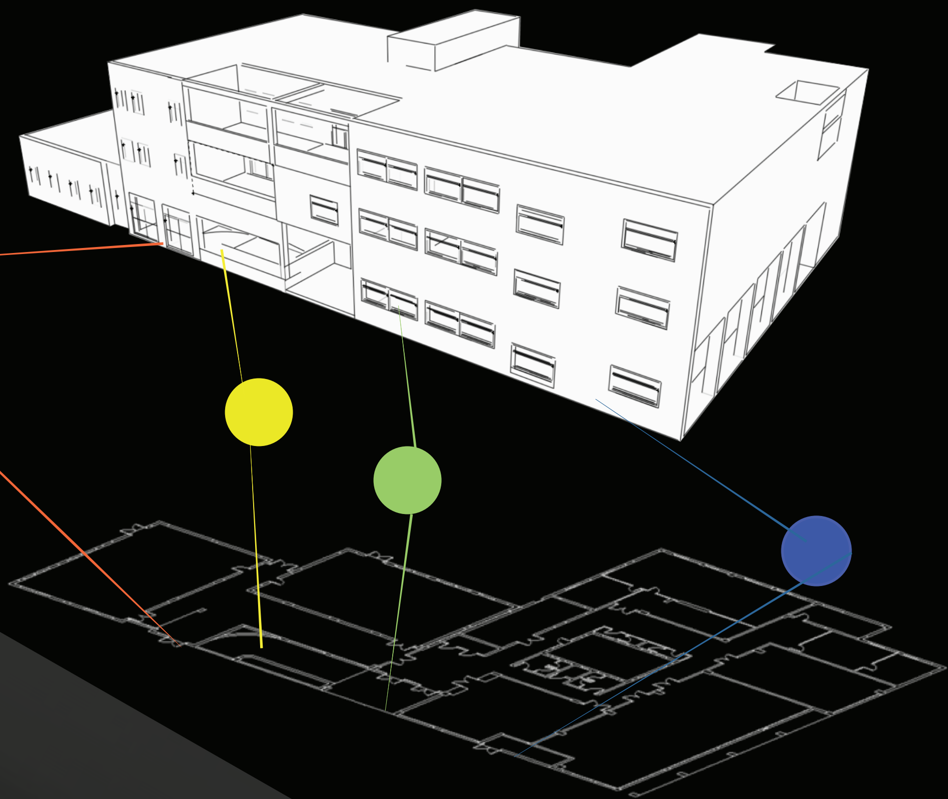
NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF RURAL AND SURVEYING ENGINEERING
MSc GEOINFORMATICS



DATA STRUCTURING & INTEROPERABILITY

OPTIONS FOR OPTIMISING 3D CITY MODELLING

ATHANASIOU AIKATERINI
SUPERVISOR: DIMOPOULOU EFI
Athens, June 2018

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF RURAL AND SURVEYING ENGINEERING
MSc GEOINFORMATICS

DATA STRUCTURING & INTEROPERABILITY OPTIONS FOR OPTIMISING 3D CITY MODELLING

MSc Thesis

Examination Committee:

Efi Dimopoulou	Marinos Kavouras	Margarita Kokla

Athanasiou Aikaterini

Athens, June 2018

ACKNOWLEDGMENTS

During this journey in NTUA, I had the opportunity to work in several research fields and at several projects and I have finally reached a conclusion with this MSc thesis that would not have materialized without support and encouragement by my dear friends and colleagues.

First and foremost, I want to express my sincere gratitude to my supervisor, professor Efi Dimopoulou. My studies would not have evolved in the same way without her presence. She has given me the freedom to explore and learn new things throughout my studies in NTUA. Most of all, I thank her for the constant motivation, the opportunities she gave me to access knowledge and finally for her trust in every step I made!

I gratefully acknowledge Marinos Kavouras and Margarita Kokla for their valuable feedback to complete this thesis.

I would also like to express my deepest appreciation to Eftychia Kalogianni. Without her constant support which so selflessly offered to me, I would have not managed to carry out my tasks so smoothly as I have done. She has always been there for me, especially under the most complicated and demanding conditions, offering me a guiding light and advising me throughout hard times. I am equally thankful to Maria Oikonomou and Manw Zygouri, who have always been available whenever I needed their assistance and guidance.

I would also like to thank Ioannis Pispidkis for his contribution to my work and for our collaboration all these years, as well as the team of 3D Campus project for all the great discussions related to this field.

On a personal level, my friends' support and continuous interest, is very much appreciated. While I cannot list them all I trust that everyone will feel recognized for their valuable contribution. Also, I would like to thank my colleagues in Click Law Partners, for their understanding all these years.

Last but not least, a heartfelt thanks to my family and my closest companions (Theoni and Eugenia) for their constant support and encouragement throughout this period, which motivated me to try to achieve my goals.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A city is decomposed into elements with clear semantics and defined spatial and thematic properties. Such elements are buildings, roads, railways, terrain, water, vegetation etc. These objects are further decomposed into different objects and even more detailed features, depending on the Level of Detail (LoD). 3D city models visually integrate the diversified geoinformation of the aforementioned elements within a single framework to provide notions of multiple resolutions and at different levels of abstraction. Various terms are used to define 3D city models, such as "Virtual City", "Cybertown", "Cybercity", "Digital City", "Smart City". Nowadays, 3D City Models are increasingly used in different cities and countries for an intended wide range of applications beyond mere visualisation. The generation of 3D city models is a relevant and challenging task from a practical and a scientific point of view.

3D city models are of two types, design and real world models. Design models are usually used for building industry purposes and to fulfil the requirements of maximum level of detail in the Architecture, Engineering and Construction (AEC) industry. Real world models are geospatial information systems that represent spatial objects around us and are largely represented in GIS applications. Research efforts in the AEC industry resulted in Building Information Modelling (BIM), a process that supports information management throughout buildings' lifecycle and is increasingly widely used in the AEC industry.

Currently, the development of BIM and GIS shares several overlapping application areas. At the same time, the gaps between the two concepts are gradually reduced as multiple integration methods are being developed serving various reasons, aiming to solve different problems. The benefits brought by the integration of BIM and GIS are being proved by more and more research. The integration of the two systems is difficult for many reasons. Among them, data incompatibility is the most significant, as BIM and GIS data are created, managed, analyzed, stored, and visualized in different ways in terms of coordinate systems, scope of interest, and data structures. Different 3D data models have been developed and utilized within several domains. At first, the 3D data models that were developed were purely geometrical/graphical mainly used for visualisation purposes such as COLLADA (COLLAborative Design Activity), VRML (Virtual Reality Modelling Language), X3D (eXtensible 3D) etc. and neglected the semantics of objects and relationships. However, for a large portion of the applications, not only the geometrical/graphical features are important, but the semantics as well, which are lacked by purely geometrical/graphical models. In some cases the integration of 3D data becomes almost impossible. Difficulties arise while translating data from one data model to another such as loss of information, improper conversion, loss of relationships, topological and geometrical inconsistencies, etc. In such circumstances, there arises the need to develop interoperability between 3D data models to exchange data seamlessly. The ideal, successful data interoperability should be able to fully transfer information from BIM to GIS, or vice versa, in terms of both geometry and semantics without data loss. This must first be a reality before integration of BIM and GIS at application level can be achieved.

Many data formats could be used to store 3D geometry, such as 3D Studio Max (.3ds), SketchUp (.skp), VRML and GeoVRML (.wrl), Openflight (.flt), and Collada (.dae).

However, the most relevant 3D data formats involved in BIM/GIS integration are Industry Foundation Classes (IFC), City Geography Markup Language (CityGML).

IFC is the primary open data schema used for information exchange within AEC/FM domains, which is EXPRESS-based and developed by buildingSMART. There are three ways for IFC to represent 3D geometry—boundary-representation (b-rep), constructive solid geometry (CSG), and sweep volumes. B-rep represents a 3D object using its bounding surfaces.. IFC classifies BIM models into five groups according to the details they contain by Levels of Development (LODt), from LOD 100 to LOD 500. Apart from the EXPRESS-based IFC file, buildingSMART also introduced a XML-based IFC standard, ifcXML. However, an ifcXML file is normally 3–4 times larger for storing the same information and is not as widely used as the EXPRESS-based file.

CityGML is an open standard data model and exchange format to store 3D models of cities and landscapes based on Geography Markup Language defined by the Open Geospatial Consortium (OGC) in Extensible Markup Language (XML) format. It is an application schema for GML 3.1.1 (GML3) that is a standard for sharing or exchanging 2D and 3D geospatial information over the internet. It defines the basic entities, attributes, and relations of a city, which is essential for cost-effective sustainable 3D city model maintenance. Similarly, for most XML-based data models, there are two parts to CityGML—the schema that describes the document and the instance document that contains the actual data. As with IFC, CityGML has definitions for different Levels of Detail (LoDs) from LoD0 to LoD4 to reflect the amount of detail included in a model. LoD0 model is just the footprint of the building (in 2D), while LoD1 models are the basic block model with flat roofs. In LoD3 and LoD4, the models incorporate doors and windows and have close exterior views, while their internal components are quite different. LoD4 contains interior spaces (rooms) and internal walls, while the model in LoD3 does not. However, the building model in CityGML is less complete and mature as in BIM, even in LoD4.

Currently, IFC and CityGML are representative data formats for BIM and GIS, respectively. Even though there are other formats involved, they are the most studied and accepted exchange formats. Apart from that, they are also complete ontologies for building and city models that could contribute to the construction of the semantic web. Therefore the integration of the two domains in data level is focused on the integration of these two formats.

In the integration of IFC and CityGML building models, substantial difficulties may arise in translating information from one to the other. Professionals from both domains have made significant attempts to integrate CityGML and IFC models to produce useful common applications.

This research aims to contribute from the very first step towards this integration and proposes a framework to better organise the elements during the design process in order to achieve semantically enhanced 3D information models according to IFC and CityGML standards. This is accomplished through the investigation of various options regarding the organisation of 2D architectural floor plans as input data, as well as through the exploration of CityEditor potentials in terms of this integration.

A set of rules are proposed to redraw architectural floor plans from real life. These rules mainly focus on reorganising information contained in floor plans, taking advantages of the layering and blocking functions supported by CAD application, in order to easily produce 3D models, through other applications, which can be easily extracted to IFC and CityGML formats, since the structure of the data is based on these models.

The research presented in this thesis can be used for future work on the interoperability between AEC/FM and Geomatics.

ΠΕΡΙΛΗΨΗ

Ο χώρος που μας περιβάλλει δομείται από στοιχεία με σαφή σημασιολογία και καθορισμένες χωρικές και θεματικές ιδιότητες. Τα στοιχεία αυτά ενδεικτικά είναι: τα κτήρια, οι δρόμοι, οι σιδηρόδρομοι, το έδαφος, το νερό, η χλωρίδα κλπ. Τα στοιχεία αυτά αναλύονται περαιτέρω σε διαφορετικές υποκατηγορίες ακόμη πιο λεπτομερειακών χαρακτηριστικών σύμφωνα με τον Βαθμό Λεπτομέρειας (Level of Detail (LoD)). Τα τρισδιάστατα (3D) μοντέλα πόλεων ενσωματώνουν οπτικά την ποικίλη γεωπληροφορία των προαναφερθέντων στοιχείων, μέσα σε ένα ενιαίο πλαίσιο, έχοντας ως στόχο τη διαχείριση των πόλεων. Διάφοροι όροι χρησιμοποιούνται για τον ορισμό των τρισδιάστατων μοντέλων πόλεων, όπως "Virtual City", "Cybertown", "Cybercity", "Digital City", "Smart City". Σήμερα, τα τρισδιάστατα μοντέλα πόλεων χρησιμοποιούνται όλο και περισσότερο σε διάφορες πόλεις και χώρες για ένα ευρύ φάσμα εφαρμογών, πέρα από την απλή απεικόνιση. Η δημιουργία τους αποτελεί ένα σημαντικό και δύσκολο έργο από πρακτική και επιστημονική άποψη.

Τα τρισδιάστατα μοντέλα μπορούν να διακριθούν σε δύο κατηγορίες: αυτά που αναφέρονται σε κλίμακα πόλης και αυτά που αναφέρονται σε κλίμακα κτηρίου. Τα μοντέλα σχεδιασμού κτηρίου χρησιμοποιούνται συνήθως για σκοπούς κατασκευής και για την εκπλήρωση των απαιτήσεων του μέγιστου επιπέδου λεπτομέρειας στον κλάδο της Αρχιτεκτονικής, Μηχανικής, των Κατασκευών και Αξιοποίησης Περιουσίας (Architecture, Engineering and Construction/ Facility Management AEC/FM). Οι ερευνητικές προσπάθειες στον κλάδο των AEC/FM οδήγησαν στη δημιουργία Μοντέλων Κτηριακών Πληροφοριών (Building Information Modelling (BIM)), μια διαδικασία που υποστηρίζει τη διαχείριση πληροφοριών σε όλο τον κύκλο ζωής των κτηρίων και χρησιμοποιείται όλο και περισσότερο στο πεδίο των AEC/FM. Τα μοντέλα σχεδιασμού πόλης είναι συστήματα γεωχωρικών πληροφοριών που αντιπροσωπεύουν τα χωρικά αντικείμενα γύρω μας και αντιπροσωπεύονται σε μεγάλο βαθμό από GIS εφαρμογές.

Επί του παρόντος, η ανάπτυξη του BIM και του GIS μοιράζεται αρκετούς επικαλυπτόμενους τομείς εφαρμογής. Ταυτόχρονα, η απόκλιση μεταξύ των δύο συστημάτων μειώνεται σταδιακά καθώς αναπτύσσονται πολλαπλές μέθοδοι ενσωμάτωσης που εξυπηρετούν διάφορους λόγους, με σκοπό την επίλυση διαφορετικών προβλημάτων. Τα οφέλη που προκύπτουν από την ενσωμάτωση των BIM και των GIS αποδεικνύονται από όλο και περισσότερες έρευνες. Η ενσωμάτωση, ωστόσο, των δύο συστημάτων καθίσταται δύσκολη για πολλούς λόγους. Μεταξύ αυτών, η ασυμβατότητα των δεδομένων είναι η πιο σημαντική, καθώς τα δεδομένα BIM και GIS δημιουργούνται, διαχειρίζονται, αναλύονται, αποθηκεύονται και απεικονίζονται με διαφορετικούς τρόπους όσον αφορά τα συστήματα συντεταγμένων, το πεδίο ενδιαφέροντος και τις δομές δεδομένων. Διαφορετικά αρχεία ανταλλαγής δεδομένων έχουν αναπτυχθεί και χρησιμοποιούνται σε διάφορους τομείς. Αρχικά, τα τρισδιάστατα μοντέλα που αναπτύχθηκαν ήταν καθαρά γεωμετρικά/γραφικά και χρησιμοποιούνταν κυρίως για σκοπούς απεικόνισης, όπως COLLADA, VRML, X3D κλπ. χωρίς να δίνουν σημασία στη σημασιολογία των αντικειμένων και τις σχέσεις τους. Ωστόσο, για ένα μεγάλο μέρος των εφαρμογών μεγάλη σημασία, πέραν των γεωμετρικών / γραφικών χαρακτηριστικών, έχει η σημασιολογία, η οποία λείπει από τα καθαρά γεωμετρικά / γραφικά μοντέλα. Προβλήματα προκύπτουν κατά τη μετάφραση δεδομένων από ένα μοντέλο σε άλλο, όπως απώλεια πληροφοριών, ακατάλληλη μετατροπή, απώλεια σχέσεων, τοπολογικές και γεωμετρικές ασυνέπειες κλπ. Υπό αυτές τις συνθήκες, προκύπτει η ανάγκη για διαλειτουργικότητα

μεταξύ των διαφόρων αρχείων ανταλλαγής δεδομένων για την απρόσκοπτη ανταλλαγή δεδομένων. Μέσω της διαλειτουργικότητας των συστημάτων θα καταστεί δυνατή η μεταφορά της πληροφορίας μεταξύ των BIM και GIS ή αντιστρόφως, όσον αφορά τόσο στη γεωμετρία όσο και στη σημασιολογία, χωρίς απώλεια δεδομένων.

Πολλά αρχεία ανταλλαγής δεδομένων θα μπορούσαν να χρησιμοποιηθούν για την αποθήκευση τρισδιάστατης γεωμετρίας, όπως το 3D Studio Max (.3ds), το SketchUp (.skp), το VRML και το GeoVRML (.wrl), το Openflight (.flt) και το Collada (.dae). Ωστόσο, τα πιο σχετικά πρότυπα που εμπλέκονται στην ενσωμάτωση BIM / GIS είναι το Industry Foundation Classes (IFC) και το CityGML Markup Language (CityGML).

Το πρότυπο IFC χρησιμοποιείται για την ανταλλαγή πληροφοριών στους τομείς AEC/FM, βασίζεται στο πρότυπο STEP και αναπτύχθηκε από την buildingSMART, ενώ έγινε πρότυπο από τον Διεθνή Οργανισμό Προτυποποίησης (International Organisation for Standardisation (ISO)) το 2013. Οι 3 βασικές κατηγορίες γεωμετρίας που χρησιμοποιεί είναι οι ακόλουθες: boundary representation (B-Rep), swept volumes και Constructive Solid Geometry (CSG). Το IFC ταξινομεί τα μοντέλα BIM σε πέντε ομάδες σύμφωνα με τις λεπτομέρειες που περιέχουν με βάση τα Επίπεδα Ανάπτυξης (Levels of Development (LODt)) όπως τα ονομάζει, από LODt 100 έως LODt 500. Εκτός από το αρχείο IFC που βασίζεται σε EXPRESS, το buildingSMART εισήγαγε επίσης ένα πρότυπο IFC που βασίζεται σε XML, ifcXML. Ωστόσο, ένα ifcXML αρχείο είναι κανονικά 3-4 φορές μεγαλύτερο για την αποθήκευση των ίδιων πληροφοριών και δεν χρησιμοποιείται τόσο ευρέως.

Το CityGML είναι ένα αντικειμενοστραφές μοντέλο πληροφοριών, βασισμένο στην γλώσσα XML. Εγκρίθηκε ως πρότυπο το 2010 από την Open Geospatial Consortium (OGC), ενώ η τρέχουσα έκδοση του βγήκε το 2012. Καθορίζει τις βασικές οντότητες, τα χαρακτηριστικά και τις σχέσεις των οντοτήτων μιας πόλης. Όπως και με το IFC, το CityGML έχει ορισμούς για πέντε διαφορετικά επίπεδα λεπτομέρειας (Levels of Detail (LoDs)) από LoD0 έως LoD4 για να αντικατοπτρίζει την ποσότητα λεπτομέρειας που περιλαμβάνεται σε ένα μοντέλο. Ωστόσο, το επίπεδο λεπτομέρειας του μοντέλο κτηρίου στο CityGML είναι λιγότερο πλήρες και ώριμο από το IFC, ακόμη και στο LoD4. Το CityGML οργανώνεται σε ενότητες (modules). Αποτελείται από 12 ενότητες, οι οποίες παρέχουν τον ορισμό διαφορετικών θεματικών μοντέλων (building, relief, city furniture, land use, water body, transportation), ενώ οι ενότητες - core, appearance, generics - ορίζουν δομές που μπορεί να εφαρμοστούν σε όλα τα παραπάνω θεματικά μοντέλα. Με αυτή τη δομή επιτρέπεται τόσο η διατήρηση διαφόρων εφαρμογών όσο και η επέκταση των θεματικών μοντέλων. Το σημασιολογικό μοντέλο του CityGML υιοθετεί το πρότυπο ISO 19100 για τη μοντελοποίηση γεωγραφικών χαρακτηριστικών. Σύμφωνα με το πρότυπο ISO 19109 τα γεωγραφικά χαρακτηριστικά είναι αφαιρέσεις του πραγματικού κόσμου. Τα γεωγραφικά χαρακτηριστικά δύνανται να έχουν ένα τυχαίο αριθμό χωρικών και μη χωρικών χαρακτηριστικών. Οι αρχές της αντικειμενοστραφούς μοντελοποίησης μπορούν να εφαρμοστούν προκειμένου να δημιουργηθούν οι σωστές ιεραρχίες.

Επί του παρόντος, τα IFC και CityGML είναι αντιπροσωπευτικά μοντέλα δεδομένων για συστήματα BIM και GIS, αντίστοιχα. Εκτός αυτού, παρέχουν όλες τις βασικές οντότητες ενός κτηρίου και ενός μοντέλου πόλης που θα μπορούσαν να συμβάλουν στην κατασκευή ενός σημασιολογικού ιστού. Επομένως, η ενσωμάτωση των δύο τομέων στο επίπεδο δεδομένων επικεντρώνεται στην ενσωμάτωση αυτών των δύο προτύπων.

Κατά την ενσωμάτωση των IFC και CityGML, προκύπτουν σημαντικές δυσκολίες στη αντιστοίχιση πληροφοριών από το ένα στο άλλο. Επαγγελματίες που ασχολούνται και με τους δύο τομείς έχουν κάνει σημαντικές προσπάθειες ενσωμάτωσης των μοντέλων CityGML και IFC για την παραγωγή χρήσιμων κοινών εφαρμογών.

Η εργασία αυτή στοχεύει να συνεισφέρει στο πρωταρχικό στάδιο αυτής της ενσωμάτωσης και προτείνει ένα πλαίσιο για την καλύτερη οργάνωση των στοιχείων κατά τη διάρκεια της διαδικασίας σχεδιασμού, προκειμένου να επιτευχθούν τα σημασιολογικά ενισχυμένα τρισδιάστατα μοντέλα πληροφοριών σύμφωνα με τα πρότυπα IFC και CityGML. Αυτό επιτυγχάνεται με τη διερεύνηση των διαφόρων επιλογών όσον αφορά την οργάνωση των δισδιάστατων αρχιτεκτονικών σχεδίων ως δεδομένα εισαγωγής, καθώς και με την διερεύνηση των δυνατοτήτων του CityEditor σε σχέση με αυτή την ενσωμάτωση.

Προτείνεται μια σειρά κανόνων για την αναδιατύπωση των αρχιτεκτονικών σχεδίων που παρέχονται από το πεδίο των AEC/FM. Αυτοί οι κανόνες επικεντρώνονται κυρίως στην αναδιοργάνωση των πληροφοριών που περιέχονται στις κατόψεις, επωφελούμενοι των λειτουργιών στρωματοποίησης που υποστηρίζονται από εφαρμογές CAD, για την εύκολη παραγωγή τρισδιάστατων μοντέλων μέσω άλλων εφαρμογών, τα οποία μπορούν εύκολα να εξαχθούν σε μορφές IFC και CityGML εφόσον η δομή των δεδομένων βασίζεται σε αυτά τα μοντέλα.

Σε αυτό το πλαίσιο, η ερευνητική εργασία έχει την εξής δομή:

Το Κεφάλαιο 2 ορίζει το θέμα κάνοντας μια εισαγωγή στα τρισδιάστατα μοντέλα πόλεων, τα πεδία εφαρμογής και τα κύρια χαρακτηριστικά τους. Εξετάζεται η ταχεία ανάπτυξη των εννοιών των GIS και BIM, ενώ παρουσιάζονται αναντιστοιχίες και ομοιότητες των δύο τομέων. Επιπλέον, σε αυτό το Κεφάλαιο αναφέρονται τα πιο συναφή διεθνή πρότυπα που χρησιμοποιούνται για την υποστήριξη λειτουργιών μοντελοποίησης και διαλειτουργικότητας στα συστήματα BIM/GIS, καθώς και τα πλεονεκτήματα και τα μειονεκτήματά τους.

Ο στόχος του Κεφαλαίου 3 είναι να διερευνήσει τις σχετικές ερευνητικές μελέτες όσον αφορά τη διαλειτουργικότητα και την ενσωμάτωση των τομέων BIM και GIS. Η επίτευξη της ενσωμάτωσης αυτών των δύο εννοιών αναμένεται να έχει σημαντικό αντίκτυπο στην επίλυση προβλημάτων στους κλάδους AEC/FM και της Γεωπληροφορικής. Επιπρόσθετα, αναλύονται διάφορα λογισμικά που υποστηρίζουν την ενσωμάτωση αυτών των τομέων, ενώ γίνεται αναφορά στα προβλήματα που προκύπτουν από τις γεωμετρικές και σημασιολογικές ασυνέπειες κατά τη διάρκεια αυτής της διαδικασίας.

Στο Κεφάλαιο 4 παρέχεται το μεθοδολογικό πλαίσιο για την έρευνα αυτή, σε συνδυασμό με μια πρακτική προσέγγιση στην περιοχή του Εθνικού Μετσόβιου Πολυτεχνείου (ΕΜΠ), εφαρμόζοντας τη συλλογική γνώση των προηγούμενων κεφαλαίων. Ο στόχος είναι η καλύτερη οργάνωση των δεδομένων κατά τη διάρκεια της διαδικασίας σχεδιασμού για τη δημιουργία μοντέλων σημασιολογικών τρισδιάστατων δεδομένων, αξιοποιώντας τα πρότυπα IFC και CityGML. Η πρακτική προσέγγιση υλοποιείται με τη χρήση του Trimble SketchUp η αξιολόγηση και τα αποτελέσματά του οποίου εξετάζονται στο Κεφάλαιο αυτό.

Τέλος, στο Κεφάλαιο 5 παρουσιάζονται τα κύρια συμπεράσματα της σχετικής βιβλιογραφικής έρευνας των τελευταίων χρόνων και αξιολογείται η προτεινόμενη μεθοδολογία. Προτείνονται επίσης, τομείς για περαιτέρω έρευνα σχετικά με τις επιλογές ολοκλήρωσης και διαλειτουργικότητας μεταξύ της CityGML και της IFC, σύμφωνα με την ανασκόπηση της βιβλιογραφίας και τα συμπεράσματα της παρούσας εργασίας.

Η έρευνα που παρουσιάζεται σε αυτή την εργασία μπορεί να χρησιμοποιηθεί σε μελλοντικές εργασίες με θέμα τη διαλειτουργικότητα μεταξύ των AEC/FM και της Γεωπληροφορικής.

