building information models for supporting the early design stages". In: *Advanced Engineering Informatics* 27 (2013), pp. 413–426. URL: http://www.sciencedirect.com/science/article/pii/S1474034613000499.

[51] Umit Isikdag, Sisi Zlatanova, Jason Underwood. "A BIM-Oriented Model for supporting indoor navigation requirements". In: *Computers, Environment and Urban Systems* 41 (2013), pp. 112–123.

[52] Jianchao Zhang, Boon-Chong Seet, Tek Tjing Lie. "Building information modelling for smart built environments". In: *Buildings* 5.1 (2015), pp. 100–115.

[53] Thomas Barkowsky, John Bateman, Christian Freksa, Wolfram Burgard, Markus Knauff. “Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition: Reasoning, Action, Interaction (Sonderforschungsbereich/Transregio SFB/TR 8 Raumkognition: Schließen, Handeln, Interagieren)”. In: *it-Information Technology* 47.3/2005 (2005), pp. 163–171.

[54] Mehul Bhatt, Joana Hois, Oliver Kutz. “Ontological modelling of form and function for architectural design”. In: *Applied Ontology* 7.3 (2012), pp. 233–267.

[55] Carl Schultz, Mehul Bhatt. “A multi-modal data access framework for spatial assistance systems: use-cases with the building information model (bim/ifc)”. In: *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*. ACM. 2010, pp. 39–46.

[56] Raoul Wessel, Ina Blümel, Reinhard Klein. “The room connectivity graph: Shape retrieval in the architectural domain”. In: (2008).

[57] Stuart Porter, Terence Tan, Tele Tan, Geoff West. “Breaking into BIM: Performing static and dynamic security analysis with the aid of BIM”. In: *Automation in Construction* 40 (2014), pp. 84–95.

[58] Donatello Conte, Pasquale Foggia, Carlo Sansone, Mario Vento. “Thirty years of graph matching in pattern recognition”. In: *International journal of pattern recognition and artificial intelligence* 18.03 (2004), pp. 265–298.

[59] Lisa A. Walton, Michael Worboys. “A qualitative bigraph model for indoor space”. In: *International Conference on Geographic Information Science*. Springer. 2012, pp. 226–240.

[60] Eloi Pereira, Christoph Kirsch, Raja Sengupta. “Biagents–a bigraphical agent model for structure-aware computation”. In: *Cyber-Physical Cloud Computing Working Papers, CPCC Berkeley* (2012).