ABBREVIATIONS

2.5D 2.5 Dimensions

2D 2 Dimensions

3D 3 Dimensions

3DCIM 3D City Information Model

3DCityDB 3D City Database

3DIF 3D Industry Forum

3DPIE 3D Portrayal Interoperability Experiment

3DPS 3D Portrayal Service

AEC/FM Architecture, Engineering, and Construction/Facility Management

AIA American Institute of Architects

API Application Programming Interface

B-Rep Boundary Representation

BeIM Building environment Information Modelling

BIM Building Information Modelling

BIMserver Building Information Modelserver

bSDD buildingSmart Data Dictionary

CAD Computer - Aided Design

CityGML City Geography Markup Language

COLLADA COLLAborative Design Activity

CSG Constructive Solid Geometry

DDM District Data Model

DFS Data/File conversion System

DXF Drawing Exchange Format

EMF Eclipse Modelling Framework

ESRI Environmental System Research Institute

ETL Extract Transform Load

FME Feature Manipulation Engine

GIS Geographic Information System

GLoD Geometrical Level of Detail

GML Geography Markup Language

GNSS Global Navigation Satellite System

GPS Global Positioning System

IAI BuildingSMART Alliance

ICT Information and Communication Technology

IDMs information Delivery Manuals

IFC Industry Foundation Classes

IGES Initial Graphics Exchange Specification

IGIM Integrated Geospatial Information Model

IoT Internet of Things

ISO International Organisation for Standardisation

KIT Karlsruhe Institute of Technology

KML Keyhole Markup Language

LADM Land Administration Domain Model

LBS Location Based Service

LIDAR Photogrammetry and Airborne Light Detection And Ranging

LOD Level of Detail

LoDt Level of Development

LoI Level of Information

MRC	Multi Representation Concept
NBIMS	National Institute of Building Sciences
OGC	Open Geospatial Consortium
OOCAD	Object-Oriented Computer Aided Design
OWL	Web Ontology Language
RDF	Resource Description Framework
REST	Representational State Transfer
SHP	SHaPefile
SLoD	Semantical Level of Detail
SM	Semantic Mapping systems
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
STEP	STandard for Exchange of Product model data
TLS	Terrestrial Laser Scanning
UBM	Unified Building Model
UML	Unified Modelling Language
VRML	Virtual Reality Markup Language
W3D	Web 3D Consortium
W3DS	Web 3D Service
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
XML	eXtensible Markup Language

1. INTRODUCTION

This thesis is within the field of Geomatics, a field which is specialized in, but not limited to, the usage, analysis and distribution of (geo) data and information. The value of the geodata depends on a range of factors such as the availability, accuracy, amount of detail, whether the data is up-to-date and the consistency with which it is stored relative to similar geo-datasets. Not only do these factors contribute to the ease and ability to analyse the geodata, but also to the interoperability between applications.

Architecture, Engineering, and Construction/ Facility Management (AEC/FM) is a field in which a vast amount of high value data is created manually. By bridging the gap between the fields of Geomatics and AEC/FM, a new data source can become available for geo-applications with an unprecedented amount of details.

Following these trends, new terms of standardising processes have been formulated and defined at different levels and different scopes. Interoperability is an important example of standardisation methods for information system components. In this thesis, interoperability is seen as a strategic tool for use in modelling buildings in 3D applications and exchanging information for 3D city models. To this end the thesis considers interoperability at the semantic level and aims to develop higher levels of interoperability in future research and studies.

1.1 Research Questions

The core research question of this MSc Thesis is:

The optimisation of transformation/ integration between IFC and CityGML standards, through suitable data structure and semantically enhanced 3D information modelling.

Including, but not limited to the following sub questions:

- Which is the optimal way to organize architectural and construction information as input data for generating semantically 3D information models?
- Which are the required steps for the preparation of 2D architectural floor plans for reconstructing 3D building models?
- How to address interoperability options between IFC and CityGML standards, without information loss and incorrect geometry.

1.2 Scope

Currently, the development of BIM and GIS shares several overlapping application areas. At the same time, the gaps between the two concepts are gradually reduced as multiple integration methods are being developed serving various reasons, aiming to solve different problems. However, there are still many obstacles and challenges to the achievement of BIM and GIS integration, as they interpret 3D modelling from two different perspectives and hence, mismatching information between them is one an important factor.

Thus, this research aims to contribute from the very first step towards this integration and proposes a framework to better organise the elements during the design process in order to achieve semantically enhanced 3D information models according to IFC and CityGML standards. This is accomplished through the investigation of various options regarding the organisation of 2D architectural floor plans as input data, as well as through the exploration of CityEditor potentials in terms of this integration.

1.3 Outline

Chapter 2 sets the scene by introducing 3D city models, their application fields and main characteristics. The rapid development of GIS and BIM concepts is discussed, while mismatches and similarities of the two domains are presented. Additionally, at this Chapter the most relevant international standards used to support modelling and interoperability functionalities in BIM/GIS systems are introduced together with their advantages and disadvantages.

The objective of Chapter 3 is to review the relevant research studies, in terms of interoperability and integration of BIM and GIS domains. Achievement of integrating these two concepts is expected to have significant impact on solving problems in AEC/FM and Geoinformation sectors and thus, the most relevant and frequently used data models for this purpose are highlighted. Additionally, various software supporting the integration of these domains are discussed, while the emerged problems referring to geometrical and semantical inconsistencies during this procedure are mentioned.

In Chapter 4 the methodological framework for this research is provided, along with a practical approach to the area of the National Technical University of Athens (NTUA), applying the collective knowledge of the previous chapters. The goal is to better organise the data during the design process, for generating semantic 3D data models, exploiting IFC and CityGML standards. The practical approach is implemented using Trimble SketchUp and its evaluation and results are discussed in Chapter 4.

Finally, in Chapter 5 the main conclusions of the related bibliographic research carried out over the last years are presented, and the proposed methodology is evaluated. Last but not least, discussion for further research on the integration and interoperability options between CityGML and IFC, according to literature review and the conclusions of the current thesis are included in Chapter 5.

2. BACKGROUND & RELATED WORK

3D city models are increasingly used in different cities and countries for an intended wide range of applications beyond mere visualisation. Section 2.1 sets the scene by giving the main characteristics of 3D city models. Section 2.2 discusses the development of GIS and BIM, it also focuses on the differences and similarities of two domains and identifies that while GIS and BIM have been developed for different purposes in two different domains, they do share some common components. And finally, Section 2.3 identifies the most relevant standards used in BIM/GIS systems and demonstrates their advantages and disadvantages.

2.1 3D City Modelling

A city is decomposed into elements with clear semantics and defined spatial and thematic properties. Such elements are buildings, roads, railways, terrain, water, vegetation etc. These objects are further decomposed into different objects and even more detailed features, depending on the Level of Detail (LoD). 3D city models visually integrate the diversified geoinformation of the aforementioned elements within a single framework to provide notions of multiple resolutions and at different levels of abstraction. Various terms are used to define 3D city models, such as "Virtual City", "Cybertown", "Cybercity", "Digital City", "Smart City".

3D city models can be defined as computerized or digital representations of the Earth's surface and its related spatial objects within a city or a vision to integrate multiple Information and Communication Technology (ICT) and Internet of Things (IoT) solutions in a secure fashion to manage a city's assets (Ma & Ren, 2017). These models enable a wide variety of applications that in turn create a demand for detailed models of a specific area or even a focused building model. In such focused models, the representation and relationships among spatial objects should also be understood and modelled (Stadler & Kolbe, 2007).

3D city models are increasingly used in different cities and countries for an intended wide range of applications beyond mere visualisation. Nowadays the association of a domain (energy consumption, air quality, etc.) to these 3D models is very common, in order to obtain richer 3D representations that can be used for simulations and reasoning (visual reasoning). A comprehensive inventory of usage of 3D city modelling has been presented by Biljecki et al. (2015). The authors identified 29 distinct use cases in several domains and more than a hundred applications of those cases (Table 2.1). Their research is based on Batty et al. (2000) who have segmented the use of 3D city models into 12 categories of industries (urban planning, telecommunications, emergency services, architecture, facilities and utilities management, marketing and economic development, property analysis, tourism and entertainment, e-commerce, environment, education and learning, and city portals) 15 years ago. The taxonomy of Batty et al. (2000) is oriented towards application domains, rather than use cases that have been used in (Biljecki et al., 2015) research.

CLASSIFICATION OF USE CASES	No	USE CASES	EXAMPLES OF AN APPLICATION
1. Non-Visualisation Use Cases	1.1	Estimation of the solar irradiation	Determining the suitability of a roof surface for installing photovoltaic panels
	1.2	Energy demand estimation	Assessing the return of a building energy retrofit
	1.3	Aiding positioning	Map matching
	1.4	Determination of the floorspace	Valuation of buildings
	1.5	Classifying building types	Semantic enrichment of data sets
2. Visualisation Use Cases	2.1	Geo-visualisation and visualisation enhancement	Flight simulation
	2.2	Visibility analysis	Finding the optimal location to place a surveillance camera
	2.3	Estimation of shadows cast by urban features	Determination of solar envelopes
	2.4	Estimation of the propagation of noise in an urban environment	Traffic planning
	2.5	3D cadastre	Property registration
	2.6	Visualisation for navigation	Navigation
	2.7	Urban planning	Designing green areas
	2.8	Visualisation for communication of urban of citizenry	Virtual tours
	2.9	Reconstruction of sunlight direction	Object recognition
	2.10	Understanding SAR images	Interpretation of radar data
	2.11	Facility management	Managing utilities
	2.12	Automatic scaffold assembly	Civil engineering
	2.13	Emergency response	Planning evacuation
	2.14	Lighting simulations	Planning lighting of landmarks
	2.15	Radio-wave propagation	Optimising radio infrastructure
	2.16	Computational fluid dynamics	Predicting air quality
	2.17	Estimating the population in an area	Crisis management
	2.18	Routing	Understanding accessibility
	2.19	Forecasting seismic damage	Insurance
	2.20	Flooding	Mitigating damage to utility management
	2.21	Change detection	Urban inventory
	2.22	Volumetric density studies	Urban studies
	2.23	Forest management	Predicting tree growth
	2.24	Archaeology	Visualising ancient sites

Table 2.1: 29 use cases of 3D city models divided into two groups: non-visualisation and visualisation (Biljecki et al., 2015).

Figure 2.1: City Elements

The main criteria have been used, by the later, for the classification of use cases, is the visualisation aspect. As the authors stated in their research the ambiguous terminology and fuzzy boundaries of use cases make the grouping schema subjective, which is sensible considering the vast usage of 3D city modelling. Additionally, Morton et al. (2012) identified over one thousand 3D city models worldwide, being used in smart cities, campuses, commercial properties, shopping malls, airports, museums, and more.

Prominent examples of 3D city modelling are: indoor navigation (Becker et al., 2009), emergency response (Chen et al., 2014; Tashakkori et al., 2015), energy demand estimation (Krüger & Kolbe, 2012), disaster management (Kolbe et al., 2008; Zlatanova & Holweg, 2004) urban and telecommunication planning (Lamberti et al., 2011) and 3D Cadastre (Góźdz et al., 2014; P. van Oosterom, 2013).

A typical 3D city model is derived from various acquisition techniques, photogrammetry, (Close range, Aerial, Satellite) (Suveg & Vosselman, 2004), Lasergrammetry (Tomljenovic et al., 2015), Global Positioning System (GPS), synthetic aperture radar (Shahzad & Zhu, 2015), architectural models and drawings (Yin et al., 2009), procedural modelling (Biljecki et al., 2016a), volunteered geoinformation (Goetz, 2013) or combination of the aforementioned techniques. Indicative bibliography is given for each of the various acquisition techniques. Each acquisition approach is tied to the LoD, a measure that indicates the spatio-semantic adherence of a model to its real-world equivalent, and the LoD has implications on its usability (Biljecki et al., 2016a). The concept of LoD will be analysed in sections 2.1.2, 2.2.1.2, 2.2.2.1.

The generation of 3D city models is a relevant and challenging task from a practical and a scientific point of view. Jusuf, Mousseau, Godfroid, and Soh (2017) distinguishes two different levels of planning process and modelling: the city/neighbourhood-scale and the building-scale. Urban planners and urban designers usually develop the city/neighbourhood-scale model, while building design teams such as architects and/or building developers develop building-scale models. The commonly used technologies for both scales are Geographic Information System (GIS) and Building Information Modelling (BIM) respectively, which will be analysed in Section 2.2.1. Both concepts work on different data formats and data exchanges (Section 2.3).

Current questions under research related to 3D city modelling are: Could the above mentioned 3D city models be used to integrate urban and environmental knowledge? (Lancelle & Fellner, 2010); How could they be improved to fulfil such a role?

Geospatial applications are often based on spatial objects as well as topological relationships. This information is usually classified in two sets that can be defined as the following models:

- Geometric model that defines the geometric objects and elements types.
- Semantic model that defines entities and their non-spatial characteristics and relationships among the entities.

Current research in those models will be presented in Sections 2.1.3 and 2.1.4.

2.1.1 Lifecycle of 3D City Modelling

The life-cycle of a 3D city model is about data acquisition (in terms of conception, design and construction), data management and storage (related to the usage and maintenance of the collected information) and data visualisation/dissemination (referring to the disposal of urban assets) for a specific application. Each step can follow various workflows/methodologies that have been developed and can be encoded and exchanged through numerous formats. As presented in Figure 2.2 (Biljecki et al., 2016a), data used for the creation of the model can be derived from designs (either representing planned/built objects or simulations) and/or can be collected as a sub-set of real-world according to a predefined LoD. Data management refers to the creation of the expected LoD either from reduction/generalisation from an existing 3D city model of a finer LoD or generation of the model from a lower LoD or raw data. Following, the spatial analysis for the relevant application results at the visualisation and dissemination of the 3D city model to the users.

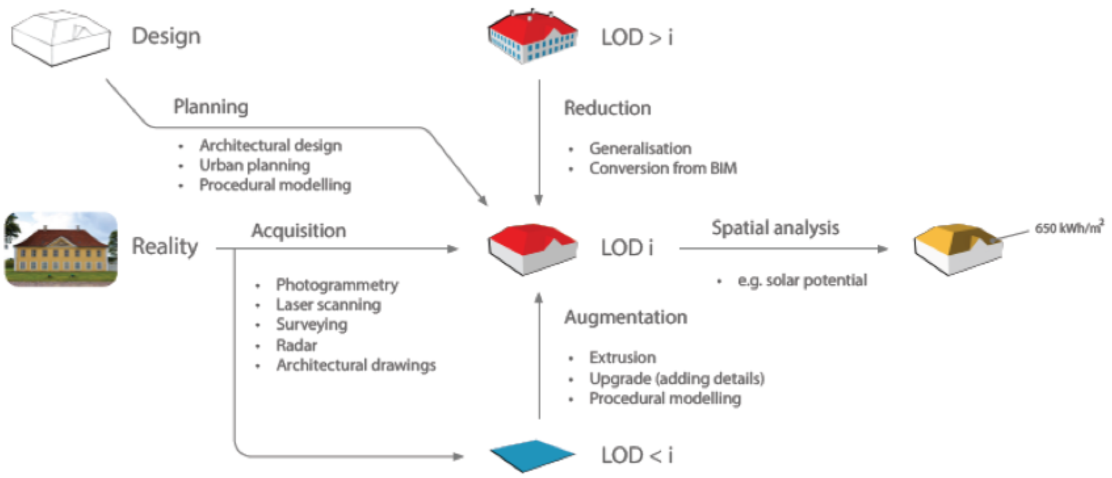


Figure 2.2: Life cycle of a 3D city model: different production workflows of 3D city models from the perspective of the level of detail (Biljecki et al., 2016a)

Given this background, 3D city models vary from the data and the techniques used for their creation to the representations that they provide, their modelling aspects, as well as their storage and exchange file formats. Different geo-data acquisition methods and technologies varying from photogrammetry, remote sensing and topographic surveying can be used for their creation. To name the leading ones, Global Navigation Satellite System (GNSS), mobile GIS platforms, Location Based Service (LBS), Terrestrial Laser Scanning (TLS), Photogrammetry and Airborne Light Detection And Ranging (LIDAR) (Kolbe et al., 2009; Lemmens, 2011).

Following, 3D city models address the 3D decomposition of urban space as they represent meaningful features of the real world, such as relief, infrastructure (bridges, tunnels, buildings, transportation features), vegetation, water bodies, etc.; characterised by different aspects, including multi-scale representation, semantics, geometry, topology, appearance, spatio-semantic coherence and extensibility (Kolbe, 2009).

The two dominant 3D city models that are used the last years are the CityGML model and the MVSM model. The first one is defined by the Open Geospatial Consortium (OGC) as an eXtensible Markup Language (XML) - based exchange format for 3D city models and supports the exchange of geometric data, semantic and topological aspects in a well-defined way (OGC, 2012b). The common information model behind CityGML defines classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantic and appearance properties (see Section 2.3.2). By covering thematic information and structures, CityGML complements 3D graphics formats like Keyhole Markup Language (KML) and X3D/ Virtual Reality Markup Language (VRML).

On the other hand, the use of MVSM models has increased since the technology behind their creation has evolved and it can provide creations of 3D city models with high amount of geometric and texturing information, in short time (Furukawa & Hernández, 2015). MVSM models are usually provided in a textured triangulated mesh and exchange through different file formats, such as COLLADA and OBJ. Kolbe et al. (2009) investigate the possibility to make CityGML the standard for 3D representation and exchange of MVSM models. A comparison of the basic characteristics of these two 3D city model formats are listed in the following table.

Criteria	CityGML	MVSM
Geometry	B-rep	Mesh
Semantics	Rich in semantics	No semantics
LoD	5 discrete LoDs	Not supported
Texture	Basic Support	Rich textures
Web rendering	Supported	Supported
Construction	Semi-automatic	Fully-automatic

Table 2.2: Comparison of CityGML and MVSM data model formats (Kavisha & Saram, 2015)

Generally, 3D city models can be stored in different ways, either file-based or in a database in a hierarchically, multi-scale way. 3D city models, structured according to CityGML standard, are often used for various complex GIS simulation and analysis tasks (Yao et al., 2018). Due to the large size and complexity of the 3D geospatial data, there is need to develop software solutions that can provide full support of efficient storage, analysis, management, maintenance, interaction, and visualisation of such 3D city models and also support their implementation through a multitude of formats.

Today 3D city models are generally stored in object-relational databases, such as Oracle and PostgreSQL (Stadler et al., 2008) and in most cases these databases are extended with dedicated city model. The CityGML schema is complex and leads to hundreds of tables in the database but most of these tables may not be used in most applications. Thus, a 3D geodatabase for CityGML must be able to describe each semantic object, which can be decomposed into parts and subparts and has a number of geometric properties of different geometry types and LoDs. The geodatabase must also be able to handle appearance data like individual surface textures, which typically are given in binary image file formats. All semantic objects have predefined thematic attributes and they can have an arbitrary number of generic attributes. Finally, since 3D city models cover large areas up to entire countries, the geodatabase must be able to manage the large data volumes and provide efficient access to the stored data for thematic and spatial queries. Thus, the last decade some geo-database solutions for CityGML-based 3D city models have been developed and are briefly presented below.

To start with, the '3D City Database' (3DCityDB) is a free 3D geo-database solution for CityGML-based 3D city models, developed as an open source and platform-independent software suite to facilitate the development and deployment of 3D city model applications (Yao et al., 2018). The 3DCityDB software package consists of a database schema for spatially enhanced relational database management systems (ORACLE Spatial or PostgreSQL/PostGIS) with a set of database procedures and software tools allowing to import, manage, analyse, visualize, and export virtual 3D city models according to the CityGML standard (Figure 2.3) (Yao et al., 2018). 3DCityDB contents can be directly exported in KML, COLLADA and glTF formats creating individual visualisations for different requirements. KML/COLLADA/glTF models are supported in a broad range of applications, such as earth browser like Google Earth, Cesium, NASA World Wind or ArcGIS with ArcGIS Explorer. Moreover, 3D City Database Importer/Exporter offers XML validation of CityGML documents (3DCityDB, 2018).

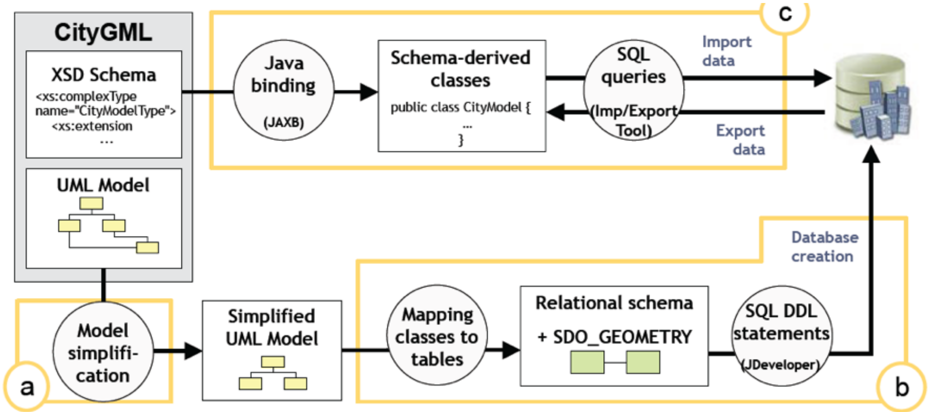


Figure 2.3: Development cycle of 3D City Database (3DCityDB, 2018)

Moreover, Environmental System Research Institute (ESRI) has developed semantically enriched database schema, the 3D City Information Model (3DCIM), aiming to provide compact and yet simple in structure, information models. This commercial solution has several in- built extension points and can be localized by utilising file geodatabase features and tools that are delivered with the information model. For building custom applications, it maintains a high expressiveness (Reitz & Schubiger-Banz, 2014). 3DCIM is organized into basic themes: the built environment, the legal environment, and the natural environment; and each of these themes shares some common attributes and traits. The last years several tools have been developed to achieve interoperability between 3DCIM and the CityGML-based models.

What is more, DB4GeO is a geospatial object-oriented database for 3D data, considering multidimensional aspects that are not offered in more traditional databases. A data model was developed by (Breunig et al., 2016) to handle spatial, temporal and thematic attributes. DB4GeO is fully implemented in the Java programming language and is based upon the open source object-oriented database management system db4o (Pouliot et al., 2010) using R-Tree based spatial access structures. Figure 2.4 illustrates the general system architecture of DB4GeO including a topological, a geometric and a spatio-temporal model. The geometric model was developed to handle simplex-based 3D objects utilising spatial access structures. Moreover, both the topological model and the spatial–temporal model are based on the geometric model. The model is also constantly evolving to integrate OGC standards such as CityGML and provide web services capabilities.

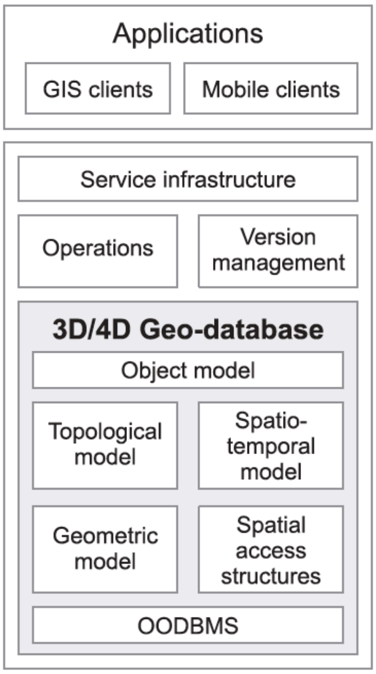


Figure 2.4: System architecture for DB4GeO (Breunig et al., 2016)

Due to the complexity of CityGML schema and the tables that are generated from it, it is difficult to implement and cannot be easily extended and parallelized. For that reason, apart from storing the data in object-relational databases, this can be done with cloud computation and NoSQL databases. Luan et al. (2014) proposed a 3D model management strategy based on Hadoop distributed file system to tackle the big data problem that 3D model management is facing. Sugumaran et al. (2014) developed and implemented a web-based 3D data processing system using Amazon's EC2 cloud computing environment. Their tests demonstrated the advantages of cloud computing over traditional approaches in time, cost and performance. Dobos et al. (2012) employ NoSQL database (Mongodb) to store 3D models in aid of public consultation. Mao et al. (2014) built a 3D city model management system with Mongodb, supporting storage and management of 3D city models in different format such as COLLADA, X3D, CityGML and 3DS, with a multiple representation structure (CityTree). Cloud computation methods can be deployed on NoSQL database to increase the analysis speed. It is suitable for big data applications such as 3D city model generalisation and visualisation.

As stated by Zlatanova et al. (2012) the formats used for data storage and exchange vary in order to serve GIS, BIM, Computer - Aided Design (CAD), computer graphics or Web domains. The different domains provide their own standards for the representation and the exchange of 3D city models. The components of the 3D city models are encoded by common file and exchange formats for 2D raster-based GIS data (e.g. GeoTIFF), 2D vector-based GIS data (e.g. AutoCAD DXF) and 3D models (e.g. CityGML, OBJ, COLLADA, KML, IFC, etc.). Some formats have been developed as standards by international organisations, while others have been accepted as standards due to their wide use by users and software (Zlatanova et al., 2012).

Research carried out by Zlatanova et al. (2012) and Kavisha and Saram (2015) has shown that several problems can occur during the conversion of data from one 3D city model to another, such as loss of information, loss of relationships, topological inconsistencies, etc. Additionally, Julin et al. (2018) propose a concept (Figure 2.5) that aims to harmonize 3D city models by combining the perspectives of 3D GIS, BIM and computer graphics. Developing the level of interoperation and integration reduces fragmentation and enables new applications.

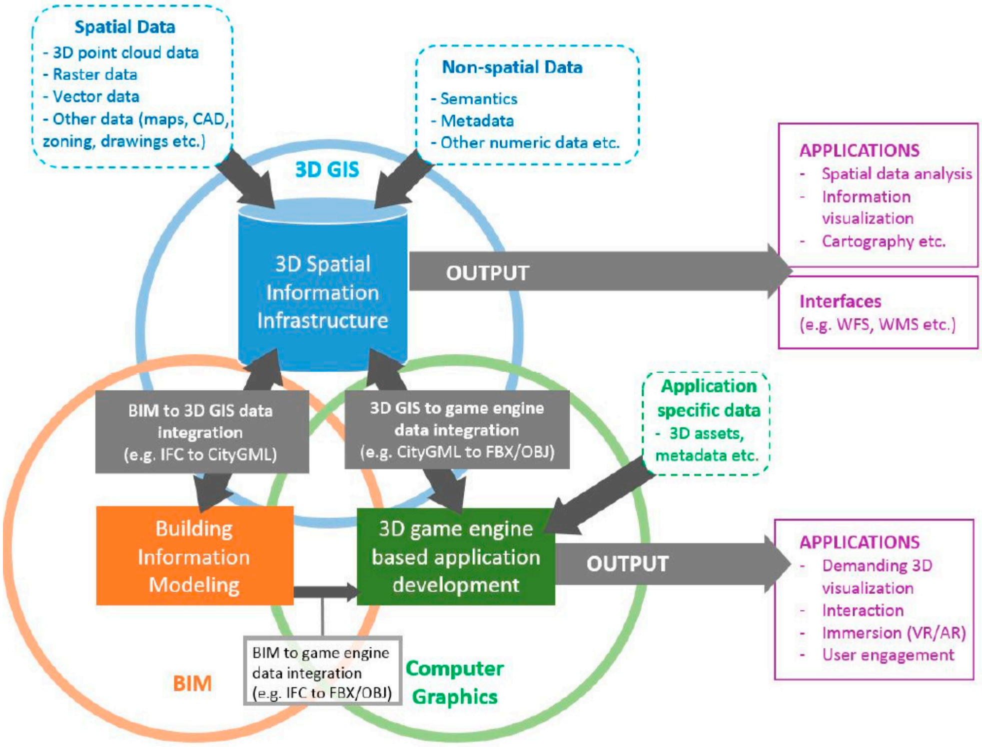


Figure 2.5: The concept for harmonising 3D city modelling (Julin et al., 2018)

In this scene, there is a growing research interest on the interoperability between 3D city models, focusing on the way data is encoded in the different format files, which is further discussed at Chapter 3.

Lastly, visualisation systems that are based on virtual 3D city models serve as enabling technology in GIS, as basis for 3D computational analysis and in manifold fields of application (e.g. urban planning, navigation, etc.). Due to the massive amounts of data to be processed, managed and rendered, common applications for visualisation of 3D city models characterised by a high degree of complexity, demand specialized systems with high-performance 3D graphics hardware. To face these challenges, many approaches have been developed the last years following a service-oriented approach for visualising virtual 3D city models.

In general, Service-Oriented Architecture (SOA) for 3D geo-visualisation foster the separation of concerns into management and rendering of virtual 3D city models and their interactive provision by client applications. While geospatial web services for geodata access (Web Feature Service-WFS), processing (Web Processing Service-WPS) and 2D portrayal (Web Map Service-WMS) are standardized trough OGC and are successfully used in the context of spatial data infrastructures, service-based management and rendering of 3D models as well as corresponding 3D client applications have not matured yet to a similar degree. The most mature 3D portrayal services are: the OGC 3D Portrayal Service (3DPS) supporting visualisation of very large 3D geospatial datasets online, image-based rendering, 3D Scenegraph (X3D, KML, X3DOM, etc.) (OGC, 2017); the OGC Web 3D Service (W3DS) creating 3D scenes of landscape and city models that can be explored interactively on the client and is designed as a Portrayal Service (OGC, 2010a); OGC Web View Service which is an extendable, interactive, image-based portrayal service for complex 3D geodata delivered as finally rendered images (OGC, 2010b); OGC 3D Portrayal Interoperability Experiment (3DPIE) addressing the testing and demonstration of different mechanisms for the portrayal, delivery, and exploitation of 3D geo data based on open standards-based formats and services (OGC, 2012a) and the portrayal scene service developed by ESRI, which allows users to share content via web scenes using ArcGIS.

Apart from the afore-mentioned web services for visualisation of 3D city models, there are several web viewers developed to service visualisation purposes for multiple applications. To name the leading ones: Cesium JS[1] is an open-source JavaScript library which uses WebGL to create 3D geospatial applications supporting open formats and developing new open formats when needed. It allows to switch among 3D virtual globe, 2D map and 2.5D Columbus view within a single Application Programming Interface (API) supporting at the same time, time-dynamic simulations and realistic environment features (sunlight, fog, moon, etc.)[2]. There are several web-viewer powered by Cesium JS, such as WebGL Earth, which is an open source software for the visualisation of satellite images and aerial photographs on top of a virtual globe (Chaturvedi, 2014) and Geo Borwser 3D which is an OGC compliant solution supporting 3D visualisation of maps, 3D buildings, etc. and precise measurements of distances and areas directly on the viewer[3]. Moreover, ESRI has released CityEngine Web Viewer, supporting the creation of users' own 3D scenes using the software package ESRI CityEngine or the Export To 3D Web Scene geoprocessing tool[4]. It is noted that comparing to the afore-mentioned 3D web-viewers this is a commercial solution.

Last but not least, there are also desktop-based viewers developed for 3D visualisation of geodata, to name a few: FZK Viewer developed by the Karlsruhe Institute of Technology (KIT) visualises the semantic data models IFC, CityGML and gbXML; Elyx3D allows integration of 3D models based on CityGML format and also any vector file or scene[5], azul is a 3D viewer for 3D city models in CityGML, CityJSON, OBJ, OFF and POLY supporting loading multiple files, selecting objects by clicking them or selecting them in the sidebar, and browsing their attributes (only supported by macOS)(TUDelft, 2018) CityGML Viewer is a platform-independent offline CityGML viewer for use in mobile Augmented Reality environments (Blut et al., 2017), Adobe 3D PDF, etc.

2.1.2 Level of Detail (LoD)

The concept of **LoD** can be found in many fields. In writing, for example, the LoD refers to three concepts: the precision in using the right words; the generality of statements and the organisational strategy in which authors arrange ideas according to a common topic in the hierarchy of detail. Sometimes is known as **level of abstraction**. This term is being used to other fields as well but with different definitions, some indicative examples are: i) the computer graphics (Luebke et al., 2002), ii) GIS and 3D city modelling and iii) rendering and modelling software..

[1] https://cesiumjs.org
[2] https://www.agi.com/products/open-source
[3] http://geobrowser3d.com
[4] http://doc.arcgis.com/en/arcgis-online/reference/about-cityengine-web-viewer.htm
[5] https://1spatial.com/products/elyx/elyx-gis-platform/elyx-3d/

In 3D city modelling, , the notion of LoD is of much importance. LoD defines the degree of abstraction of real-world objects, primarily designated to use an optimum amount of details of real-world objects according to the user's needs, and computational and economical aspects (Biljecki, 2017). The amount of detail that is captured in a 3D model, both in terms of geometry and attributes, is collectively referred to as the LoD, indicating how thoroughly a spatial extent has been modelled. As a result, the LoD is an essential concept in GIS and 3D city modelling, but as (Goodchild, 2011; He, 2012) mention is a subjective task.

LoD in the Life-cycle of 3D city models

In fact, the LoD concept is important in all steps of a typical 'life-cycle' of a 3D city model as Biljecki et al. (2016b) mention. It is a significant factor in contracting — the LoD is frequently found in tenders to describe the characteristics of the desired 3D city model. Having the LoD in mind when planning the acquisition of data is essential for proper budgeting of resources, and the LoD determines the acquisition technologies that ought to be employed as different LoDs are a result of different data acquisition approaches. In the acquisition of the data, the LoD further serves as the principal instruction on how thoroughly to acquire the data (Yang & Lee, 2017). Processing the data is influenced by the LoD as well. After the dataset is acquired, the LoD influences the storage aspect, substantially influencing the storage footprint, and necessitating compression and integration techniques. Quality control is another aspect in the life cycle of the 3D city model where the LoD is consulted in order to ensure that all bits of data have been presented according to the specified LoD. Once the data is ready for dissemination, the LoD drives aspects such as the exchange of data, materialisation (3D printing), streaming, and delivery; topics all relevant for interoperability. Since most workflows involve portraying the data, visualisation is also an important aspect where the LoD plays a prominent role in balancing the cognitive and performative aspects. Finally, after these steps have been completed, 3D city models are subject to maintenance and update in which the data is updated to reflect the latest real-world situation. In this process it has to be ensured that the refreshed data is of the same LoD as the original data. Eventually, the data is delivered to users, who employ it for an application where the LoD may affect the performance and reliability of a spatial analysis.

2.1.3 Geometrical and Topological Aspects

Geometric data is defined in one of three dimensions such as a single point (zero-dimensional), lines (one-dimensional), areas (two-dimensional) and volumes (three-dimensional) in space. In GIS a series of fundamental geometric primitives are used to represent city objects in a 2D or 3D space. These primitives include point (node), line (edge), and polygon (face). Points are described and stored with two numbers (x, y) coordinates in 2D space. While in 3D Euclidean space, every point can be determined by a triple coordinates (x, y, z) e.g. point cloud obtained from Terrestrial Laser Scanning or LiDAR etc. A line is a 1D shape that links two points with coordinates by a direct path therefore it is considered as a simplest type of segment. Volumetric geometries of 3D city models (e.g. buildings, trees etc.) are modelled using geometric primitives (points, edges, faces) and surfaces or solids is bounded by a closed surface (U. S. Baig, 2013).

In the literature, two major 3D abstractions are distinguished for modelling 3D objects and phenomena (Lattuada, 2006; Mäntylä, 1988): Surface-based and Volume-based. Constructive Solid Geometry (CSG) and Boundary representation (B-Rep) are typical examples of Surface-based representations, while voxel (regular space subdivision) is an example of volume-based representations.

B-Rep is the approach that is widely accepted for modelling discrete real-world objects and design (computer graphics) models (Hughes et al., 2014). The 3D object is represented by bounding low-dimensional primitives (vertex (0D), line (1D), polygon (2D), polyhedron (3D)), which are organised in data structures. The primitives can be either simple such as planar faces and straight edges or complex such as curved surfaces and edges. The main advantage of boundary representations is that they represent real-world objects as they are perceived by humans. The boundary of the objects can be obtained by measuring properties that are visible (i.e. "boundaries"). Main disadvantages of boundary representation include that the complexity is high and no unique data structure exists. The primitives may vary and i.e. can be a face (topologically described), a triangle or a polygon (geometrically described). Depending on the used primitive (triangle or polygon) different constraints can be enforced such as: 'polygons must be planar', 'orientation of polygons must be clockwise', etc. Although more complex, B-Rep is in the basis of most the file formats (models) used in GIS and CAD used for exchange of information. A way of representing 3D features in commercial GIS that begins to get attention in the GIS community is MultiPatch geometry, which is available in ESRI's software ArcGIS. A MultiPatch is a type of B-Rep which facilitates a polyhedron approach to represent a 3D object (Ford, 2004).

CSG is the most popular constructive representation. Its primitives are parameterized solids, which may be simple shapes (such as cylinders, cones, blocks) or more complex features suitable for a particular application domain (such as slots or counter-bored holes). The primitives may be instantiated multiple times (possibly with different parameter values, positions, and orientations) and grouped hierarchically. Primitive instances and groups may be transformed through rigid body motions (which combine rotations and translations) or scaling. The transformed instances may be combined through regularized Boolean operations: union, intersection, and difference (Hughes et al., 2014). The advantages of CSG models is in that they are good in Computer-Aided Manufacturing. The disadvantages for the use of CSG in real world modelling are that the objects and their relationships might become very complex.

Figure 2.6: Modeling of a simple building model using voxels (left), CSG (middle), and B-Rep (right) representations (Pfund, 2002)

A voxel is a regular 3D volume element. A 3D object is then represented as an array of voxels. Each voxel holds one (or more) data values. Voxel representation is appropriate for modelling of continuous phenomena such as geology, ocean, climatology, soil, etc. The benefit of voxels is in that they are regular in modelling and thus their management and analysis on them is simple (i.e. the volume is very easy to compute). A disadvantage of voxels is that high resolution data results in large volumes of data. Furthermore, the surface is always somehow "rough", which might result in unrealistic visualisations.

2.1.4 Semantic Aspects

Globally, the word semantics is used to refer to the improvement of an information system by making information more explicit, especially in relation to the meaning of things. Applied to models and their implementation, we can see different interpretations of what semantics is in information systems.

A semantic 3D city model refers to a 3D city model that contains semantic information as opposed to a geometric 3D model that would contain pure geometric information. Such models integrate attributes of objects (nature, usage, etc.), but they can also contain relationships between objects (topological, spatial, etc.).

Semantic 3D city models comprise besides the spatial and graphical aspects particularly the ontological structure including thematic classes, attributes, and their interrelationships. Objects are decomposed into parts due to logical criteria (and not due to graphical considerations) which follow structures that are given or can be observed in the real world (Kolbe, 2009).

The more information the model contains the more uses it may have. Nevertheless, more information also means more potential conflicts between attributes and relationships and more complex consistency issues. The construction and the update of information also become a complex challenge.

In order to describe such a model, a typification of urban entities meant to be described is necessary. This typification can take the form of a Unified Modelling Language (UML) class diagram, a formal logical ontology (e.g. in Web Ontology Language (OWL)), a database schema, an XML schema, etc. This category of model is called "type model" (Kühne, 2006)

Figure 2.7: The semantic 3D city model connects different aspects by supporting information exchange with the city model entities (Kolbe, 2009).

2.2 3D Modelling Techniques and Software

2.2.1 Building Information Modelling (BIM)

Building (Construction) Information Model (BIM) is defined by international standards as "shared digital representation of physical and functional characteristics of any built object [...] which forms a reliable basis for decisions" (ISO, 2016). According to (Eastman et al., 2011) who introduced BIM into the AEC/FM domain, the term "BIM" has two meanings. One means Building Information Model, while the other is Building Information Modelling. The former refers to virtual 3D building models containing rich building information, while the latter means the process of creating and processing 3D building models. BIM technology is subject to variation and confusion, especially by software developers when describing the capabilities that their products offer. From the modelling perspective, which is the main focus of this research, the kind of models that don't utilize BIM design technology are the following:

a) Models that contain 3D data only and no (or few) object attributes, these models can only be used for graphic visualisations and have no intelligence at the object level.
b) Models with no support of behaviour, that don't utilize parametric intelligence.
c) Models that are composed of multiple 2D CAD reference files that must be combined to define the building.
d) Models that allow changes to dimensions in one view that are not automatically reflected in other views.

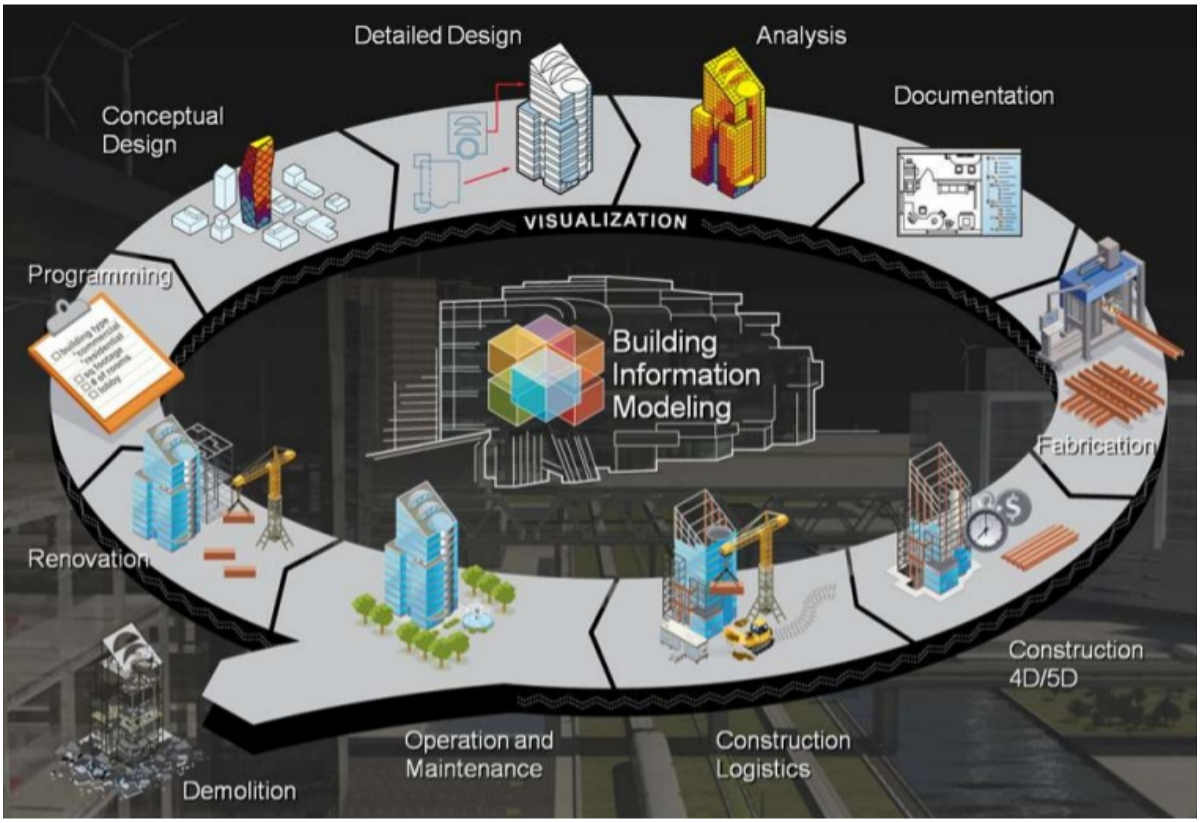


Figure 2.8: What is BIM? (Canada, 2012)

2.2.1.1 BIM Characteristics

BIM can be considered the latest generation of Object-Oriented Computer Aided Design (OOCAD) systems in which all of the intelligent building objects that combine to make up a building design can coexist in a single 'project database' or 'virtual building' that captures everything known about the building. The emergence of 3D CAD initially focused almost entirely on creating geometry in support of visualisation, and subsequent advances concentrated on creating realistic rendering and lighting effects. More recently, OOCAD replaced 2D symbols with building elements (objects), capable of representing the behaviour of common building elements. These building elements can be displayed in multiple views, as well as having non-graphic attributes assigned to them. The inclusion of parametric 3D geometry, with variable dimensions and assigned rules, adds "intelligence" to these objects, permitting the representation of complex geometric and functional relationships between building elements (Howell & Batcheler, 2005).

BIM has emerged as a method for creating, sharing, exchanging, and managing information among all stakeholders throughout project life cycle. BIM serves the AEC/FM domain by providing detailed 3D building models that could be used throughout the lifecycle of a construction project, including plan, design, construction, operation, and dismantling (Zhu et al., 2018). The logical structure and well-defined meaning of spatial objects of a building make it possible to go beyond visualisation.

In constructing BIM models a great amount of manual work is involved. They can be created by architects or civil engineers in the planning or constructing phases. Most of the BIM models today are available for only newly planned or recently constructed buildings. Recently research focus shifts from earlier life cycle stages to maintenance, refurbishment, deconstruction and end-of-life considerations especially of complex structures (Akbarnezhad et al., 2014; Volk et al., 2014)

Different works has been done to define the concepts of BIM models. Howell and Batcheler (2005) have been identified two approaches according to different software vendors who deal with the building industry:

a) the Transitional approach, where the building model is divided into groups of objects and central project database approach. These groups can be aggregated to form the complete view of a building.
b) the Central Project Database approach, where a central database is used to store a building model. The advantage of this approach is that the building of model parts can be organised and managed in one central database, although any modification or error appears in the whole model.

Additional to this classification, Isikdag et al. (2007) specify the special characteristics of BIM as follows:

a) Object-oriented: BIMs are mostly developed in an object-oriented fashion in order to facilitate the navigation and tracing processes through the model parts.
b) Data-rich/comprehensive: BIMs cover physical and functional characteristics of building parts. Therefore, they are data-rich and comprehensive.
c) Three-dimensional: In contrast to CAD, BIMs always represent geometries of buildings and their spatial objects in three dimensions.

d) Spatially-related: Spatial relationships between building elements are maintained in the BIMs in a hierarchical manner (allowing for several representations such as constructive solid geometry, sweeping and boundary representations),
e) Rich in semantics: BIMs are designed at the building scale. Therefore, they maintain a large amount of detail and semantic information about building parts and spatial relationships between their elements.
f) BIMs support view generation: BIMs usually have different views of the building based on user needs. These views can be generated from the base information model and can also be aggregated to form the bigger model as well.

One of the most developed and established semantic models that implements BIM concepts is the Industry Foundation Classes (IFC) standard. Today, there are several CAD/AEC applications (such as Archicad, AutoCAD and Bentley MicroStation) as well as business analysis applications (such as SAP 2000) that have the abilities to import and export their internal models according to the IFC standard (Isikdag & Zlatanova, 2009a). Drawing Exchange Format (DXF) and the Initial Graphics Exchange Specification (IGES) by Autodesk and a different joint initiative by Boeing and General Electric were amongst the initial efforts for the development of a drawing exchange format. However, IFC which will be analyzed in Section 2.3.3 is the most predominant one, due to the provision of exchange semantic information.

Indicative questions under research regarding BIM world are: Can a BIM be implemented in a geospatial environment?; What can be achieved by a possible implementation in site selection process?; What can be achieved by a possible implementation in fire response management process or in an evacuation procedure?

2.2.1.2 LoD in BIM

BIM has its own LoD concept as well. However, there is a certain degree of ambiguity when it comes to the LoD in BIM: some researchers state that BIM does not have an equivalent LoD as is in the GIS notion (Vilgertshofer, 2016) while others borrow the concept from GIS (Volk et al., 2014). Hence, LoD in BIM is subject to various interpretations (Bolpagni & Ciribini, 2016). However, it is of borderline relevance to LoD in GIS, as the 'D' in the LoD concept in BIM denotes "Development", thereafter LoDt. The AIA document E202 defines level of development for model elements. The LoDt is the combination of the graphical level of the object's geometry (LoD) and the nongraphical data in the object, which is called the Level of Information (LoI). The level of development provides the description of content and reliability of BIM models at various stages in the design and construction process, which do not reflect specific modelling guidelines for any particular software, rather a generic definition of model content and, more importantly, authorised uses of the model for the respective LoD. In 2008, the American Institute of Architects (AIA) published its first set of Level of Development definitions in AIA Document E202™-2008 Building Information Modelling Protocol (AIA, 2008). AIA G202 talks about five categories of level of development (100, 200, 300, 400, and 500). In 2011 the BIMForum initiated the development of this LoD Specification and formed a working group comprising contributors from both the design and construction sides of the major disciplines. The definitions of the five Level of Development in BIM industry are given in Table 2.3 and illustrated in Figure 2.9, according to 2017 Level of Development Specification Guide of BIMForum (BIMForum, 2017). L.A.H.M. van Berlo and Bomhof (2014) have worked on the refinement of the BIM LoDs after analysing industrial practices and conducting a series of geometric tests. In their research the need to develop the additionals LoDt 250 and LoDt 350 is highlighted.

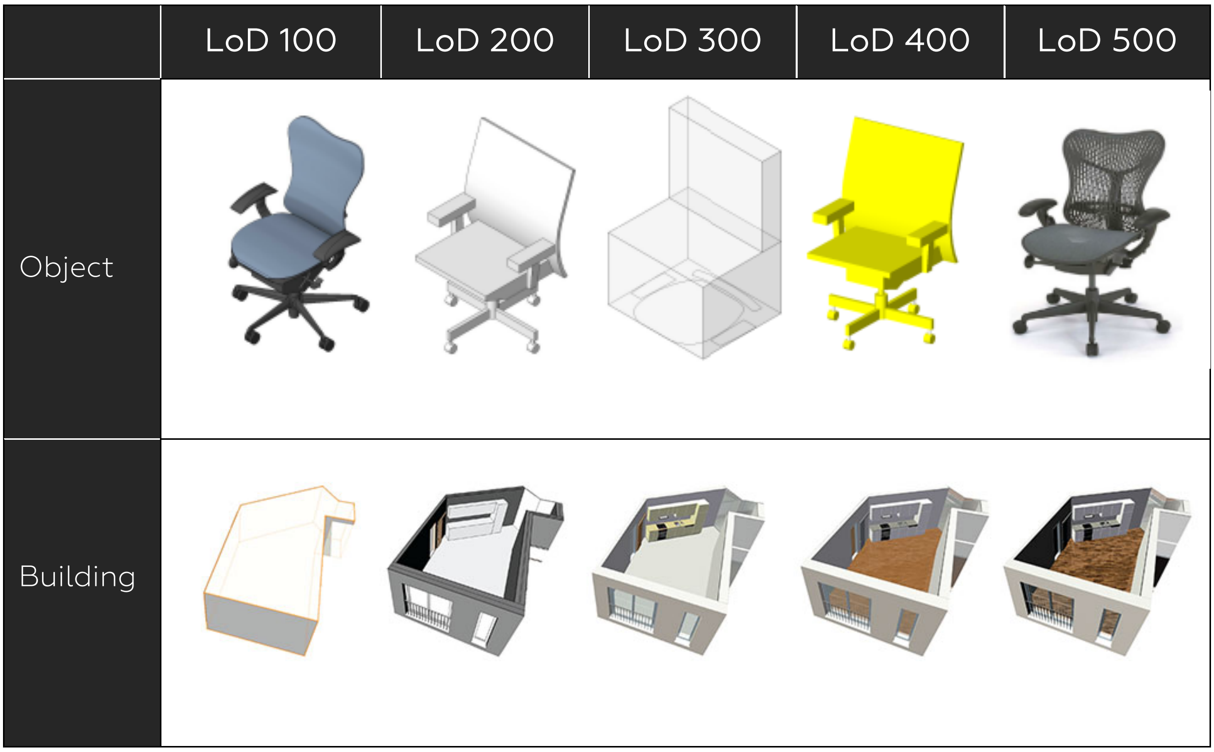


Figure 2.9: BIM LoDts for a building (PracticalBIM, 2013)

LoD	Description	AIA Definition	BIMForum Interpretation
100	Pre-Design	The Model Element may be graphically represented in the Model with a symbol or other generic representation but does not satisfy the requirements for LoDt 200. Information related to the Model Element can be derived from other Model Elements.	At this LoDt elements are generic placeholders. They may be recognizable as the components they represent, or they may be volumes for space reservation. Any information derived from LoDt 200 elements must be considered approximate.
200	Schematic Design	The Model Element is graphically represented within the Model as a generic system, object, or assembly with approximate quantities, size, shape, location, and orientation. Non-graphic information may also be attached to the Model Element.	The quantity, size, shape, location, and orientation of the element as designed can be measured directly from the model without referring to non-modeled information such as notes or dimension call-outs. The project origin is defined and the element is located accurately with respect to the project origin.
300	Design Development	The Model Element is graphically represented within the Model as a specific system, object or assembly that is accurate in terms of size, shape, location, quantity, and orientation with detailing, fabrication, assembly, and installation information. Non-graphic information may also be attached to the Model Element.	Parts necessary for coordination of the element with nearby or attached elements are modeled. These parts will include such items as supports and connections. The quantity, size, shape, location, and orientation of the element as designed can be measured directly from the model without referring to non-modeled information such as notes or dimension call-outs.
400	Construction Stage	The Model Element is graphically represented within the Model as a specific system, object or assembly that is accurate in terms of size, shape, location, quantity, and orientation with detailing, fabrication, assembly, and installation information. Non-graphic information may also be attached to the Model Element.	An LoDt 400 element is modeled at sufficient detail and accuracy for fabrication of the represented component. The quantity, size, shape, location, and orientation of the element as designed can be measured directly from the model without referring to non-modeled information such as notes or dimension call-outs.
500	As Built	The Model Element is a field-verified representation in terms of size, shape, location, quantity, and orientation. Non-graphic information may also be attached to the Model Element.	Since LoDt 500 relates to field verification and is not an indication of progression to a higher level of model element geometry or non-graphic information, this Specification does not define or illustrate it.

Table 2.3: Definitions of LoD in BIM (BIMForum, 2017)

2.2.2 Geographic Information Modelling

The term "Geographic Information System (GIS)" was first used by Roger Tomlinson in 1969 in his paper "A Geographic Information System for Regional Planning" (Tomlinson, 1969). The major difference between GIS and other information systems is that the GIS data are geo-referenced. Normally, the spatial information represents the reality through an abstract model (M. Li et al., 2007) and includes coordinates, the spatial relationship between features and additional non-spatial attributes. The key research topics of GIS can be summarized as: locations, conditions, trends, patterns and models.

Narrowly speaking, GIS is a platform made up with hardware, software, spatial data, and system manager, with various toolsets for heterogeneous data, especially spatial data, integration, storage, manipulation, analysis, and visualisation, to reveal patterns, trends, and relationships that might not be directly seen from the original form (Zhu et al., 2018).

GIS as a technology/system allows the storage of spatial information in the relational database, and, as a science, is also beyond data storage system. The attribute information associated with spatial features stored in the database allows for further spatial and temporal analysis using both the spatial and non-spatial attributes. GIS is able to implement spatial analysis based on the functional and physical spatial relationship of outdoor environment at large spatial scale; however, it lacks detailed and comprehensive digital repository of building information.

No more than a few years ago, geospatial information systems were totally different from their shape today. As their main purpose was to define the urban and regional scale, the details wasn't the main focus of geospatial models. Today, however, it is very important to model different focused areas around and between buildings in the streets as well as urban furniture. Therefore a real 3D representation is a definite requirement compared with the 2D/2.5D representation in the last decade. Following this change, two key issues need to be considered in representation of the urban environment in 3D. First, more advanced building geometric information models should be constructed and, second, integration rules and framework between building models and the geospatial environment models should be developed (Claus Nagel et al., 2009).

As a result of the fast technological development in the last decade, geospatial models have become increasingly important for modelling the world around us, taking into consideration the third dimension. Applications for regional or urban areas require the modelling of vast areas. Therefore, geospatial models use simplified but efficient geometric methods in order to represent a large number of spatial objects. They usually use simple geometric representation for building parts and spatial objects by sweeping and boundary representation (B-Rep) methods (Isikdag & Zlatanova, 2009a). One of the major difficulties in geospatial information modelling is the collection of data for vast regional and urban areas. Different approaches have been found in the literature dealing with the acquisition of sufficient data for building geospatial models.

2.2.2.1 <u>LoD in GIS</u>

In GIS community various attempts have taken place to standardise the LoDs to differentiate datasets. NAVTEQ, an American company that provides geo-information data for navigational products and BLOM, a Geomatics company from Norway, offer four LoDs for their 3D city modelling products ranging from simple wire frames to fully textured models. However, the standard CityGML contains one of the most prominent LoD categorisations. Since its inception, this LoD categorisation has become the primary LoD standard. A comparison table (Table 2.4) between different LoDs paradigms is presented by (Biljecki, 2013). These different approaches show that in 3D city modelling there is not an uniform approach to what defines an LoD, and what drives their definition. The levels are generated according to a specification, and are used regardless of the application or the context, e. g. they are equal whether the 3D city model is used for estimating the solar potential or for urbanism applications, or if they have a specific application they are used only for one application.

	No. of LoD	Exterior-driven	Texture-driven	Interior-driven
CityGML	1+4	Yes	No	Yes
BLOM	4	Yes	Yes	No
NAVTEQ	4	Yes	Yes	No

Table 2.4: Comparison between different levels of detail paradigms (Biljecki, 2013)

CityGML version 2.0 defines five LoD (OGC, 2012b). The concept is intended for several thematic classes of objects but it is primarily focused on buildings. The five described instances increase in their geometric and semantic complexity. A more detailed view on the different LoD is following, according to the specification guide of OGC.

LoD0 is a representation of footprints and optionally roof edge polygons marking the transition from 2D to 3D GIS. In LoD0 there are no volumetric representations. Subsequent LoD are improving in terms of the complexity of objects in the geometric and semantic sense. LoD1 is a coarse prismatic modelled usually obtained by extruding an LoD0 model. LoD2 is a model with a simplified roof shape, and where the object's parts can be modelled in multiple semantic classes (e.g. roof, wall). LoD3 is an architecturally detailed model with windows and doors, being considerably more complex than its preceding counterpart (Kolbe, 2009). LoD4 completes an LoD3 by including indoor features but otherwise it retains virtually the same properties as LoD3. This taxonomy has been developed in the German Special Interest Group 3D (SIG 3D) initiative (Albert et al., 2003; Gröger et al., 2005; Gröger et al., 2004) has been further described in (Gröger & Plümer, 2012). The textures can be added to any LoD (i.e. the texture is not part of the LoD specification), and the standard includes other thematic classes, e.g. roads and vegetation which have their LoD description, but they are less prominent and their acquisition is not prescribed for each LoD. The LoD requirements of CityGML 2.0 are presented in 2.3.2.

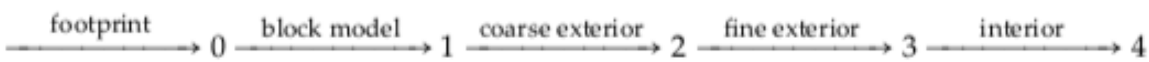


Figure 2.10: Relation between CityGML LoDs (Biljecki, 2013)

	LoD0	LoD1	LoD2
Model scale description	Regional/landscape	City, region	City, city districts
Class of accuracy	Lowest	Low	middle
Absolute 3D point accuracy (position/height)	Lower than LoD1	5/5m	2/2m
Generalisation	Maximal generalisation	Object blocks as generalized features: >6*6m/3m	Object as generalized features: >4*4m/2m
Building Installations	No	No	Yes
Roof structure/representation	Yes	Flat	Differentiated roof structures
Roof overhanging parts	Yes	No	Yes, if known
CityFurniture	No	Important objects	Prototypes, generalized objects
SolitaryVegetationObject	No	Important objects	Prototypes, higher objects
PlantCover	No	>50*50m	>5*5m

Table 2.5: The LoD requirements by CityGML 2.0 (Gröger & Plümer, 2012)

However, mainly due to the emergence of new applications of 3D city models, the current LoD concept of CityGML is no longer flexible enough. Hence a number of new approaches have been proposed. They range from more practice-oriented (C. Nagel, 2014) to very detailed (Benner et al., 2013; Löwner et al., 2013) or even go beyond the context of CityGML(Biljecki et al., 2016b; Biljecki et al., 2014; Biljecki et al., 2013). Löwner et al. (2013) distinguishes the original concept to a Geometrical Level of Detail (GLoD) and a Semantical Level of Detail (SLoD). The authors of this paper propose that these two LoDs should be further characterized for the interior features and the outer building parts. (Benner et al., 2013) conduct a comparative study on various LoD concepts, and identifies the shortcomings of the current CityGML LoD concept for 3D city modelling. (Biljecki, 2017; Biljecki et al., 2016b) provide a more detailed concept of the LoD, specifically provides a series that contains 16 LoD (4 refined LoD for each of the LoD0−3). After all, the preponderant approach is from (Löwner et al., 2016), which will be the base of City GML 3.0 LoD concept.

In **CityGML version 3.0** the concept of LoD will be different. (Löwner et al., 2016) have presented a multi representation concept (MRC) that enables a user-defined definition of LoD. The main barrier in the current concept of LoD, according to the authors, is that the building's interior can only be represented if the exterior shell is represented in LoD4, which implies the highest semantic complexity and geometric detail. This definition hinders applications that require detailed information on the building's interior structure without geometrically exact representation of the exterior shell (e.g. firefighting, emergency operations or indoor navigation). Thus, the possibility to combination of a rough LoD1 or LoD2 model of the exterior shell with a detailed interior model would be beneficial and notably cost-effective. Further, indoor navigation may require the representation of floors, rooms and other interior objects in coarse but not detailed LoD. However, one level (LoD4) is defined for interior features. In CityGML 3.0, LoD4 is replaced by LoD0 – LoD3 for indoor features. For testing and demonstration purposes, the UML of the multi-representation property was actually implemented, by the authors as an ADE of CityGML 2.0 (Figure 2.11)

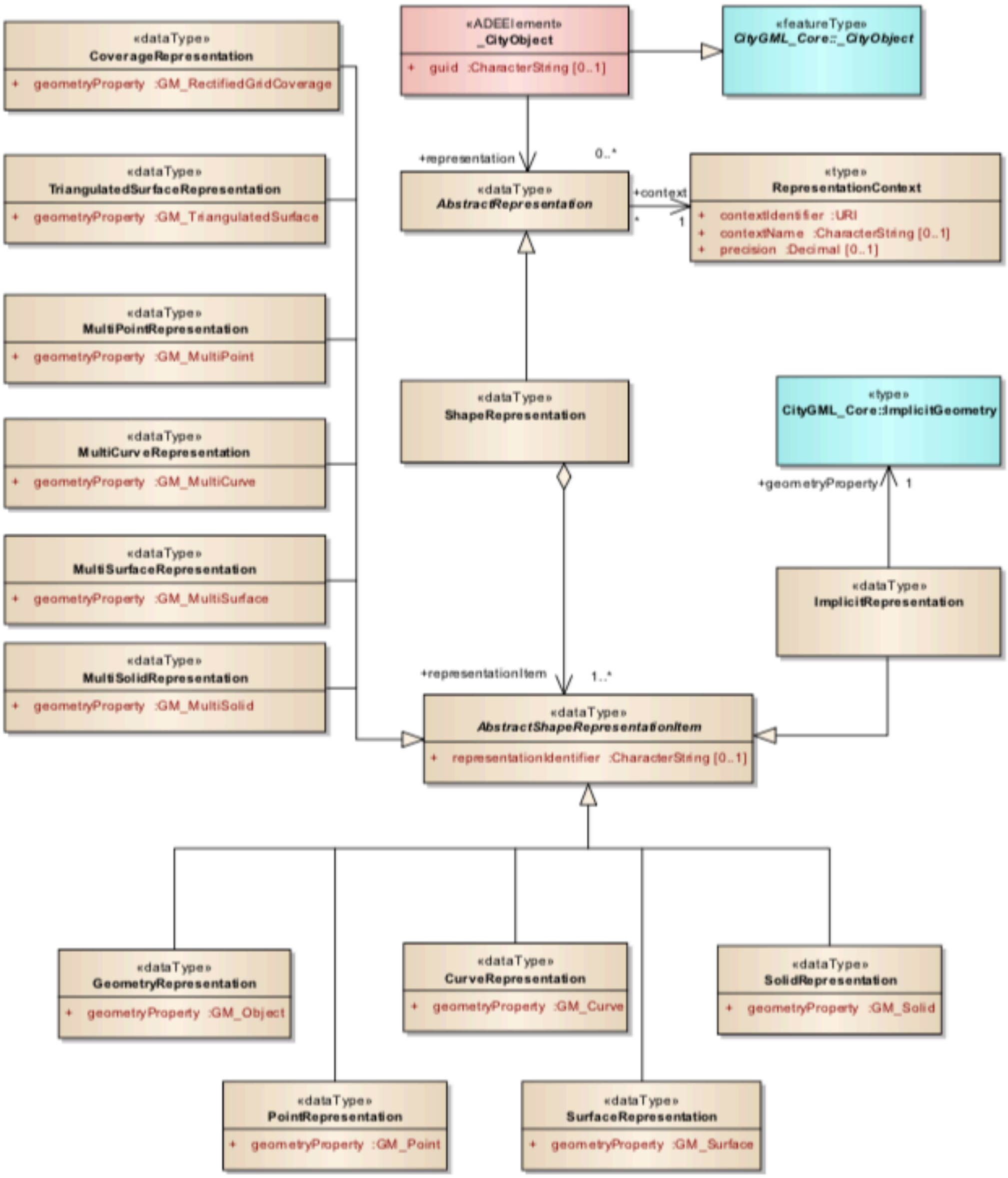


Figure 2.11: UML class diagram of the proposed multi representation concept as a CityGML 2.0 ADE (Löwner et al., 2016)

The LoD concept they have presented allows multiple geometric representations for any CityGML object. In their proposal every feature in CityGML (CityObject) can in principle be represented multiple times by any geometry type: (Multi)Solid, (Multi)Surface, (Multi)Curve, Point, or implicit representation. Profiles can be defined based on the multi- representation concept in order to maintain interoperability of CityGML.

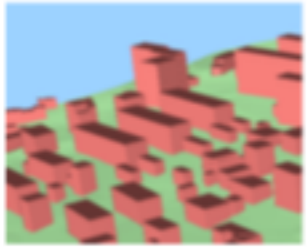
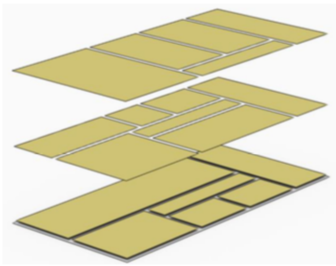
	LoD0	LoD1	LoD2	LoD3	LoD4
City General Concept (CityGML version 2.0) (OGC, 2012b)					
Building (CityGML version 2.0) (OGC, 2012b)					
Building (CityGML version 3.0) (Löwner et al., 2016)					–

Figure 2.12: CityGML V2.0 & v3.0 LoDs of building module (Biljecki et al., 2016b; Löwner et al., 2016; OGC, 2012b)

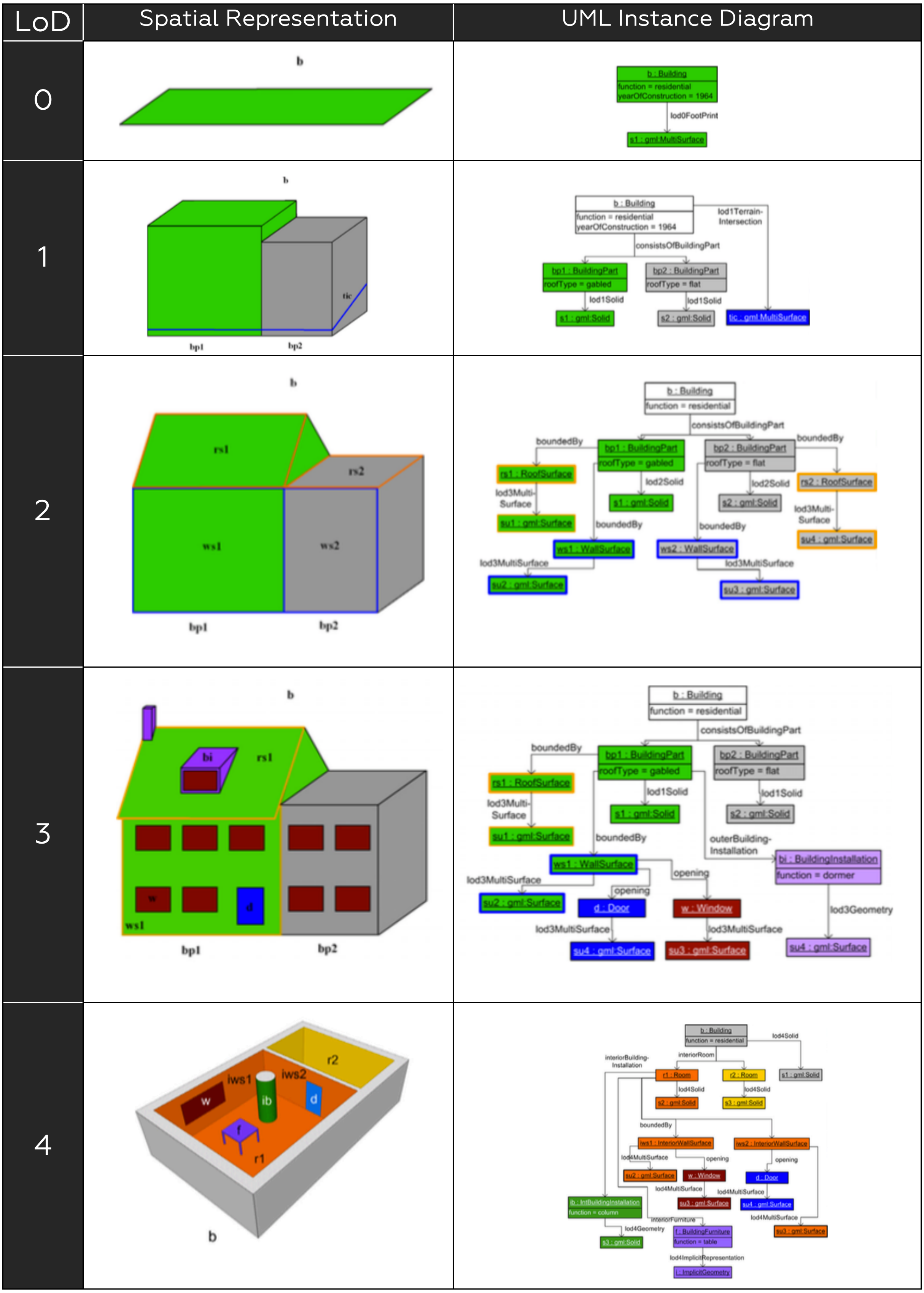


Figure 2.13: Illustration of CityGMl LoDs 0,1,2,3,4 of building module as a spatial representation and as UML instnce diagram (Gröger & Plümer, 2012)

2.2.3 Differences and similarities between BIM and GIS

BIM and GIS originate from different domains, as previously mentioned, one from AEC/FM, another from geospatial science and are quite different. One is for detailed 3D building model creation and sharing, another for geospatial data and non-geospatial data management and analysis.

Generally, their dissimilarities and mismatches can be summarized as: different users, different application focuses, different development stages, different spatial scales, different coordinate system, different semantic and geometric representations, different levels of granularity, and different information storage and access methods. Although these two concepts/ technologies have matured in different ways, the overlap between them has become bigger recently (Figure 2.14). With the recent demand for merging outdoor and indoor applications for different purposes, attempts have been made to design methods and tools to integrate building models within a geospatial context. However, barriers coming from these dissimilarities and mismatches exist for the integration between the two different concepts and their respective enabling systems.

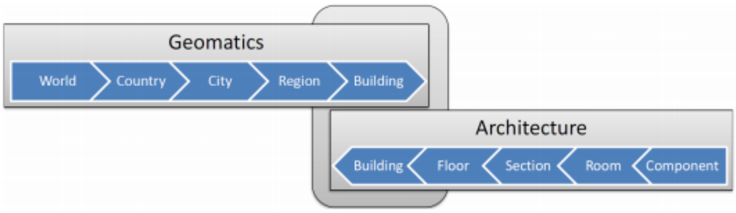


Figure 2.14: Overlap between the fields of GIS and BIM (Liu et al., 2017)

Regarding the most used models that have been developed for these domains (IFC and CityGML): IFC focuses on the building modelling aspect, in contrast to CityGML, which focuses on the city modelling aspect. An important difference between IFC and CityGML regarding data conversion is the modelling of buildings and building elements like walls, slabs or roofs. IFC usually uses IfcBuilding as pure container for building elements. Building elements are objects with properties, relations, and usually volumetric geometry representations. In contrast, CityGML allows Building to have an explicit solid or surface geometry. Building elements are not modeled as objects but as boundary surfaces for buildings or rooms without any further property or relations to other boundary surfaces. This means that the volumetric representation of building elements in IFC has to be converted into boundary surfaces of the CityGML building or room, as illustrated in Figure 2.15.

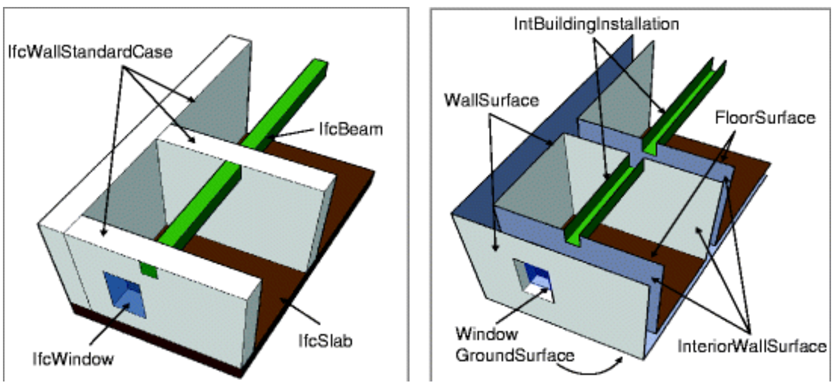


Figure 2.15: Building component representation in IFC [left] and surface representation in CityGML [right] (S. Amirebrahimi et al., 2016)

However, the next versions of the standards (CityGML 3.0 and IFC5) will eliminate the gap, due to the new concept of LoDs in CityGML 3.0, that includes the interior structure of a building from LoD1 and the expansion of IFC in the other infrastructures that are already part of CityGML (IfcTunnel, IfcRoads, IfcRailway, IfcBridge).

A comparison table (Table 2.6) between these domains, is presented below, on the basis of six criteria:

Criteria	Computer Graphics	BIM	GIS
Geo-reference	Local 3D Coordinates, some exchange formats support geo-referencing	Limited support	Always
Geometry	Parametric primitives; Boundary representation	Parametric primitives; B-Rep; CSG; Sweep volumes	B-rep; Multipatch
Topology	Limited	Yes	Yes
Semantics	No	Explicit within BIMs	Fully
Material/ Appearance	Fully	Limited	Limited
Exchange formats	GeoVRML, X3D, KML, COLLADA, OpenFlight	IFC	GML

Table 2.6: Comparison table for BIM and GIS

	Virtual 3D City model (CityGML)	BIM (Revit/IFC)
Graphical illustration		
Information details	• Surfaces (wall, floor, roof) • Installed objects (windows, doors) • Edges and object demarcation lines • Visible internal details (e.g. floor slabs)	• Surfaces (wall, floor, roof) • Installed objects (window, doors) • Edges and object demarcation lines • Visible internal details (e.g. floor slabs) • Detailed outline for edges of objects (e.g. frame around windows) • Partition lines for surfaces of an object (e.g. double doorpanels, window panes) • Can be used for constructionpurposes as dimensions are accurate

Figure 2.16: Differences between LoD 4 for CityGML and the BIM counterpart (Tah et al., 2017)

2.2.4 Modelling Tools and Software

3D modelling in Geoinformatics refers to the description of real world entities and phenomena, with BIM and 3D GIS being the two main pillars for the modelling, exchange and simulation of 3D objects. Literature provides a variety of 3D modelling techniques, to name the leading ones:

- Image based rendering: realistic visualisation of 3D objects and scenes, without reconstruction of the corresponding 3D models (Chan et al., 2007). This technique is based on photogrammetry and in particular aerial, satellite and close-range photogrammetry (Singh et al., 2013)
- Range based modelling: usage of a dense and clean 3D point cloud of the object's surface, which can be derived by a Terrestrial or Aerial Laser Scanner, UAV or Mobile Mapping System.
- Hybrid methods: Image and range-based modelling: high quality, detailed and textured 3D models mostly used at the of archaeological cultural heritage field (Brutto & Spera, 2011)
- Procedural modelling: usage of CGA shape grammar to programmatically produce building models with high visual quality and geometric detail (Xie et al., 2013)

Different modelling approaches are employed in 3D GIS, computer graphics, AEC/FM areas. The last decade BIM provides an important means at the AEC/FM field as a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life cycle, from earliest conception to demolition. BIM and GIS interpret 3D modelling from two different perspectives: GIS focuses more on multi-scale real world modelling driven by the requirements of mapping tasks from a geographical perspective, while BIM (and the AEC/FM domain) is more focused on the detailed design process and the representation of construction details of 3D facilities (El-Mekawy et al., 2012; Liu et al., 2017). What is more, planning at the built environment requires at least two different levels of planning process and modelling (Jusuf, Mousseau, Godfroid, & Soh, 2017): the city/neighbourhood-scale and the building-scale. GIS and CityGML are the leading applications for the first scale, while for building-scale, BIM and in particular IFC standard are used.

To support the afore mentioned applications, multiple 3D modelling platforms and tools that meet different requirements related to the available data, the desired application and their cost have been developed. Several platforms/tools are more technologically oriented (Web, servers, etc.) and support 3D data process and storage (databases, format-based), others focus on the data models supporting multiple geometries (B-Rep, voxels, etc.), some have good geo-reference characteristics supporting multiple coordinate systems and their transformations, others are targeted to visualisation, etc. In the last decade there have been various successful academic and industrial efforts to create a strong and robust synergy between BIM and GIS, specifically focusing on integration processes between IFC and CityGML.

Today, BIM lies is used in multiple domains: Architecture, Structural Engineering, MEP, Facility Management, Construction Industry, etc. and thus several BIM-compatible software suites and platforms have been developed to serve those fields. The different marketing strategies lead to packages with different collections of functionality. At the following table, the most common-used 3D modelling platforms are listed, indicating whether or not they are BIM-compatible (support of IFC import/export functionality).

SOFTWARE/ PLATFORM	(COMMON) FILE TYPES SUPPORTED	IFC
Autodesk AutoCAD Civil 3D	DWG, DWF, DXF, SDF, FBX, ESRI SHP, ESRI ARCINFO, GML, MIF, CATPART, IGES, JT, X_B, X_T, TAB, DGN, SQLITE, FT, 3DS, 3DM, SAT, CATPART, SLDPRT, STP, PRT, TIFF, BMP, PNG, JPG, PDF, WMF	√
Autodesk Architecture	DGN, DWG, DWF, DXF	√
Autodesk Navisworks	DWG, DWF, DXF, DGN, IGES, SAT, CATPART, STP, FBX, IPT, JT, PRT, DRI, X_B, PRT .ASM, NEU, RVM, RVT, SKP, SLDPRT, STP, STEP, STL, WRL, PDF, 3DS, 3DM	√
Graphisoft ArchiCAD	DWG, DWF, DGN, BFC, C4D, 3DM, 3DS, ATL, KML, KMZ, SKP, DAE, EPX, FACT, OBJ, STL, GIF, BMP, JPEG, JPG, PNG, TIFF, PDF, ASCII POINT FILES (TXT, CSV, etc.), XLS, WMF, EMF	√
Autodesk REVIT Suite (Revit Architecture; Revit Structure; Revit MEP, etc.)	DGN, DWF, DWG, DXF, IFC, FBX, SAT, SKP, RVA, BMP, PNG, JPG, AVI, PAN, IVR, TGA, TIF, ODBC, HTML, TXT, MDB, XLS, gbXML, GSM, IES, ASCII POINT FILES (TXT, CSV, etc.).	√
Bentley Suite (Bentley Architecture; Bentley Structures; Bentley HVAC; Microstation, etc.)	DWG, DXF, XTM Parasolid, ACIS SAT, STEP, IGES, 3DS, 3DM, SKP, STEP, STL, OBJ, JPEG, BMP, AVI, VRML, PDF, KML, DAE, U3D, CGM, AP203/AP214, ESRI SHP.	√
SketchUp Pro	DWG, DXF, FBX, JPG, PNG, TIF, BMP, PSD, TGA, EPS, PDF, STL, DAE, 3DS, DEM, KMZ, OBJ WRL, XSI	√
Rhinoceros	DWG, DXF, FBX, 3DS, AI, EPS, GF, IGES, LWO, GDN, SCN, OBJ, PFF, PLY, ASCII POINT FILES (TXT, CSV, etc.), RAW, SKP, SLC, STP, STL, VDA, VRML, GDF, ZPR, DAE, GTS, KMZ, PDF, WMF	√
Nemetschek All Plan	DWG, DXF, DWT, DXB, DGN, PDF, C4D, SKP, 3DM, STL, WRL, XML, PLT, REB, RE1, RE2, ASF, CPIXML, DAE, KMZ, STL, U3D, PLT, SVG, 3DS, JPG, TIFF, BMP, EPS, TGA, PCT, PCX, PNG, PSD, XLS	√
Tekla Structures and Tekla BIMsight	DWG, DXF, ISM, BCF, ABS, 3DD, IGES, CALM, STP, STEP, CPIXML, FA,	√

SOFTWARE/ PLATFORM	(COMMON) FILE TYPES SUPPORTED	IFC
	IGES, LANDXML, ELO, KSS, SKP, TBP, TCZOP, CXL, TFL, UNI, PLM, DGN, PDMS, STD, SDT, ED PDF, ASCII POINT FILES (TXT, CSV, etc.).	
ESRI CityEngine	DXF, FBX, ESRI SHP, OBJ, DAE, FGDB, KML, KMZ, OSM	
Intergraph CADWorx	DWG, DXF, DRV, MDB, IDF, PCF, DGN, SAT, CATPART, STP, PRT, PSM, SLDPRT, IGES, DTM	√
Digital Project (DP)	CIS/2SDNF, STEP AP203 and AP214, DWG, DXF, VRML, TP, STL, CGR, 3DMAP, SAT, 3DXML, IGES, STL, HCG.	
Vectorworks Architect	DWG, DXF, DWF, C4D, EPS, BMP, GIF, JPG, PNG, TIF, ICO, WDP, DDS, SGI, TGA, EXR, PSD, ICNS, EMF, PCT, IGES, OBJ, SAT, STL, SKP, LAD, LAZ, XYS, ERSI SHP, TFW, X_T, 3DS, ASCII POINT FILES (TXT, CSV, etc.).	√

Table 2.7: 3D Modelling Software

2.2.5 Drivers for Integration of BIM and GIS

The motivation for merging the two systems arises from both the GIS and AEC domains and the deficiencies that they have.

From the perspective of AEC domain, BIM is focused on every building-related activity, including plan, design, construction, operation, and demolition. Apart from modelling buildings, the environment is also closely involved in the construction processes. Apart from modelling buildings, the environment is also closely involved in those construction processes. For instance, in the planning phase, the location of a building is to be determined considering various environment factors, such as light, terrain, and heat; in the construction phase, weather conditions are monitored, as they may affect the construction progress and safety environment; and before demolition, the environmental impact should be fully investigated. Unfortunately, BIM cannot handle those data efficiently. Another reason BIM will benefit from GIS capability is that it needs some of the rich spatial analysis functions from GIS to extend its capability, such as distance calculation for construction material supplier selection.

From the perspective of GIS, it was initially focused on 2D data, and its capability in 3D data has been limited. For example, its ESRI desktop GIS applications, ArcScene and ArcGlobe, can only create 3D models by extruding 2D drawings, and only simple editing functions are provided. The models created could reach LoD1 of CityGML. Therefore, the best practice for making 3D models for GIS is still using CAD or BIM software, such as Revit and SketchUp. The detailed 3D models created in BIM could not only help GIS extend its scope by applying spatial analysis at a finer scale, using building models to all those applications of Table 2.1. The most important contribution of BIM to GIS would be providing detailed 3D building models, as well as their rich building information. All of these aforementioned examples would be hard to realize for GIS without BIM.

BIM and GIS are complementary. GIS practitioners are helping BIM link to the outside world, and BIM practitioners are introducing GIS to the indoor environment. They could achieve much more in combination rather than by working separately.

2.3 Review of Existing Standards and Exchange Formats related to 3D City Modelling

Interoperability may be defined as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged. Standardisation is the most efficient and global solution to interoperability problem, in other words standards is an agreed way of doing something. In (UNIDO, 2006) approach, the role of standards can be labelled by the '3C' (competitiveness, conformity and connectivity).

Several organisations, industry consortiums and communities are involved in standards development activities related to urban matters related to 3D Information Models (International Organisation for Standardisation (ISO) TC/211, OGC, BuildingSMART Alliance (formally called IAI), Web 3D Consortium (W3D), 3D Industry Forum (3DIF), Open Design Alliance, Khronos group). ISO and OGC are the two dominant standards organisations. ISO is an independent, non-governmental membership organisation and the world's largest developer of voluntary International Standards. OGC is also an international not-for-profit voluntary organisation committed to making quality open standards for the global geospatial community. It is consisting of more than 525 member organisations, 48 open standards, 230 OGC certified products.

A range of 3D Standards is available concerning firstly, 3D data models and data exchange formats and secondly the web services for data access, registry & visualisation. Since this research is focused on the modelling aspect, only the former will be analysed.

2.3.1 3D Exchange formats and standards

The characteristics that distinguish data sharing from data exchange are the centrality of data and the ownership of that data. In the data exchange model, one software system maintains the master copy of the data internally and exports snapshots of data for other users. In the sharing model, there is a centralized control of ownership and there is a master copy of data. In theory, the data sharing model facilitates the revision control issue associated with the data exchange model (Isikdag et al., 2007).

Much effort is made in the AEC/FM and Geospatial domains to address interoperability issues via standardising exchange formats. These efforts evolved from simple file and drawing exchange formats to those product and domain-modelling initiatives that were followed by the object-oriented approaches to address the interoperability challenges (Isikdag et al., 2007). Zlatanova et al. (2012), Kolbe (2009) and Eastman et al. (2011) have presented some of the most comprehensive works regarding the exchange of 3D data in BIM and GIS domains.

Zlatanova et al. (2012) classifies the standards created for data exchange in three categories:

1) Those have been developed as standards by international organisations (IFC, CityGML, VRML, X3D);
2) Those have been developed by vendors, but due to their wide use they are accepted as standards (KML);
3) Those that have become de-facto standards due to their wide acceptance by users and software vendors (SHP, DXF, COLLADA, 3D PDF).

A short review of 3D file formats is following.

X3D file format: is an enhancement of VRML, in fact, this file format is even less used than VRML. The VRML was released in 1995 and accepted as a standard by the Web 3D consortium. It was designed as a web standard for exchange of graphics while providing possibilities for interaction with 3D world. Practically it is a language for modelling 3D realistic scenes and interaction. However, due to its disadvantages, the Web 3D consortium has stopped the development of the standard in 1998 and focused on X3D format. Only few viewers are available for visualisation of data on Internet and almost no vendor supports export and import of X3D. Some experts suggest that the format became too complex for interpretation.

KML is an XML format for geographic annotation and visualisation of objects inside Google Maps and Google Earth. At that time KML was adopted as standard by the OGC. The support of geometry in KML is aligned with GML. KML is suitable for Web applications and widely accepted, but geometries are represented without semantics.

COLLADA (stands for COLLAborative Design Activity for establishing an interchange format of 3D interactions) is another open standard for describing 3D data. Originally, this standard is from Sony (used for Playstation). Google uses this standard frequently (it is the core of all 3D objects in Google Earth and a key part of Google SketchUp) which has greatly increased the use of COLLADA. It is an XML-based ISO-adopted standard for the exchange of 3D digital assets among different interactive 3D software applications (Barnes & Finch, 2008; Khronos Group, 2017). It uses an object-based approach, has B-Rep geometry with limited support for tessellation, supports custom object-based semantics insofar as they relate to geometry, and supports multiple LoDs. COLLADA supports use of a local engineering CRS and one geodetic plus vertical CRS (WGS84 and WGS84-EGM96). COLLADA provides possibilities for describing geometry, topology and texture, but has no semantics.

GML (Geography Markup Language), also known as ISO19136, was created by the OGC and is an XML structure for the representation of geographic (spatial and location) information. It is a typical example of a standard created for the exchange of data. GML3 has a modular structure, which allows to select the schemas or schema components that are needed for a specific application. The geometry model of GML follows the ISO19107 standard and therefore GML provides classes for 0D to 3D geometric primitives, 1D-3D composite geometries (e.g. CompositeSurface), and 0D-3D geometry aggregates (e.g. MultiSurface or MultiSolid) consisting of geometries which are not connected by common boundaries. GML 3 includes support for spatial and temporal reference systems, topology, dynamic features, units of measure, metadata, gridded data, and is designed to be semantically extendable. CityGML is based on GML3 and all the geometry of GML are inherited by the semantic extension.

DXF has been created by Autodesk, December 1982 as part of AutoCAD 1.0, and has grown through the years as one of the most used formats for exchange of 2D and 3D CAD drawings. The file structure is ASCII-based and well described. Specifications for DXF are available on the web site of AutoDesk. The file formats support many different geometries (simple and complex), layers and drawing attributes. Since it has been designed as drawing Interchange format, it does not support thematic attributes. DXF is not designed for web, but tools have been provided by some vendors for web- based visualisation. The file format is mostly used to exchange data from one software package to another. Topology, texture, objects, LoDs are not explicitly supported, although different methodologies have been proposed by various users for the export of the aforementioned information in DXF.

The shapefile (SHP) is created by the Environmental System Research Institute (ESRI) in 1998 and is a typical example of a GIS file format. It supports simple geometry (i.e. OGC point, multi-point, polygon, polyline and since ArcGIS 10 polyhedron), including vendor specific data types (i.e. multi-patches). SHP is binary file format and adapted for faster drawing speed and editing capabilities. The major advantage of this file format is that the objects are kept with their thematic attributes. The file format consists of three files: main file: *.shp; index file: *.shx and DBase file: *.dbf. Shapefile is the most commonly used format for GIS data. There are many tools and applications that can read and export SHP file. SHP does not have a topological structure and the representation of textures is very basic.

A very interesting new development is the 3D PDF. The intention of this file format is to publish and share 3D design information within a normal PDF file format. Several Large CAD vendors (Bentley Systems, Autodesk, SketchUp through CityEditor) allow export to this file format. The 3D geometry is exported in a PDF format and can be integrated in a text document using Adobe software. The 3D model can be the explored by a special tool box for interaction (developed by Bentley Systems).

In conclusion, the 3D file formats have been described above can be classified in two broad categories: a) the data files that are for general purpose and b) those that are intended to exchange information. In the latter only CityGML and IFC can be included, which will be explained in more detail in sections 2.3.2 and 2.3.3

2.3.2 CityGML

Following the need for an open standard for a wide use of urban applications, CityGML has been developed as a common semantic information model representing different 3D urban and geographical objects that can be shared among different applications (OGC, 2012b). CityGML has been developed as an open data model with XML-based format that can be used for storing and exchanging virtual 3D objects and city models among applications. As an open standard, it has been implemented as an application schema for the Geography Markup Language 3 (GML3.1.1), the extendible international standard for spatial data exchange developed and issued by the OGC (OGC, 1994-2010) and the ISO TC211.

CityGML also goes beyond the representation of only graphical appearance and specifies classes and relations for the most relevant topographical objects in cities and urban models. It specifies 'City' to include not just its built structures, but also its elevation, vegetation, water bodies and more physical objects. Thus, CityGML allows sophisticated analysis, overlay tasks, decision support and thematic inquiries. Moreover, it represents semantic, thematic, taxonomical and aggregation properties of the models (Kolbe & Bacharach, 2006).

2.3.2.1 Overall Architecture of CityGML Data Model

CityGML is not just restricted to modelling buildings, but covers all relevant features within urban areas. Hence, the term 'City' in this context is broadly defined. Each of those features has a semantic definition, attributes, relationships and a 3D spatial representation. The features are organized into modules which can be arbitrarily combined for a specific application and to create a "profile".

In Figure 2.17 the different thematic modules of CityGML are presented. The vertical modules provide the definitions of the different thematic modules. The horizontal modules define structures that are related or can be applied to all thematic modules. This structure allows for partial CityGML implementations on the one hand and easy extension by further thematic models on the other hand (Kolbe, 2009).

The core module defines the base classes for all features in CityGML that have common attributes (creation date, deletion date, etc.) and which are inherited by all features in CityGML. All features defined in the modules have attributes and geometrical representations in different LoDs. Attributes common to all feature types are, e.g., a coarse classification, a function, and a usage (if the actual usage differs from the intended function). Also common to all feature types is the attribute relativeToTerrain that explicitly represents the relation of a feature to the terrain surface. This relation is useful for many applications, that queries involving the relation to the terrain are facilitated.

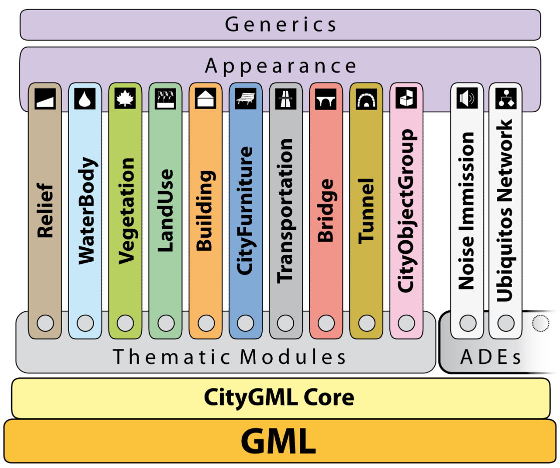


Figure 2.17: Modularisation of CityGML 2.0 (OGC, 2012b)

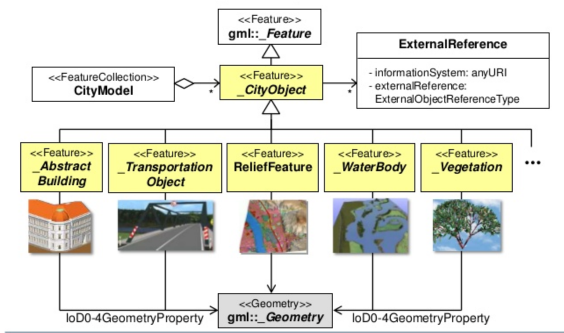


Figure 2.18: UML diagram of the top level class hierarchy of CityGML, the core class of all thematic classes is the abstract class CityObject (Kolbe, 2009)

2.3.2.2 Geometry and Topology

The geometrical feature representation of CityGML 2.0 is based on the geometrical model provided by GML 3.1.1, which is an implementation of ISO 19107 (Kolbe, 2009). The versatile interpolation options for surfaces and curves provided by GML are restricted in CityGML to planar polygons: all coordinates of the outer boundary and of the optional interior boundaries must be located in the same plane. Similarly, only straight curves are allowed. Both restrictions facilitate compatibility with many spatial databases (Oracle Spatial or PostGIS) and tools. For each position of the geometrical representation, absolute 3D coordinates must be given explicitly. Features are represented geometrically by the well-known B-Rep (Hughes et al., 2014) features such as buildings are represented by solids, which are defined by their bounding surfaces. These surfaces must be mutually non-overlapping, non-penetrating, and completely seal a volume without gaps. In general, the advantage of a solid representation of features is that the volume can be computed. In the case that a solid representation for a building is not available, alternatively a representation by surfaces that do not completely seal the feature (multi surfaces) is possible. Area or line features such as single walls or antennae are represented by surface or line geometries. Furthermore, the coordinate reference system, must be denoted explicitly. Absolute coordinates with known reference systems facilitate the integration of data. A representation using absolute coordinates is not adequate for features that have a rather complex geometry and occur multiple times , since it implies many copies of the same geometries at different locations. For such objects – and only for such objects – CityGML provides implicit geometries

Optionally, GML or ISO 19107 provide a topological representation based on the theory of cell complexes (Hatcher, 2001): each part of space is represented at most once (hence, penetrations and overlaps are avoided) and touching of objects is represented explicitly. However, GML/ISO 19107 topology is complex. As an alternative, CityGML proposes a less complex way of representing topology: By using the XML concept of links, geometrical objects (surfaces, for example) can be shared by (the boundary of) two sol- ids. This CityGML topology (often called 'backdoor-topology') is nearly as powerful as GML/ISO 19107 topology, but much simpler.

2.3.2.3 Building Module

The building model is the most important component of CityGML. It enables the representation of buildings and their component parts with regard to geometry as well as to semantics (feature types and properties). Some aspects of semantics and multiresolution are illustrated for buildings, but most of the concepts apply as well to other features in other modules.

Buildings may be represented in LoD0 by footprint or roof edge polygons. LoD1 is the well-known blocks model comprising prismatic buildings with flat roof structures. In contrast, a building in LoD2 has differentiated roof structures and thematically differentiated boundary surfaces. LoD3 denotes architectural models with detailed wall and roof structures potentially including doors and windows. LoD4 completes a LoD3 model by adding interior structures for buildings. The pivotal class of the Building Model is _AbstractBuilding, which is a subclass of the thematic class _Site and transitively of the root class _CityObject. It should be noted that _CityObject may refer to external data sets using the concept of ExternalReference. Such a reference denotes the external information system and the unique identifier of the object in this system. The concept of external references allows for any _CityObject an arbitrary number of links to corresponding objects in external information systems. (CityGML, 2012) The class _AbstractBuilding is specialized either to Building or to BuildingPart. These three classes follow the general composite design pattern: a Building may contain

BuildingParts, and as the latter class is also derived from AbstractBuilding a (recursive) aggregation hierarchy of arbitrary depth may be realized. A Building or BuildingPart is described by optional attributes inherited from AbstractBuilding - function, usage, and class, year of construction and demolition, roof type, measured height, number and individual heights of storeys above and below ground. Building and BuildingParts can be assigned multiple postal addresses.

Starting from LoD2 the boundary surfaces of Buildings and BuildingParts may be represented as semantic objects. BoundarySurface is the abstract superclass of these thematic objects having RoofSurface, WallSurface, GroundSurface, ClosureSurface etc. as subclasses. As they are also derived from class CityObject they inherit its attributes and relations. A building may have zero or more building installation objects such as chimneys, stairs, antennas, or balconies which are represented by the BuildingInstallation class. Moreover, a Building or BuildingPart may consist of rooms (represented by the class Room), with movable parts, such as chairs or tables (as instances of the class BuildingFurniture). In LoD3 and LoD4 openings in boundary surfaces may be represented by explicit thematic features of the classes Window or Door. Only in LoD4, buildings are allowed to have Rooms.

The definition of this data model restricts the way buildings can be represented in a city model. In conjunction with the normative definitions of the feature classes this has an impact on the way how concrete data would have to be registered and structured. While this may be seen as restrictive concerning the freedom of modelling on the one side, it establishes a bindingness or liability of the producer with respect to potential users or customers on the other side. Users and their applications can expect buildings to be structured in a well-defined way. This makes it profitable to develop new applications that exploit the semantic information and structure of the CityObjects.

2.3.2.4 Extensibility

CityGML gives the possibility to model objects and attributes of 3D city models which are not covered in the core data model of CityGML. Two different ways of extension are provided by CityGML 2.0. The first is the usage of generic city objects and generic attributes, both defined within the module 'generics'. The second concept for extending CityGML are the so-called Application Domain Extensions (ADE). An ADE specifies systematic extensions of the CityGML data model. These comprise the introduction of new properties to existing CityGML classes. The difference between ADEs and generic objects and attributes is, that an ADE has to be defined within an extra XML schema definition (XSD) file with its own namespace. Thereby, the extension is formally specified, which the main advantage of using ADE approach against the usage of generic objects and attributes.

In CityGML 2.0 specification two examples for ADEs are included:

- An ADE for Noise Immission Simulation (Annex H) which is employed in the simulation of environ- mental noise dispersion according to the Environmental Noise Directive of the European Commission;
- An ADE for Ubiquitous Network Robots Services (Annex I) which demonstrates the usage of CityGML for the navigation of robots in indoor environments.

Other well-known ADE examples are:

- The Energy ADE, which allows for both detailed single-building energy simulation and city-wide, bottom-up energy assessments, with particular focus on the buildings sector (Agugiaro et al., 2018).
- The LADM ADE, for describing the ownership structure of condominium units, with legal concepts from the LADM (L. Li et al., 2016).
- The Indoor ADE, which allows the management complicated indoor spatial information (Kim et al., 2014).
- The Cultural Heritage ADE, which focuses on modelling a typical Architectural Heritage feature that can be useful in planning restoration works and conservation state management (Costamagna & Spanò, 2013).
- The Immovable Property Taxation ADE, which expands the CityGML data model with the legal and administrative concepts defined in Turkish Law (Çağdaş, 2013).

2.3.2.5 CityGML version 3.0

In 5th of January 2018 an initial version of the integrated "CityGML 3.0 Conceptual Model" "CityGML 3.0 GML encoding specification" was published and are available on Github (Github, 2018). According to the authors this is a "pre-alpha" stage of CityGML version 3.0. The publication target date for the final documents of CityGML 3.0 will be:

- CityGML Conceptual Model specification 3.0 (January 2019)
- CityGML GML encoding specification 3.0 (March/April 2019)

CityGML version 3.0 brings a number of improvements, extensions, and new functionalities. The main focus of the proposed alterations is the interoperability with other relevant standards in the field. As (Kutzner & Kolbe, 2018a) mention the new model eliminates the distance with IndoorGML (OGC, 2018), IFC (ISO, 2013), Land Administration Domain Model (LADM) (ISO, 2012), as well as with Semantic Web Technologies like Resource Description Framework (RDF) (W3C, 2014). Furthermore, all modifications to the new version are carried out in a way to ensure backwards compatibility with versions 1.0 and 2.0 of CityGML. More information for the CityGML 3.0 can be found at (CityGML3.0, 2018). In the following paragraphs an overview of the most important changes and new modules and functionalities is given.

New concept of LoD (For more details see Section 2.2.2.1)

New Core model

All spatial representations are rephrased based on the two pivotal abstract classes Space and SpaceBoundary. These classes are further subdivided into a number of subclasses. All geometric representations are associated with the semantic concepts of Space and SpaceBoundary. The feature classes in the thematic modules all represent their spatial characteristics almost exclusively using Space and SpaceBoundary classes and no longer have direct associations with geometry classes. Geometry can now also be given by point clouds Point clouds can either be represented inline within a CityGML file (using MultiPoint geometry) or just reference an external file of some common type (like LAS, LAZ etc.). The new core model implements the new LoD concept.

New Construction module

This module groups all classes which are similar over different types of constructions like buildings, tunnels, bridges and introduces a new class "OtherConstruction" to represent other constructions not belonging to any of the other three modules. All construction objects now can specify multiple elevation levels with regard to different construction height points (e.g. top point, ridge point, lowest point on terrain). Also multiple measured height properties are allowed where the high and low reference has to be stated explicitly. Constructions can store a date of construction, demolition, and multiple dates of renovations. These properties were adopted from the INSPIRE BU data theme resolving the problem of the ambiguity of the measuredHeight property of Building objects in CityGML 1.0 and 2.0 and further improving the interoperability of CityGML 3.0 with INSPIRE. A new feature type AbstractConstructiveElement and its concrete building-, bridge-, and tunnel-specific subtypes will allow for mapping constructive elements from BIM datasets given in the IFC standard (e.g. the IFC classes IfcWall, IfcRoof, IfcBeam, IfcSlab, etc.) onto CityGML.

New Versioning module

Except of the attributes "creationDate" and "terminationDate" from CityGML 2.0, all objects now can have a second lifespan expressed by the attributes "validFrom" and "validTo" or more precise: the specific version of an object – with regard to its existence in the real world. "creationDate" and "terminationDate" refer to the time period over which the respective version of the object is considered an integral part of the 3D city model. Additionally, each geographic feature now has two identifiers: the "identifier" property and the "gml:id" attribute. The value of the "identifier" property is intended to be stable along the lifetime of the real-world object. The "gml:id" attribute is intended to be constructed from the "identifier" with a concatenated timestamp. Every time an object will be modified, the previous version should receive a proper "terminationDate" value and a new version of the object with the proper "creationDate" and a corresponding new "gml:id" will have to be created. This module also defines two new feature types called "Version" and "VersionTransition" allowing to explicitly define named states ("versions") of the 3D city model and denoting all the objects belonging to such states. This bitemporal model allows for queries like "How did the real world looked like at a specific point in time?" and "How did the 3D city model looked like at a specific point in time?".

New Dynamizer module

Defines concepts to represent and exchange time-varying data for city object properties as well as to integrate sensors with 3D city models. Both, simulations and sensors provide dynamic variations of some measured or simulated properties. The variations of the value are typically represented using time series data represented inline, in external files, or from sensor web services.

Revised Transportation module

Transportation objects can now be subdivided into sections. TrafficArea and AuxiliaryTrafficArea are changed to TrafficSpace and AuxiliaryTrafficSpace. TrafficSpace can have an optional ClearanceSpace. Transportation objects can now have an aerial as well as a center line representation for each LoD. In the highest LoD each lane is represented by an individual TrafficSpace object.

Features & Modules	Revised/New	References in CityGML 3.0
LoDConcept	Revised	(Löwner et al., 2016)
Core Model	Revised	
Construction Module	New	
Versioning Module	New	(Chaturvedi et al., 2017)
Dynamizer Module	New	(Chatuverdi & Kolbe, 2017)
Transportation Module	Revised	(Beil & Kolbe, 2017)

Table 2.8: New and revised features and modules in CityGML 3.0

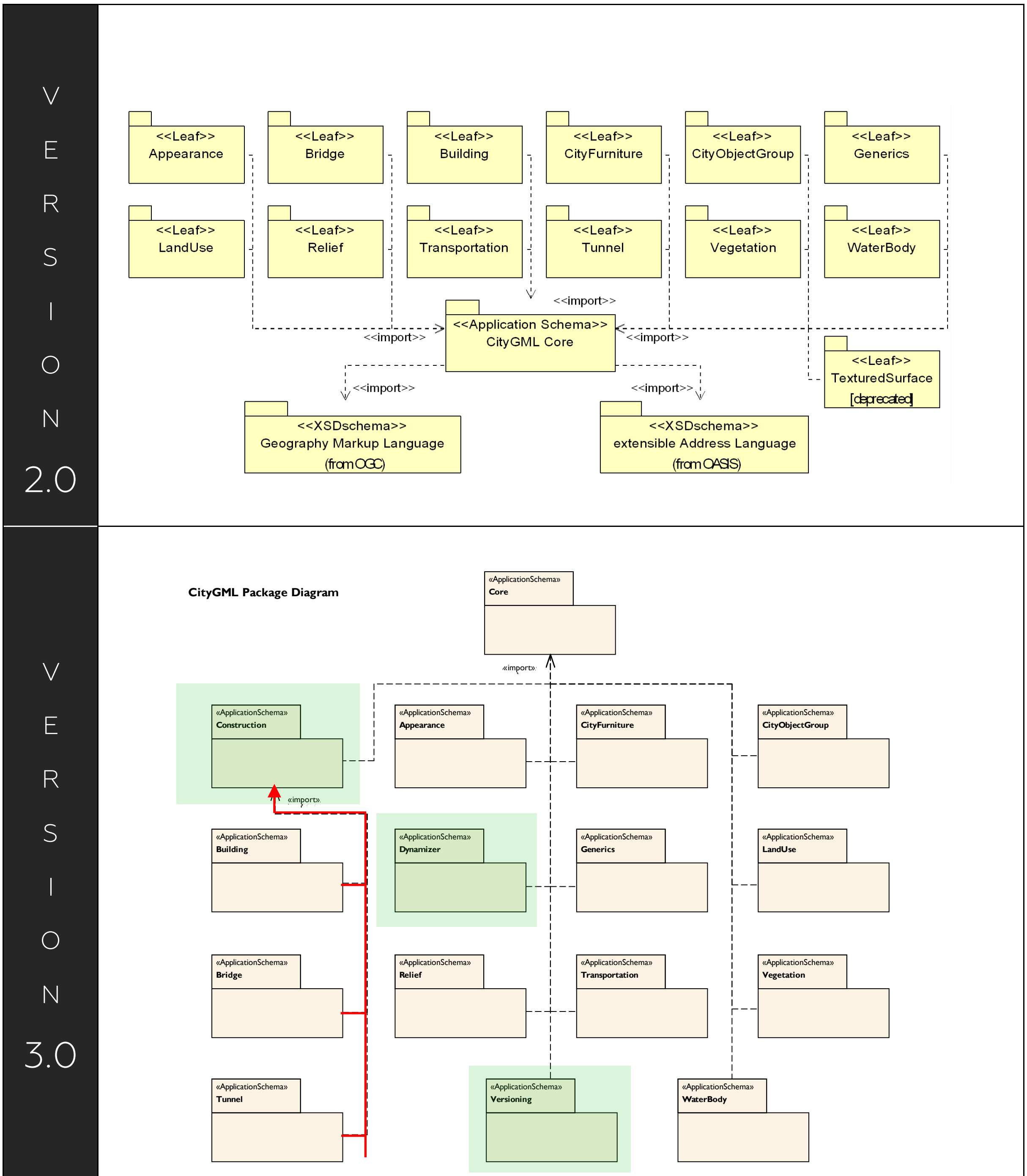


Figure 2.19: CityGML modules version 2.0 (up) and version 3.0 (down). Green rectangles show the new modules and the red line shows the new connection between the modules. (Building, Bridge and Tunnel module are under the new Construction module and not directly under the Core module (Kutzner & Kolbe, 2018b; OGC, 2012b)

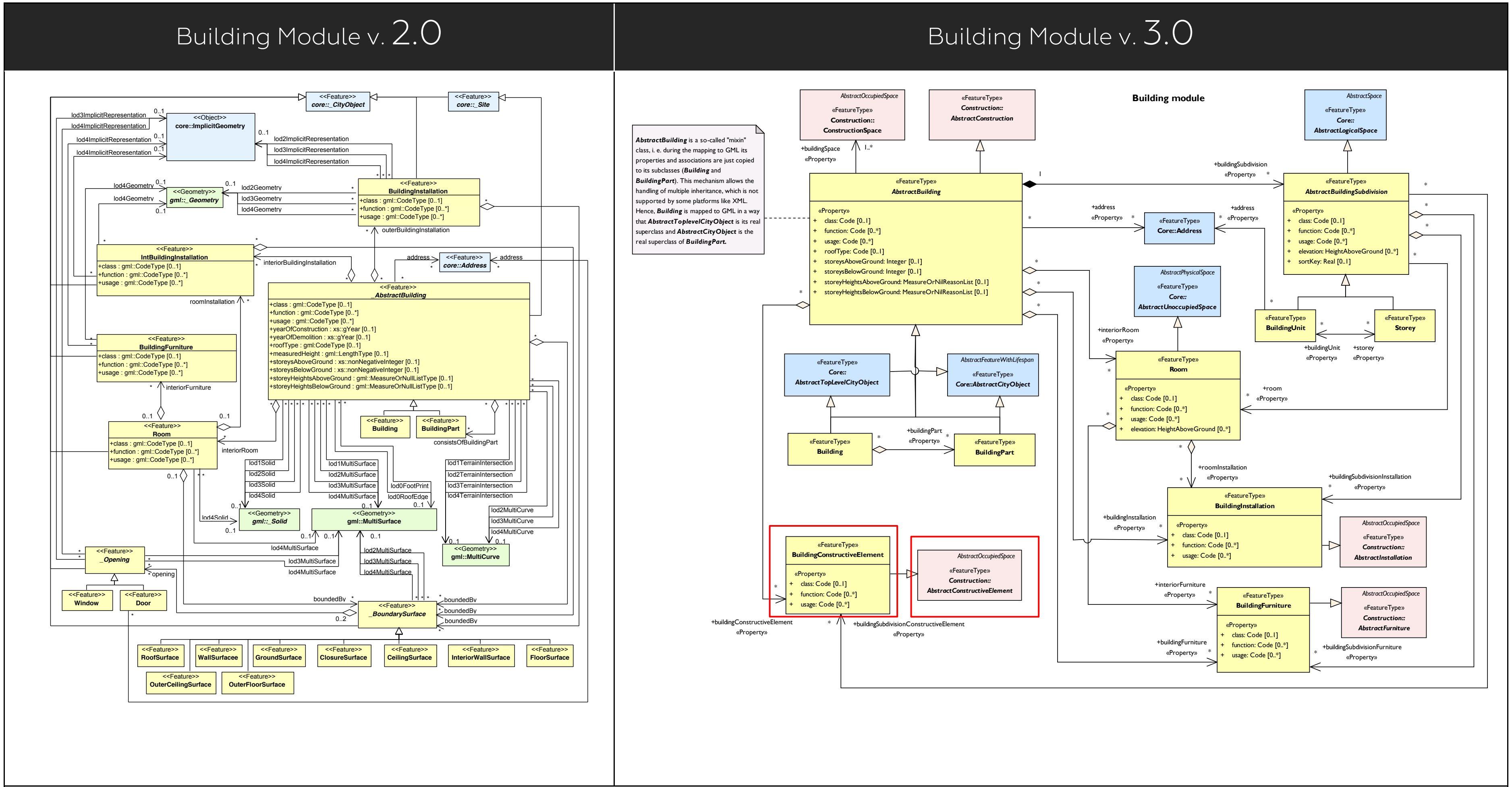


Figure 2.20: Building module in CityGML version 2.0 (left) and version 3.0 (right) (Kutzner & Kolbe, 2018b; OGC, 2012b)

2.3.3 Industry Foundation Classes (IFC)

The IFC standard [6] represent an open specification for BIM data that is exchanged and shared among the various participants in a building construction or facility management project. IFC has been designed to address all building information, over the whole building lifecycle, from feasibility and planning, through design, construction, to occupancy and building operation (Khemlani, 2004).

The history of IFC can be traced back to 1994. IFC was an output of the Industry Alliance for Interoperability, a consortium founded by Autodesk. The consortium became the International Alliance for Interoperability (IAI) in 1997 and is now known as buildingSMART–a not-for-profit organisation that describes itself as the international home of openBIM. As a data format IFC is neutral and non-proprietary. IFC was registered as an ISO standard at 2013, namely ISO16739 "Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries" (ISO, 2013). The IFC schema is regularly evolving with the current version, released in 2013, known as IFC4. (Prior releases were labelled 1.0, 1.5, 1.51 then 2x, 2x2, 2x3).

The IFC are based on ISO10303, the standard for exchange of product model data, also known as STEP. STEP includes the specification of the data modelling language EXPRESS, which is employed for defining the schema of the IFC. IFC data model structure is often represented using the EXPRESS-G notation, a graphical modelling language subset of EXPRESS, which can, however, only reach part of the expressiveness of EXPRESS. A good introduction to EXPRESS and EXPRESS-G can be found in (Schneck & Wilson, 1994).

EXPRESS adopts many object-oriented concepts, including multiple inheritances. It has the advantage of being compact and well suited to include data validation rules with the data specification. It is an ASCII file format used to exchange IFC between different applications. IFC file extensions are the following (buidingSMART, 2013):

- .ifc (IFC-SPF is a text format defined by ISO10303-21 ("STEP-File"), where each line typically consists of a single object record. This is the most widely used IFC format, having the advantage of compact size yet readable text);
- .ifcXML (is an XML format defined by ISO10303-28 ("STEP-XML"), This format is suitable for interoperability with XML tools and exchanging partial building models. Due to the large size of typical building models, this format is less common in practice.
- .ifcZIP (is a ZIP compressed format consisting of an embedded IFC-SPF file or IFC-XML file)

The total IFC4 schema holds many types of classes (130 defined types, 207 enumeration types, 60 select types, 776 entities, 47 functions, and 2 rules in IFC 4 Addendum 2). Most of them are for geometry representation, relations and topology. While these numbers indicate the complexity of IFC, they also reflect the semantic richness of building information, addressing multiple different systems, reflecting the needs of different applications, ranging from energy analysis and cost estimation to

[6] http://www.buildingsmart-tech.org/specifications/ifc-releases

material tracking and scheduling. IFC uses the American Institute of Architects (AIA) version of LoD called LoDt (Reinhardt & Bedrick, 2016) (see Section 2.2.1.2)

As a modelling standard, IFC serves as the centrepiece of three standards that make up openBIM, the de facto international BIM framework; the other two are the buildingSmart Data Dictionary (bSDD) and information delivery manuals (IDMs) (C. Clarke & Chen, 2017).

2.3.3.1 Overall Architecture of IFC Data Model

The data schema architecture of IFC defines four conceptual layers, each individual schema is assigned to exactly one conceptual layer. A system architecture perspective is diagrammed in Figure 2.21.

4) **Resource layer** — the lowest layer includes all individual schemas containing EXPRESS definitions, defining the base reusable constructs, those definitions do not include a globally unique identifier and shall not be used independently of a definition declared at a higher layer. These are generic for all types of products and are largely consistent with STEP shared library Resources, with minor extensions;
5) **Core layer** — the next layer includes the kernel schema and the core extension schemas, containing the most general entity definitions, all entities defined at the core layer, or above carry a globally unique id and optionally owner and history information. core classes may use other core classes in a limited way; however, they can reference the classes in the lower resource levels without limitation. They may not reference or use classes from the higher levels of the interoperability or domain industries.
6) **Interoperability layer** — the next layer includes schemas containing entity definitions that are specific to a general product, process or resource specialisation used across several disciplines, those definitions are typically utilized for inter-domain exchange and sharing of construction information. The classes, from this level, can reference or use other classes from the Core or the Resource layers, but not from the Domain level;
7) **Domain layer** — the highest layer includes schemas containing entity definitions that are specialisations of products, processes or resources specific to a certain discipline, those definitions are typically utilized for intra-domain exchange and sharing of information. All classes from this level, can be referenced or used from other levels or within the Domain level itself.

Because of the IFC hierarchical object subtyping structure, the objects used in exchanges are nested within a deep subentity definition tree. All physical objects, process objects, actors, and other basic constructs are abstractly represented similarly.

At the most abstract level, IFC divides all entities into rooted and non-rooted entities. Rooted entities derive from IfcRoot and have a concept of identity (having a GUID), along with attributes for name, description, and revision control. Non-rooted entities do not have identity and instances only exist if referenced from a rooted instance directly or indirectly. IfcRoot is subdivided into three abstract concepts (1. IfcObjectDefinition captures tangible object occurrences and types, 2. IfcRelationship captures relationships among objects, 3. IfcPropertyDefinition captures dynamically extensible properties about objects), which are further subdivided into classes:

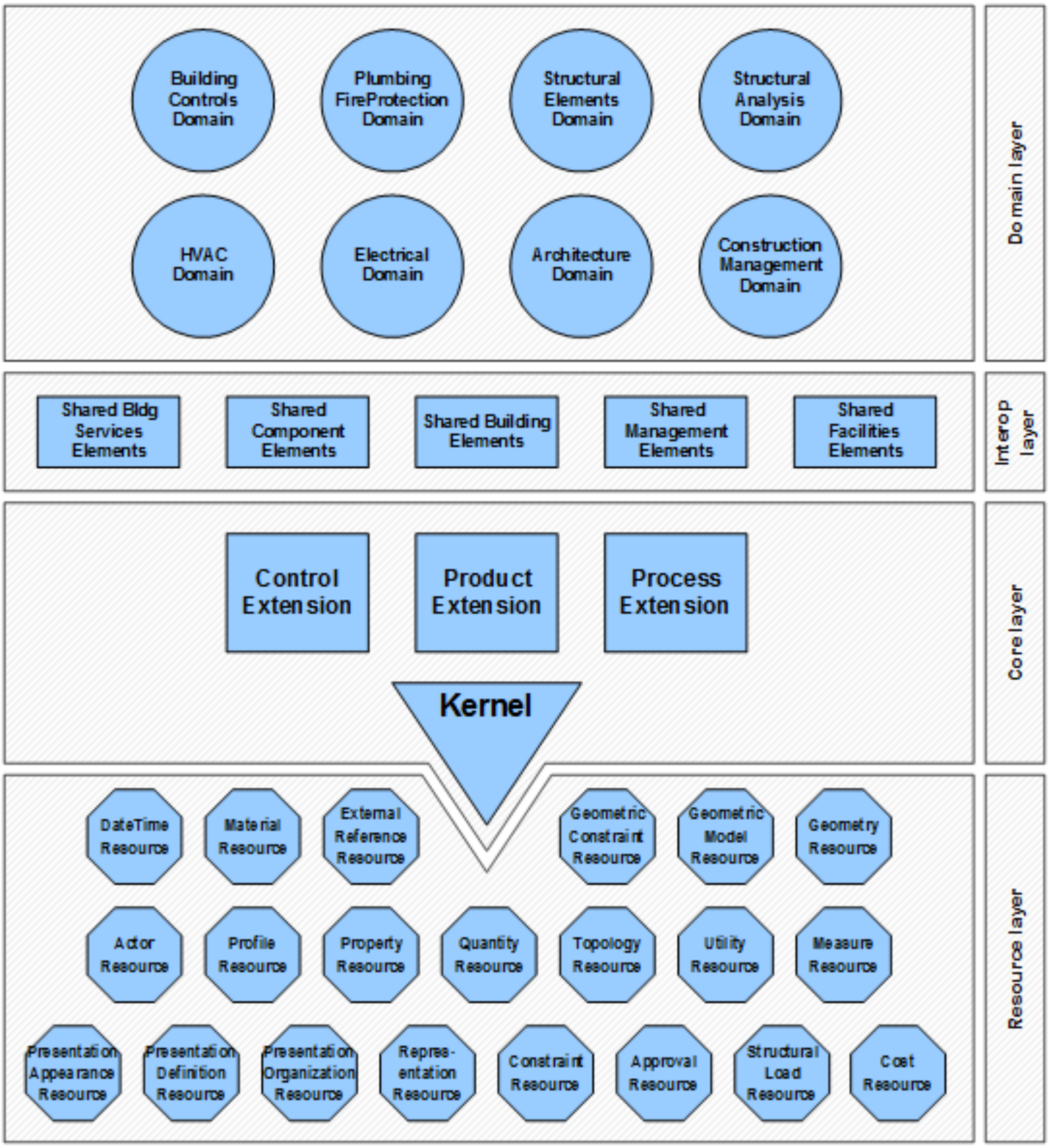


Figure 2.21: IFC4 Data schema architecture with conceptual layers (buidingSMART, 2013)

2.3.3.2 Geometry and Topology

IFC geometric aspects are mostly defined or derived from a different standard, ISO 10303 (ISO, 2014) which also specifies the STEP Physical File (SPF) encoding that is most commonly used in IFC files (.ifc). As (Arroyo Ohori et al., 2017) mention the geometries in the different types of classes can use several different representation paradigms which can be combined. In practice, most used representations are sweep volumes, B-Rep s, explicit faceted surface models and CSG (El-Mekawy et al., 2011). The representation paradigms include (Arroyo Ohori et al., 2017):

- Primitive instancing: an object is represented based on a set number of predefined parameters. IFC uses this paradigm to define various forms of 2D profiles Figure 2.22 as well as volumetric objects such as spheres, cones and pyramids.

- CSG: an object is represented as a tree of Boolean set operations (union, intersection and difference) of volumetric objects (see (Requicha, 1982) more details) for more details). Half-spaces are often used to cut out the undesired parts of surfaces or volumes.
- Sweep volumes: a solid can also be defined by a 2D profile (a circle, a rectangle or an arbitrary polygon with or without holes) and a curve (Wang & Wang, 1986) along which the surface is extruded.
- B-Rep: an object is represented by its bounding surfaces, either triangulated meshes, polygonal meshes or topological arrangements of free-form surfaces. For this kind of representation different options exist of which the most important are:

  - *IfcFacetedBRep:*A simple form of boundary representation model in which all faces are planar and all edges are straight lines. Edges and vertices are not represented explicitly in the model but are implicitly available.
  - *IfcAdvancedBRep:* A boundary representation model in which all faces, edges and vertices are explicitly represented. It is a solid with explicit topology and elementary or free-form geometry.
  - *IfcTriangulatedFaceSet:* A tessellated face set with all faces being bound by triangles.

Regarding the georeferencing parameters in the exchange process of IFC files, the IFC standard proposed the possibility to store geographic coordinates attached to the IfcSite class. The IfcSite entity describes the area where construction works are undertaken. Therefore its geographic location is important and may stand as the link between the local coordinates of a building and a global coordinate system. In that sense, the IfcSite class comes along with RefLatitude, RefLongitude and RefElevation attributes (buidingSMART, 2007). As their names suggest, the two former ones correspond to the latitude and longitude coordinates with respect to the world geodetic system WGS84, and the last corresponds to the datum elevation relative to sea level. There are also attributes dedicated to land title number (designation of the site within a regional system) and postal address of the site (LandTitleNumber and SiteAddress). The local engineering coordinate system is established by the IfcGeometricRepresentationContext entity, which contains a TrueNorth attribute that relates the axes of the coordinate system to the geodetic North.

Figure 2.22: IFC parametric curve profiles: (left) those based on the characters U, L, Z, C and T; (right) those based on trapezia, (rounded) rectangles, circles with/without holes and ellipses (Arroyo Ohori et al., 2017)

2.3.3.3 Building Module

As mentioned previously, IFC data model structure is often represented using the EXPRESS-G notation, in contrast, CityGML standard use the UML, for developing the data model. Figure 2.23 shows part of the IFC model structure in EXPRESS-G displaying the main building elements.

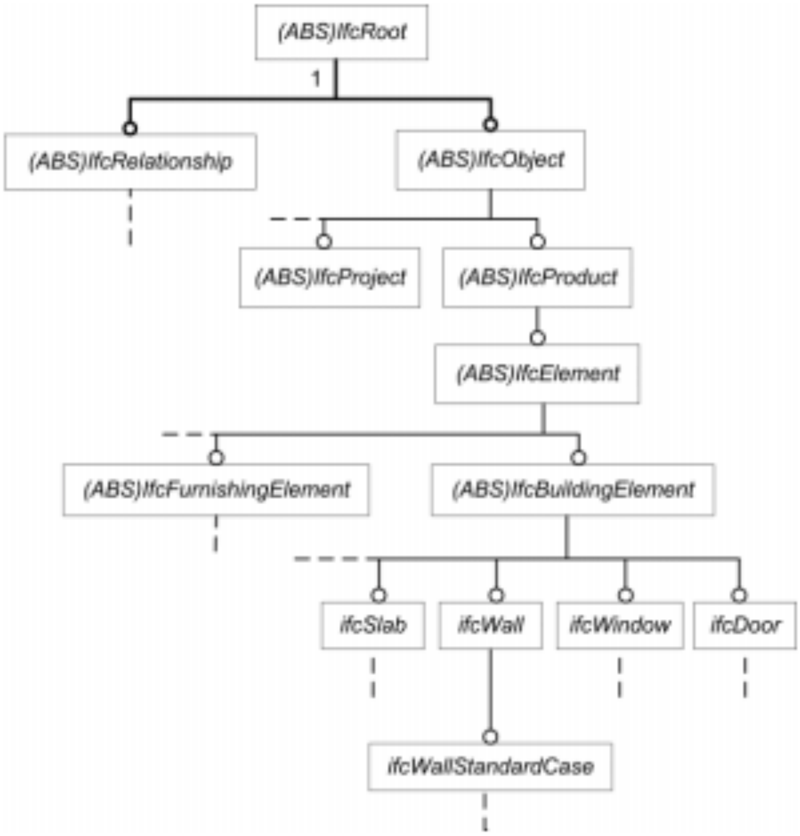


Figure 2.23: EXPRESS-G diagram showing part of the IFC model structure (Dimyadi et al., 2008)

The spatial relations between the building elements are preserved within the models' spatial structure. The information about the geographic location of the building is stored as an attribute of IfcSite object of the model. The main structure of the building is represented within *IfcSharedBuildingElements* data model. This data model defines the entities for representing the basic components of the building. Each entity in this data model is a subtype of IfcBuildingElement entity. The EXPRESS-G Representation of *IfcSharedBuildingElements* data model is shown in Figure 2.24 (Isikdag & Zlatanova, 2009b).

In IFC, a building element can have multiple geometric representations, which are differentiated by an identifier. There are pre-defined identifiers like, e.g., 'Body', 'Axis' or 'Footprint'. This concept is not comparable to the LoD concept of CityGML, because it mainly reflects modelling aspects of the corresponding AEC tools.

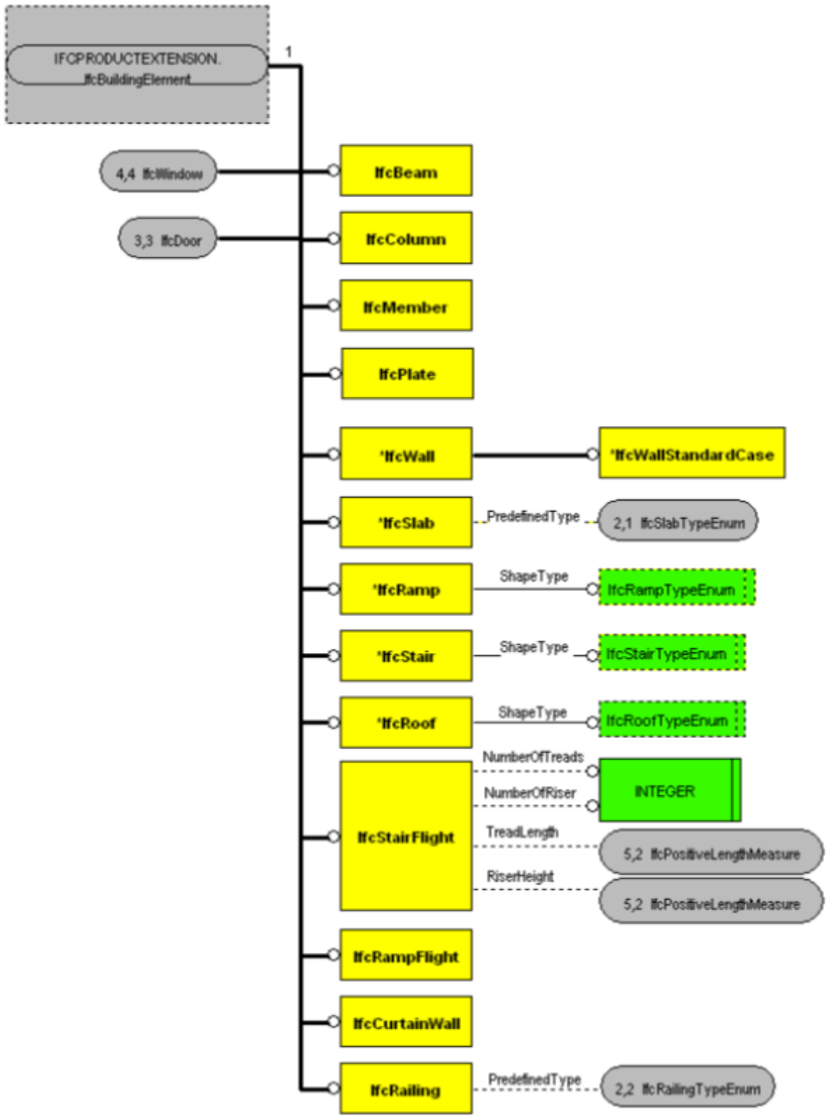


Figure 2.24: EXPRESS-G Representation of IfcSharedBuildingElements data model (buidingSMART, 2013)

Every instance of an IFC-object must belong to a spatial context. Special space-enclosing structures are the sites (IfcSite), buildings (IfcBuilding), storeys (IfcBuildingStorey) and rooms (IfcSpace). Any other spatial features, such as corridors or stair shafts are represented with the general IfcSpace definition. Additionally, any window or door placed in a wall results in an opening element (IfcOpening) that represents the cut-out in the affected wall. the concept of spaces and openings is illustrated for a simple building model. According to the specification both openings and space-enclosing objects are defined as closed polyhedrons, which can be nonconvex (Johansson & Roupé, 2009).

(El-Mekawy et al., 2011) presented an IFC building model that is primarily based on the work done by IAI and ISO in form of IFC documentation (buildingSMART7, the ISO16739 (ISO, 2013) and (Benner et al., 2005)), using UML standard notations Figure 2.25.

Based on the IFC model presented by El-Mekawy et al. (2011), a building should have at least one storey and may have multiple storeys. Each storey may have zero or more spaces related to it, i.e., a building structure which has only one wall is a building with zero spaces. Building elements and opening elements are subtypes of structural elements. Each building element has zero or more opening elements, i.e., a wall without any door or window has zero openings, whereas each opening element (like a door, a window) is attached to only one building element. Figure 2.25 shows twelve types of building elements that can represent a building structure in the IFC standard

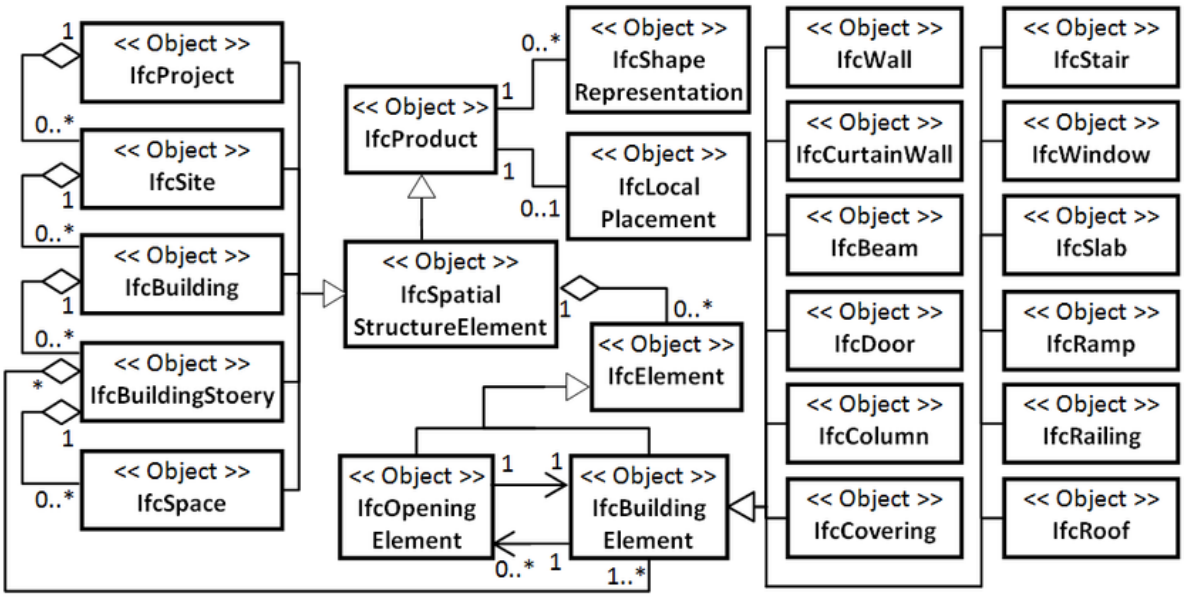


Figure 2.25: IFC building model in UML (El-Mekawy et al., 2011)

2.3.3.4 IFC5

In July 2015, the IFC Alignment 1.0 has been accepted as a buildingSMART Final Standard. IFC Alignment 1.0 is the first buildingSMART IFC for Infrastructure extension project8. The IFC-Infra overall architecture project was initiated to provide a common basis for the upcoming projects for extending IFC for the infrastructure domain, including IFC-Road, IFC-Rail, IFC-Tunnel and IFC-Bridge, which will be included in IFC5 Figure 2.26. These extensions are already included in CityGML as modules. Therefore, the new version of IFC (IFC5) will extend the thematic coverage of IFC and from a building-oriented standard, will be transformed to an infra-oriented model. This will eliminate the gap between BIM and GIS applications.

[7] http://www.buildingsmart-tech.org/
[8] http://www.buildingsmart-tech.org/infrastructure/projects/alignment

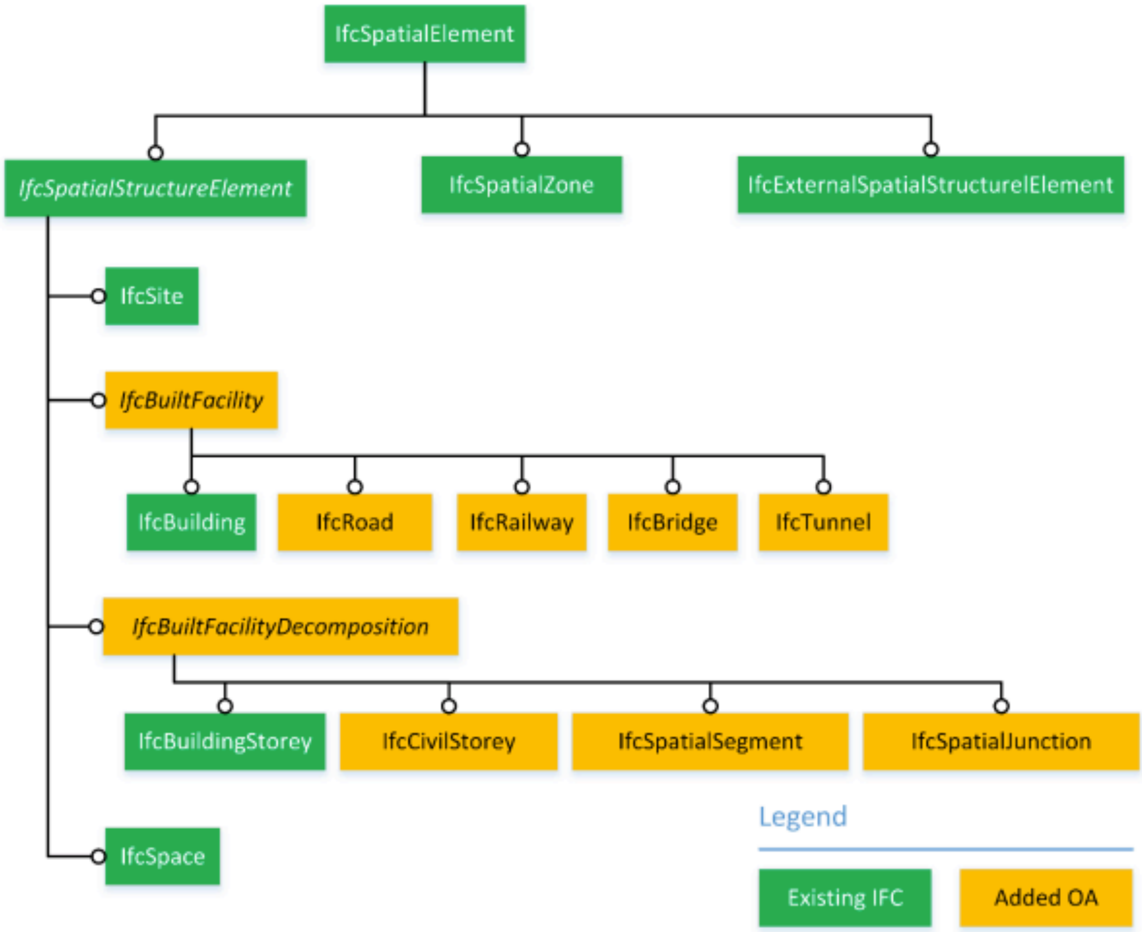


Figure 2.26: EXPRESS-G Representation of IfcSpatialElement in IFC5 (buildingSmart, 2017)

2.3.4 Comparison of 3D Formats

The comparison of standards has been performed by Zlatanova et al. (2012) on the basis of eleven criteria (Table 2.9) and by Kolbe (2009). Furthermore, Kumar and Saran (2015), presents a comparison table, using the same criteria.

The criteria they have been used in their research are the following: **Geometry**, that estimates the support of 3D geometries. **Topology** that evaluates the existence of relationships between the geometries in the model. **Texture** evaluates the support of texturing with real photos. **LoD** is an indication for support of several geometries per object. The **Objects** criterion estimates the possibility to distinguish between different objects in terms of geometry. **Semantics** indicates the possibility to assign thematic meaning to an object or a group of objects. **Attributes** estimates the possibility to incorporate attributes in the standard. The criterion **XML** indicates whether the standard is XML-based. The **Web** criterion gives an indication which standards are designed and optimized for Web use. **Geo-referencing** estimates the possibility to use geographical coordinates. Acceptance indicates the support of the standard by software vendors.

Standard/ Criterion	KML	VRML	X3D	COLLADA	CITYGML	IFC	GbXML	SHP
XML based	XML based	Text based	XML based	XML based	XML based	Text based	XML based	Text based
Geometry	Primitive Geometry	Geometry Nodes	Geometry Nodes	Mesh	B-rep	CSG	Shell	Multipatch
Semantics	No Semantics	No Semantics	No Semantics	No Semantics	Rich in Semantics	Rich in Semantics	Rich in Semantics	Basic Support
LoD	Not Supported	Basic Support	Basic Support	Not Supported	5 discrete LoDs	6 discrete LoDs	Supported	Not Supported
Texture	Basic Support	Rich Textures	Rich Textures	Rich Textures	Basic Support	Not Supported	Not Supported	Not Supported
Web Rendering	Supported	Supported	Supported	Supported	Supported	Not Supported	Not Supported	Not Supported
Georeferencing	Supported	Not Supported	Supported	Not Supported	Supported	Not Supported	Supported	Supported

Table 2.9: Comparison of 3D file formats (Kumar & Saran, 2015; Zlatanova et al., 2012)

Every 3D standard is designed for specific purpose. The main conclusions that can be extracted from the Comparison Table 2.9 are:

- VRML, X3D, COLLADA, and IFC support the largest variety of geometries;
- The most appropriate to support realistic textures are VRML, X3D and COLLADA, CityGML supports textures through the Appearance module. The use of textures in IFC is rare;
- VRML, X3D, COLLADA contain poor support for semantics and attributes;
- The most advanced standards in supporting semantics, objects and attributes are: SHP, IFC, CityGML;
- It is clear that CityGML scores relatively good on all criteria. Because of the support of semantics, objects, attributes, georeferencing and Web use, compared to SHP files (one of the most used GIS format), CityGML has almost the same power to describe real-world objects and at the same time allows better visualisation and use over the web.

3. INTEROPERABILITY OF STANDARDS & FORMATS USING CURRENT TECHNOLOGIES

Nowadays, the terms "Interoperability" and "Integration" are used frequently in BIM and GIS domains. Various efforts have been made so far to manage information towards the generation of knowledge and intelligence between BIM and GIS. The objective of Chapter 3 is to review the relevant research studies, to identify the most relevant data models used in BIM/GIS integration highlighting their advantages and disadvantages, to present the commercial software that offers the tools for the integration of these domains and finally to recognize the emerged problems at geometry and the semantic level of the procedures and software that have been used so far.

3.1 Interoperability and Integration

Every industry has a unique vocabulary and the geoinformatics is no exception. The words integration and interoperability are often used interchangeably, but semantically are quite different.

"Integration" refers to the merging or combining of two or more components or configuration items into a higher level system element and ensuring that the logical and physical interfaces are satisfied and that the integrated system satisfies its intended purpose (Kossiakoff et al., 2011). It usually involves a third party – in software terms, middleware – that translates the data and makes it "work" for the receiving system. In other words, it's not a direct path for information.

In contrast, "interoperability" reflects a more immediate form of functionality between different products. It is real-time data exchange between systems without middleware. According to the Business dictionary the definition of interoperability is the ability of a computer system to run application programs from different vendors, and to interact with other computers across local or wide-area networks regardless of their physical architecture and operating systems. Interoperability is feasible through hardware and software components that conform to open standards. In the urban domain, interoperability means the ability to use several descriptions of the same spatial area (city or urban zone) at the same time and benefit from the combined knowledge they contain.

But this is about more than semantics. Data interoperability between BIM and GIS means the ability to exchange information between the two systems. The ideal, successful data interoperability should be able to fully transfer information from BIM to GIS, or vice versa, in terms of both geometry and semantics without data loss. This must first be a reality before integration of BIM and GIS at application level can be achieved.

3D integration may be the biggest challenge in 3D domain. Current research questions under investigation are: how to integrate 3D data with different semantics; how to integrate data above and below surface; how to integrate vector and voxel data; how to integrate bathymetry data with digital terrain model; how to integrate sensor data (temperature, wind air quality) with 3D city model; how to integrate these with simulation software (Stoter et al., 2016).

3.2 Levels of Integration

The integration of BIM and GIS data is a complex topic that has been managed in different ways. In the literature several approaches for the classification of the different levels of integration of BIM and GIS has been found.

El-Mekawy et al. (2011) identifies two approaches of integration, namely: 1) Transforming IFC building models into CityGML or generating buildings in CityGML using IFC semantics and components, 2) Extending CityGML by conceptual requirements for converting CityGML to IFC models or using ADE which provide a way to represent the information that is not possible to be presented using the current CityGML classes.

Irizarry et al. (2013) identified two interrelated levels: the **fundamental** level and the **application** level. The fundamental level focuses on data exchange standards and interoperability at the data level, while the application level concentrates on the development of new methods that utilize the full potential of BIM and GIS.

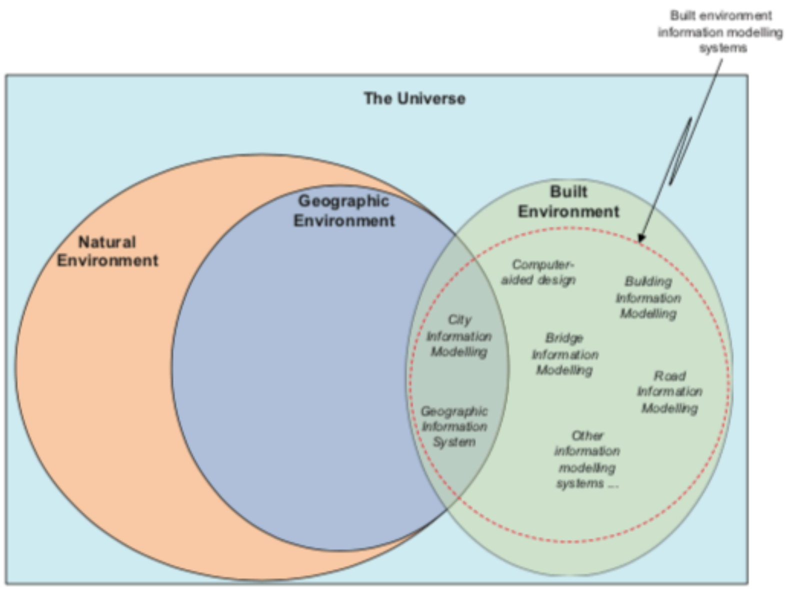


Figure 3.1: BeIM systems and the geographic environment. (Tah et al., 2017)

Kang and Hong (2015) classified them into five groups based on similar subject keywords and categorisation trends in recent research, particularly: **schema-based**, the purpose of which is to integrate heterogeneous schema models and facilitate the exchange of information among them; **service-based**, which perform services that can request and extract information from each heterogeneous model for the integration and conversion of heterogeneous data.; **ontology-based**, extract information from heterogeneous models and generate a common ontology model using a general ontology model, such as the Resource Description Framework (RDF).; **processes-based**, which offers modelling guidelines or specifications to integrate heterogeneous data into a standard model. However, the modelling for data integration is mostly carried out manually.; and finally, **system-based** approaches, which propose a systematic architecture for the integration of heterogeneous data, making full use of open libraries, components, and commercial software tools, and implement data integration architecture.

In the meantime Sam Amirebrahimi et al. (2015) classified the various attempts in three groups: at application, process, and data levels. At the **data level**, data structures are modified to meet the requirements of the other application, or existing data standards are extended. At the **process level,** both BIM and GIS are adopted in a workflow and cooperate, while the **application level** develops new applications that incorporate functionalities of both BIM and GIS, or existing applications are extended via plugins. Application level integration is the most difficult and time-consuming, as it will be built on full data interoperability, and by far, there has been no GIS software that could directly read BIM data, or vice versa. Data level integration is the most essential, and should be paid the most attention and effort. S. Amirebrahimi et al. (2016) in their research present a summary of the various integration methods (Table 3.1)

LEVEL	SUB-LEVEL	SAMPLE LITERATURE
Application	Reconfiguring or rebuilding – where an existing GIS or BIM tool is either modified by software patches or is rebuilt from scratch to support the functions (or data formats) of the other.	(Karimi & Akinci, 2010)
	Linking- facilitates data transfer between BIM and GIS software via use of API.	ArcSDE (ESRI, 2015)
Process	N/A	(Akinci et al., 2010; Karan & Irizarry, 2015; OGC, 2007; Park et al., 2014; Peachavanish et al., 2006; W. Wu et al., 2014)
Data	One-way conversion between BIM and GIS (referred to translation/conversion method)	(BIMServer, 2015; Donkers, 2013; El-dabiry & Osman, 2010; ESRI, 2006; Isikdag & Zlatanova, 2009a; Claus Nagel et al., 2009; Software, 2016; I. C. Wu & Hsieh, 2007)
	Extending standards on either BIM or GIS side to allow for storage of data from the other domain (called Extension method); or application schemas for an existing standard.	(S. Amirebrahimi et al., 2016; Cheng et al., 2013; IAI, 2005; Mignard et al., 2011; L.A.H.M. van Berlo & de Laat, 2011)
	Two-way communication between the two models using an intermediate tool or model (called Mediation)	(Benner et al., 2005; Deng et al., 2016; El-Mekawy et al., 2011; Hijazi et al., 2010)
	At DBMS level	(Bentley, 1998; Y. Li & He, 2008; Pu & Zlatanova, 2006; P. J. M. van Oosterom & Stoter, 2006)

Table 3.1: Summary of BIM-GIS integraton methods (S. Amirebrahimi et al., 2016)

Tah et al. (2017) distinguish three categories of integration: 1) **data/file conversion systems (DFC)**, 2) **semantic mapping systems (SM)** and 3) the **hybrid of both**, for which indicative references are given. They have also proposed the new term, BeIM (Building environment Information Modelling) to encompass information modelling of all systems used in the built environment to design buildings, infrastructure as well as geospatial and built facilities (Figure 3.1).

Method	Source	Area of Application
Data/File Conversion systems	(Dore & Mrphy, 2012; Saygi et al., 2013)	Historic buildings and heritage site
	(Thompson et al., 2011)	Integration of cities
	(Irizarry & Karan, 2012)	Construction site management
Semantic Mapping systems	(El-Mekawy et al., 2012)	Building elements and features
	(Irizarry et al., 2013)	Construction supply chain management
	(Cheng et al., 2013; Isikdag & Zlatanova, 2009b)	LoD
	(Akinci et al., 2010)	Semantic Web
	(Hijazi et al., 2011)	Interior Utilities
	(Borrmann, 2010)	Spatial analysis
	(Borrmann et al., 2015)	Shield tunnels
	(Karan & Irizarry, 2015)	Semantic Web on preconstruction operations
	(Döllner & Hagedorn, 2007; Karan et al., 2015)	Semantic Web
	(Benner et al., 2005)	Building elements and features
	(Clemen & Gründig, 2006)	Land survey
	(de Laat & van Berlo, 2011)	Semantic Web
	(Peachavanish et al., 2006)	Site selection and safety management
	(Stadler & Kolbe, 2007)	Spatial analysis
	(Elbeltagi & Dawood, 2011)	Project Management
Hybrid of both	(Hijazi et al., 2009)	Water utility CAD network
	(Rafiee et al., 2014)	View coverage and IFC shadow analysis

Table 3.2: Data/File Conversion systems, Semantic Mapping systems, hybrid of both applications Tah et al. (2017)

3.3 Integration approaches

There is an extensive bibliography about not only the integration methods of BIM and GIS but also the review of these integration methods. Most of the related work on this topic is concerned with converting IFC models to CityGML because that implies simplifying and removing details and additional information in the data. The inverse operation, from GIS data to BIM, is rarely discussed as it is deemed less useful and it might involve adding more details to the data. Nonetheless, there are methods to create BIM from existing models, which can be considered as GIS to BIM. An overview of this way of transformation is given by (Volk et al., 2014).

This subsection will be based on the following literature reviews of the integration approaches (de Laat & van Berlo, 2011; El-Mekawy et al., 2011; Isikdag, 2006; Jusuf, Mousseau, Godfroid, & Hui, 2017; Jusuf, Mousseau, Godfroid, & Soh, 2017; Kang & Hong, 2015; Liu et al., 2017; Ma & Ren, 2017; Song et al., 2017; Tah et al., 2017; Vilgertshofer et al., 2017; Zhu et al., 2018). Since the focus of this research is on the integration of BIM and GIS at the data level and specifically on the Conversion, Translation and Extension of Existing Standards related to the buildings, the most relevant approaches will be discussed in this section.

(L.A.H.M. van Berlo & de Laat, 2011) approached the issue of integration through the creation of a CityGML ADE for IFC building models, called the GeoBIM extension. They identified 60 to 70 IFC classes that theoretically could be transformed to a GeoBIM extension, but applied research has shown that 17 IFC classes are most likely to map to a GeoBIM extension of CityGML. The result was then presented in a XSD (Table 3.3& Figure 3.2).

IFC class	CityGML class
IfcBuilding	Building
BuildingAddress	Address
IfcWall	InteriorWallSurface or WallSurface (depending on the boundaryType)
IfcWindow	Window
IfcDoor	Door
IfcSlab	RoofSurface or FloorSurface (depending on IfcSlab-TypeEnum)
IfcRoof	Roofsurface
IfcColumn	Column
IfcFurnishingElement	BuildingFurniture
IfcFlowTerminal	FlowTerminal
IfcSpace	IfcRoom
IfcStair	Stair
IfcRailing	Railing
IfcAnnotation	Annotation
IfcBeam	Beam

Table 3.3: Mapping of IFC classes to CityGML types; including arguments and attributes. (L.A.H.M. van Berlo & de Laat, 2011)

In order to create a practical use, the GeoBIM extension was implemented in the open source Building Information Modelserver (BIMserver) (see Section 3.4.3). The issues found during the development and testing were geometric issues, excessive file size and the restriction of LoD export (LoD4 only).

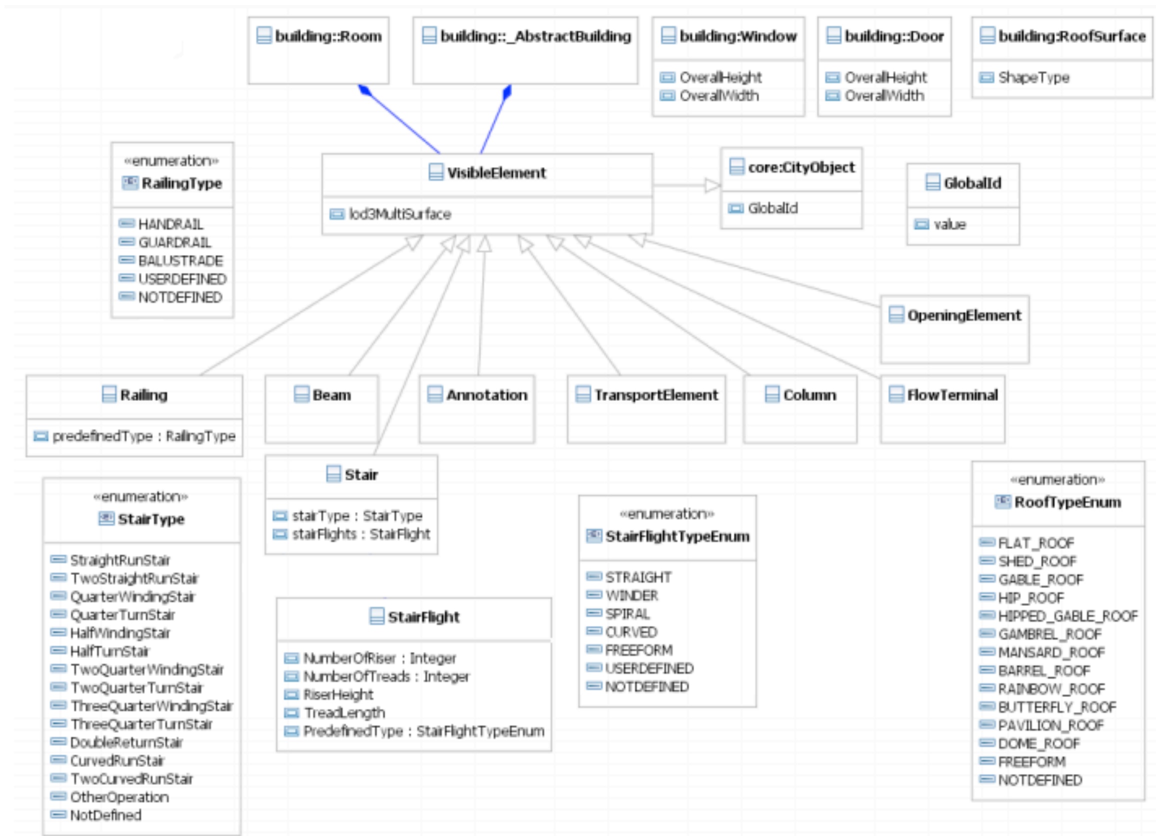


Figure 3.2: The GeoBIM extension (ADE) for CityGML represented as a UML Class diagram (L.A.H.M. van Berlo & de Laat, 2011).

(El-Mekawy et al., 2012; El-Mekawy et al., 2011) suggest the combination of information from both domains, using a unidirectional transformation method from IFC to CityGML as (L.A.H.M. van Berlo & de Laat, 2011). They create a unified model, namely Unified Building Model (UBM), in which all the semantic properties of IFC and CityGML are present, and they propose using unidirectional mappings for all the semantic classes relevant to IFC and CityGML. Their approach consists of the following steps: 1) elicitation of IFC building model, 2) development of the UBM, 3) conversion between IFC building model and UBM, and 4) conversion between UBM and CityGML building model (Figure 3.3).

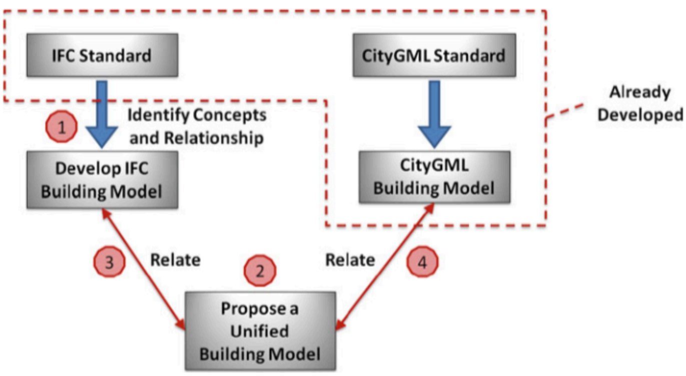


Figure 3.3: Research approach (El-Mekawy et al., 2011)

The UBM is used as an intermediate step for conversion of IFC to CityGML and vice versa. Their study is limited to the building structure, thus the objects beyond building (such as project and site) are not represented in the model. However, they extract the geometries separately from the two models and need manual editing to enrich the result. The resulting unified model does not acknowledge the particularities of the two different communities, and their focus is mostly in LoD4 since all the features are mapped.

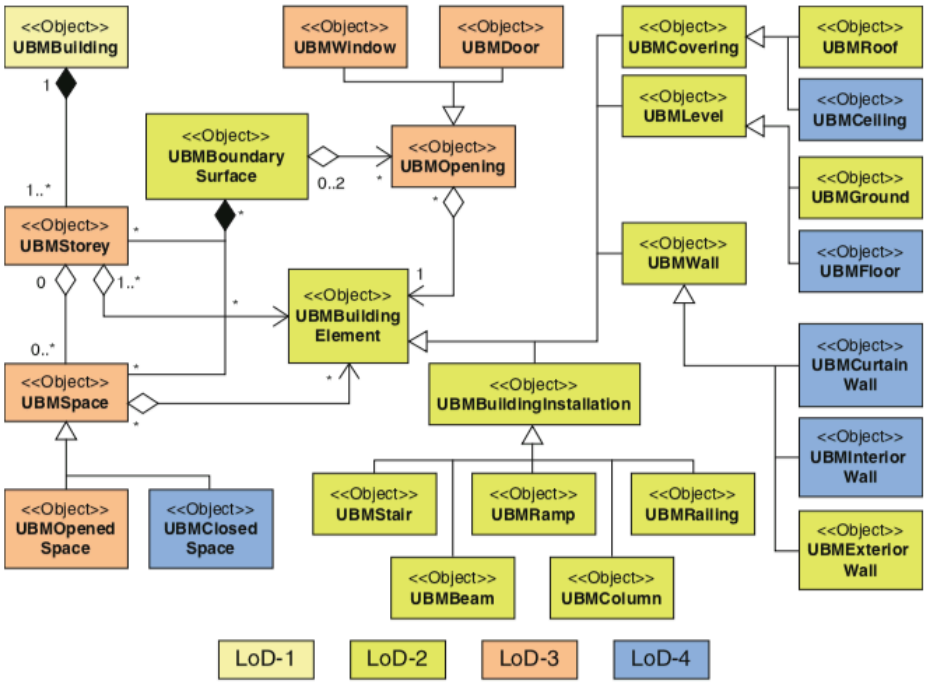


Figure 3.4: The proposed Unified Building Model (El-Mekawy et al., 2011)

Donkers et al. (2016) developed a method to automatically generate CityGML LoD3 building models from IFC from already existing architectural models stored in the IFC format. They present a conversion algorithm that applies the correct semantics from IFC models and that constructs valid CityGML LoD3 buildings by performing a series of geometric operations in 3D. Their algorithm contains three main steps: (1) the filtering and the mappings of the semantics; (2) the geometric transformations needed to extract the exterior envelope of a building; (3) further geometrical refinements to ensure validity of the output model. The conversion algorithm implemented in C++. They have validated the effectiveness of the conversion algorithms with several real-world datasets.

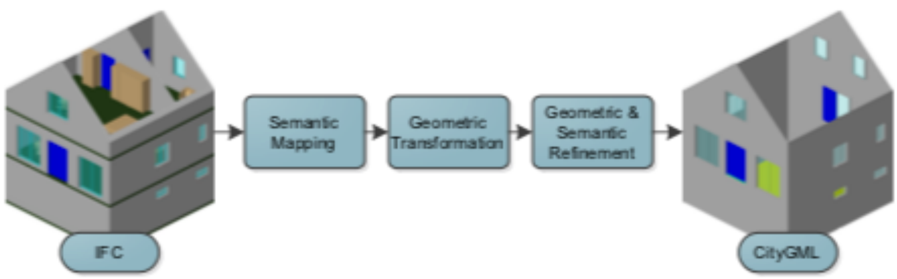


Figure 3.5: General workflow diagram of our algorithm (Donkers et al., 2016)

Geiger et al. (2015) simplified the complexity of the IFC model in terms of both geometry and semantics. They proposed a process (Figure 3.6) for BIM building models to extract different generalized representations for buildings and building elements.

The goal of generalisation is to reduce the geometric and semantic complexity of a building model without losing relevant information.

Figure 3.6: Schematic overview of the generalisation process (Geiger et al., 2015)

As a first step, an intermediate data model, which is called ExtrusionBaseModel, is generated (Figure 3.7) Based on this model, LoD1 and LoD2 representations can be derived. LoD3 is generated consecutively on the basis of the LoD2 model (Figure 3.8), LoD4 is not included in their research.

Figure 3.7: Semantic filtering of IFC data (left original model; right only IFCSLAB, IFCWALL, IFCBEAM) (Geiger et al., 2015)

Specifically, based on the ExtrusionBaseModel, different strategies are implemented for the LoD1 generalisation process. The first strategy calculates one extrusion for the whole building. The second variant focuses on walls and in the third case, footprints of horizontal slabs are used to create the extrusion geometries. Compared to LoD1, the generation process for LoD2 is completely different. The basis is also the ExtrusionBaseModel but the generated outer contour of the building is much more detailed with a correct roof shape and a detailed classification of the outer boundary surfaces. An algorithm was developed to generate clipping planes for the extrusion containers of ExtrusionBaseModel, based on the original geometry of the IFC building elements IfcRoof and IfcSlab with PredefinedType set to ROOF. Supported geometry types in this process are extrusion and B-Rep. In the second step, building elements are used for a semantic classification of the clipped extrusion geometries.

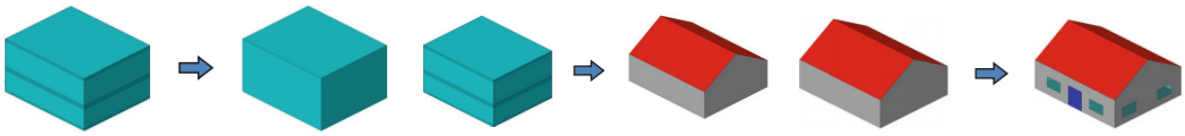


Figure 3.8: Transformation ExtrusionBaseModel into a LoD1 model (left), Transformation ExtrusionBaseModel into a LoD2 model (middle), Transformation of a LoD2 into a LoD3 model (right) (Geiger et al., 2015)

The transformation process from a LoD2 model to a LoD3 model mainly differs in applying voids for doors, windows, and openings and creating the appropriate elements. In this step, the ExtrusionBaseModel is no longer used. This process is realized by interpreting the relations of IFC. The methodology described above is implemented as an early prototype on IFCExplorer (Figure 3.9) (see Section 3.4.1).

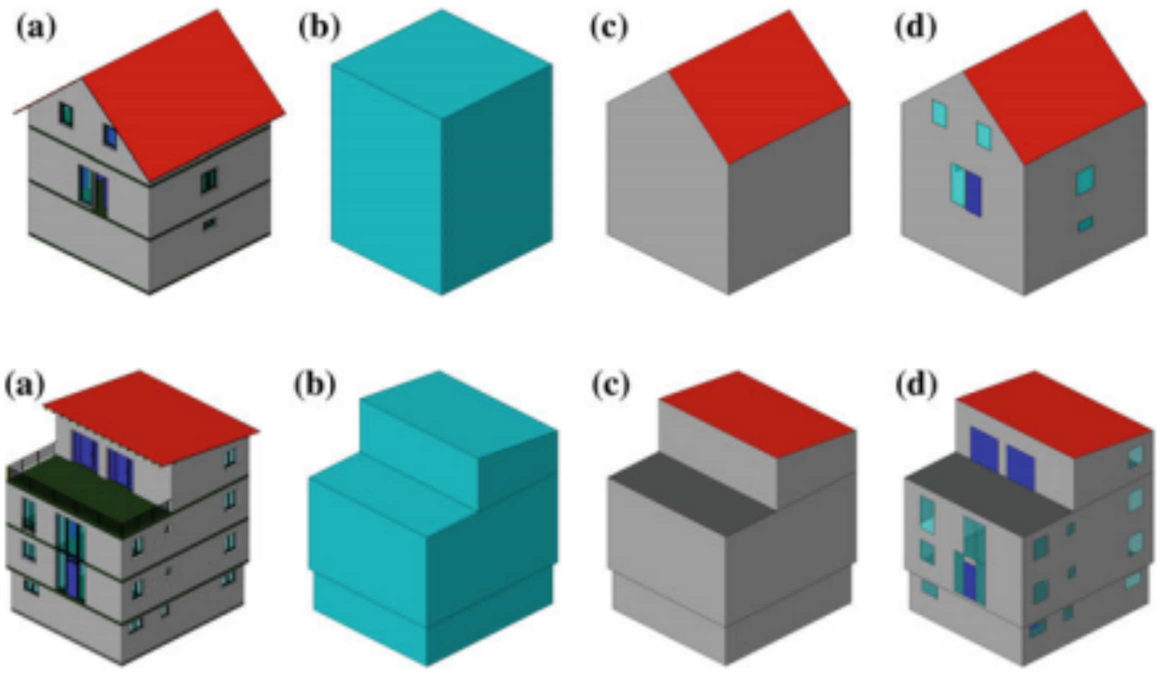


Figure 3.9: Examples of the conversion using IfcExplorer for two different buildings. (a) IFC Model, (b) CityGML LoD1, (c) CityGML LoD2, (d) CityGML LoD3 (Geiger et al., 2015).

Another semi-automatic way of BIM and GIS the data conversion/translation is through the Extract Transform Load (ETL) process, which extracts the homogeneous data from the source systems and load the data into data warehouse by transforming the data into proper format or structure. A step-by-step description on how to integrate BIM data into a spatial information model by ETL was demonstrated by Rafiee et al. (2014). Kang and Hong (2015) extended this study by providing a more detailed and structured framework using similar principles. Normally, the process starts with the geometry conversion, which is followed by the Global ID allocation. Global ID will be used to automate the semantic translation (Rafiee et al., 2014). The semantic information in the original data is mapping into a structure that destination data format can recognize. In Section 3.4 the most popular and successful commercial platforms applying ETL are presented.

(S. Amirebrahimi et al., 2016) developed a data model to assess the damage of building due to flooding. They designed a conceptual data model, using a UML class diagrams. The data model consists of seven packages inheriting its high-level feature definitions from the GML. These packages are namely: the Core (CoreUrbanFlood), Terrain, Flood, Building, Utility, Valuation and MaterialDomain. Since we focus on the buildings, the building package will be furthered analyzed.

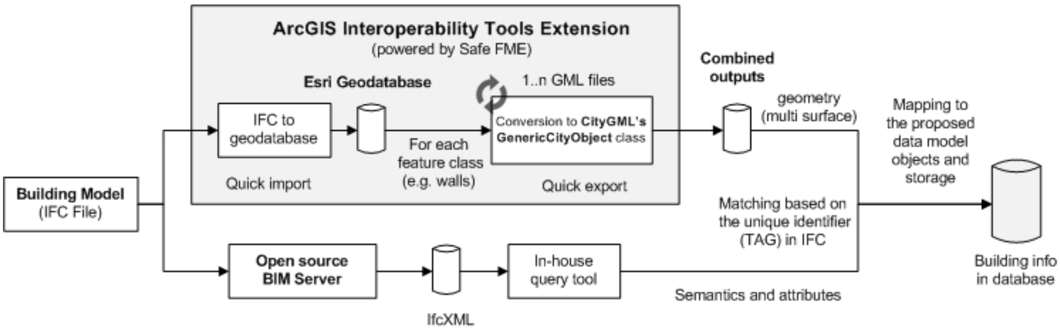


Figure 3.10: Converting the building information from IFC to the database based on the proposed data model (S. Amirebrahimi et al., 2016)

The building package (see Figure 3.11) comprises the classes that represent the building and individual or an aggregation of building components. The "Site" class represents a parcel characterised by an address and a 2D polygon which can contain one or more buildings. Each building has a value associated with it (BuildingValue) and can be represented by either its 2D footprint or the aggregation of 3D geometries of its components (e.g. stories). A building consists of at least one storey defined by "BuildingStorey" class and may contain utility objects or systems, as well as any subtype of the abstract class "BuildingElement". Each of utility or building elements has a damage state and replacement cost attributes associated with them that can be used for cost analysis. The building components defined in the model include slabs, structural beams and columns, walls, roof, stairs, framing members, windows, doors, airbricks or any type of void opening. In addition to these components, "BuildingElementPart" defines a class for a generic part of any other element. Explicit classes for wall parts ("WallComponentElement" such as the cladding) or covering parts (e.g. ceiling insulation and lining) using "CoveringLayerElement" are defined in the model to represent these objects. In this model, windows and doors are defined either as single object or a combination of a lining (its frame) and a minimum of one panel that may have their own geometry, material and cost. "Space" class in this model defines those elements for representing the internal (e.g. room) or external (e.g. the backyard) spaces for the building.

They tested the data model on a case study, using a house model exported to an IFC file (see Figure 3.10). For extracting geometry, they converted the IFC elements to ESRI geodatabase using the ArcGIS Interoperability Extension. The attributes and relationships between elements were obtained from an exported XML version of the IFC file and combined them with their geometry using a unique identifier of IFC elements. Their method for integration of BIM and GIS models at data level could facilitate comprehensive assessment and 3D visualisation of building damages costs.

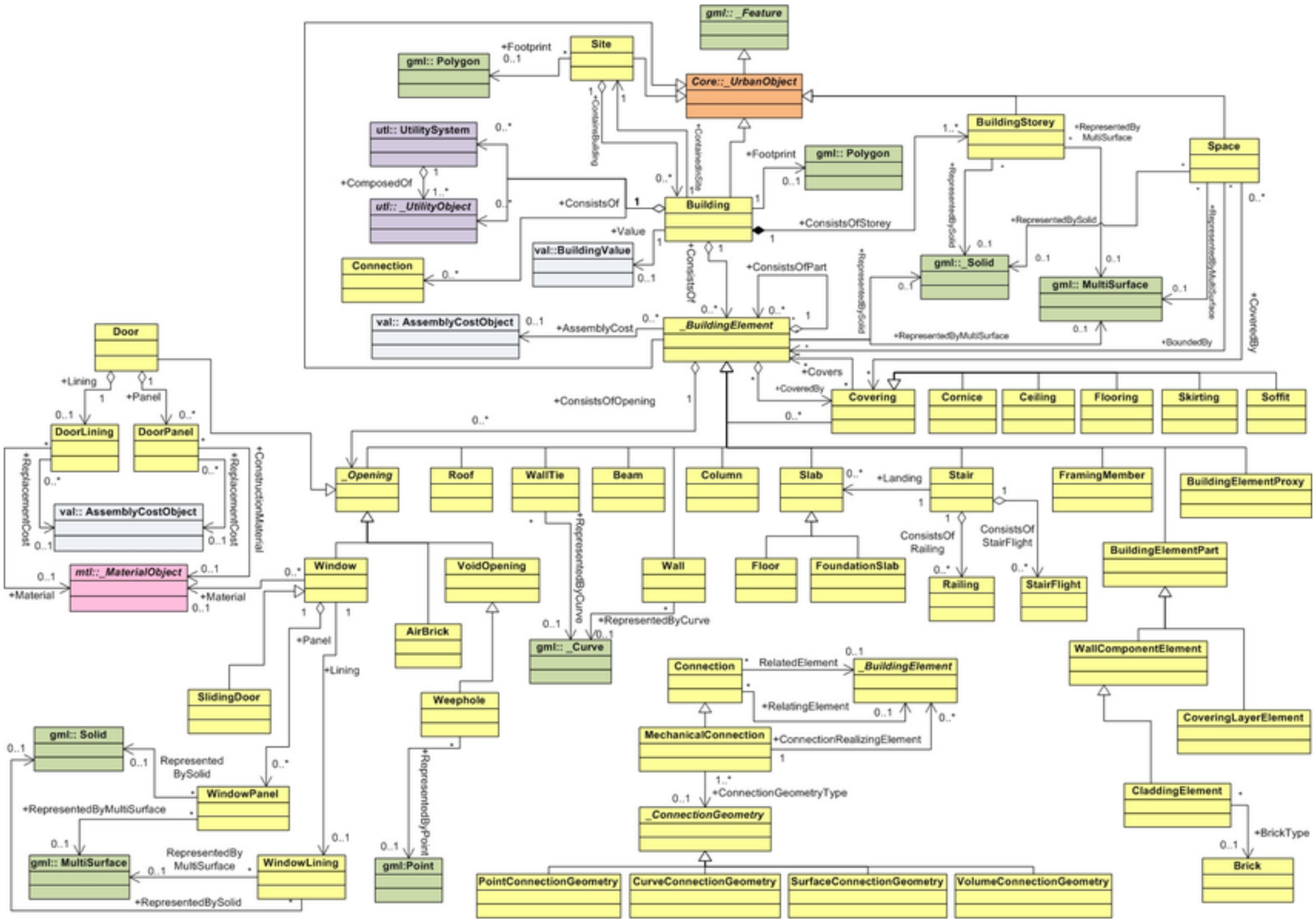


Figure 3.11: UML Model of the Building Package (S. Amirebrahimi et al., 2016)

3.4 Software for Integration

A few programs offer the possibility to convert IFC models into CityGML models. The following programs implement the ETL process, in order to convert/translate the data from one format to another.

3.4.1 Safe FME[9]

The spatial ETL used by FME is not a one-way conversion tool. It supports a bidirectional reading and writing between IFC and CityGML (Donkers, 2013). There is support for both reading IFC as well as writing CityGML, however the required data model transformations need to be built in the Workbench. The conversion is thus not a native function of FME. FME connects data in various systems, including CAD, BIM, GIS, rasters, and point clouds. FME provides the opportunity to manipulate the structure of the data, by controlling the layers and attributes, to change the projection, and to manipulate the content by applying a set of premade transformations.

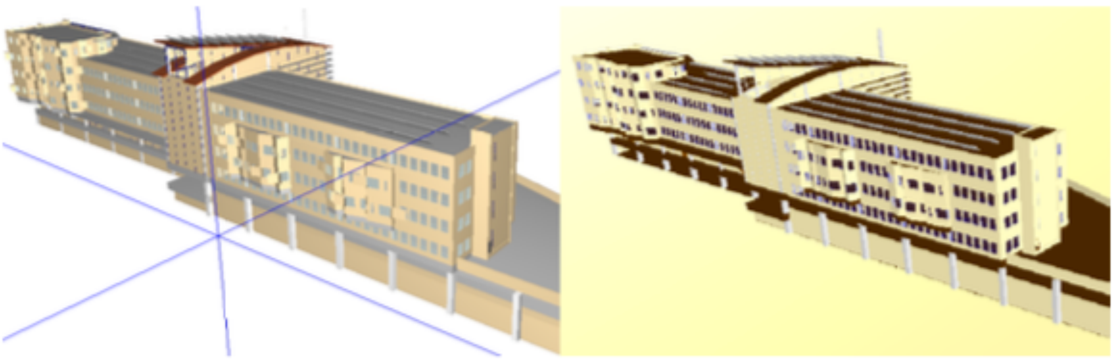


Figure 3.12: Conversion results by Safe Software FME, IFC on the left, CityGML on the right (Donkers, 2013)

3.4.2 IfcExplorer

The IfcExplorer is prototypic software for integration, analysis, 3D visualisation and conversion of spatially referenced data, that has been developed at Karlsruhe Institute of Technology (KIT) (Benner et al., 2010). Originally designed for exploring the semantic building model IFC, the software now supports different GML-based GIS data formats (CityGML with different ADEs, XPlanGML, rudimentary GeoSciML, gbXML, XPlanGML), DXF ad Map Services. Their goal is not to transform IFC to CityGML, but to have all models natively in one environment. Their software is however capable of converting between IFC and CityGML. The converter can create all LODs to some extent, but there are no solids defined and the geometry does not yet adhere to the CityGML specification. LOD1 and LOD2 can also be converted to IFC, which results in models that are in compliance with the IFC specification, but does not comply to Coordination

[9] https://www.safe.com/

View 2.0, which is an initiative to define the correct implementation of IFC. Development by KIT continues on LOD1 and LOD2. On request, the conversion results are not shown since the results are not proper CityGML, yet which could lead to misinterpretations.

3.4.3 BIMServer

The framework of BIMserver is to enable the storage, maintenance, query and centralisation of information from different data sources and encourage collaboration among participants (Beetz et al., 2010). It therefore is also capable of exporting the IFC files to COBie, CityGML, Collada, KMZ and SceneJS. It includes an Eclipse Modelling Framework (EMF) model[10] of IFC, a Berkeley DB database[11] and communication interface using web technologies, such as Representational State Transfer (REST), Simple Object Access Protocol (SOAP) and web user interface (de Laat & van Berlo, 2011) (Figure 3.13). The IFC Engine DLL library[12] and the CityGML4j java library[13] are connected to the EMF interface and compose CityGML files (de Laat & van Berlo, 2011).

However, BIMserver only allows unidirectional translation from IFC to CityGML.

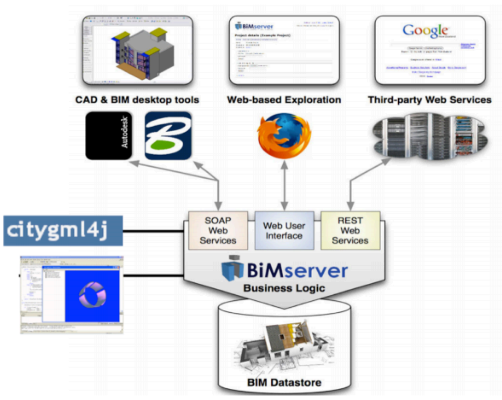


Figure 3.13: The schematic representation of the open source BIMserver software architecture (L.A.H.M. van Berlo & de Laat, 2011).

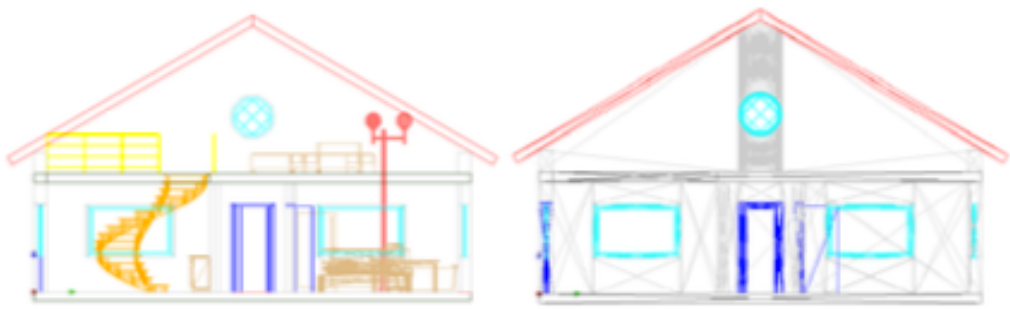


Figure 3.14: Conversion results by BIMserver, IFC on the left, CityGML on the right (Donkers, 2013)

[10] http://www.eclipse.org/modeling/emf/
[11] http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/overview/index.html
[12] http://www.ifcbrowser.com/
[13] http://www.citygmlwiki.org/index.php/CityGML-Developers

3.4.4 ArcGIS Data Interoperability & Oracle Spatial with Spatial ETL

Two other large companies adopting spatial ETL process to integrate heterogeneous data sources are ESRI and Oracle, and the functions are called ArcGIS Data Interoperability (ESRI, 2017) and Oracle Spatial with Spatial ETL (Dale, 2009), respectively.

ArcGIS Data Interoperability extension, created by ESRI comes with a visual diagramming interface for authoring and executing data connections and transformations. ArcGIS Data Interoperability provides state-of-the-art direct data access and data translation tools in addition to the ability to build complex spatial ETL tools for data validation, migration, and distribution. It supports a number of industry standards, such as CSV, CAD, JSON, XML, and RSS, OGC formats such as GML and CityGML, WFS, and KML/KMZ, brings CAD-created BIM data in Industry Foundation Classes IFC or Revit RVZ files into ArcGIS

3.4.5 Comparison of converters

The mapping process during the ETL can be flexible, and it enables both full and customized translation between BIM and GIS data. One the other hand, the model mapping process is costly in terms of both time and money, although this mapping work is not as costly as those at deeper standard level. Therefore, ETL tools are good for bulk data conversion, which translates large volume data in batch. Errors are unavoidable sometimes during the model mapping and interpretation stage, as human processing involves and the irreconcilable difference between the two standards. For example, it is not easy to fully interpret the contiguous space boundaries in IFC by ETL process, which cannot be easily solved by manual conversion either (Boyes et al., 2015). Moreover, another disadvantage of ETL is that it cannot implement on-demand and real-time data conversion. Although many previous studies have been done on this topic, quick response ETL is still not achieved at practical level.

(Donkers, 2013) has presented a comparison table (Table 3.4) for the aforementioned converters, but he evaluated only the results of the conversion from IFC to CityGML in LOD3.

Theoretically, both geometric and semantic information should stay relatively consistent, when translated between IFC and CityGML by ETL. However, this depends on the data knowledge of the operator on both GIS and BIM, especially for sematic information conversion. In addition, there is normally no coordination information stored in IFC, and an additional step (for example, LocalCoordinateSystemSetter in FME) is required to set the original location of the model in IFC.

	FME	BIMserver	IFCExplorer
Correct Explicit IFC Geometry	No	Yes	Yes
Transformation of Geometry	No	No	No
Correct Semantics	No	No	No
ISO Validity of Geometries	Equal to IFC	Equal to IFC	Equal to IFC

Table 3.4: Evaluation of current IFC to CityGML LOD3 converters (Donkers, 2013)

3.5 Problems of Integration

The main focus of this research is on the IFC and CityGML standards. The two models have many similarities but also have many differences. Both IFC and CityGML are semantic models and maintain some relations between the items. The relations are not topological (although CityGML can incorporate a topological structure). Being created for two different application domains, the class definitions differ significantly. A comprehensive overview of the differences between IFC and CityGML is presented in Table 3.5.

Criteria		IFC	CityGML
Geometry	Implicit	CSG, Sweep, NURBS, parametric	None
	Explicit	BRep, Surfaces, Polylines, Points	BRep, Surfaces, Polylines, Points
Georeferencing		Cartesian World Coordinate System, Local/Relative Coordinate System	Projected Systems
Semantic	Buildings	Complex, Highly detailed	Limited to very basic classes
	Environment	Site, Landscape, IFCInfra on the (distant) horizon	Exhaustive (city furniture, water bodies, tunnels, bridges etc.)
	Relations	Aggregation, Decomposition, specialisation on different levels	simple
Extendability	On-the fly	Property-Sets, external classification and Libraries (IFD) can be assigned to generic representations (IfcProxy)	Properties can be added; some code lists exist (classification without properties(?))
	Schema	Monolithic	XML schema's can be added via namespaces (ADE)

Table 3.5: Mismatches between IFC and CityGML

3.5.1 Geometry Level

Advantages and disadvantages of the various integration approaches, regarding the geometry level, are described in detail by many researchers, in relation to the specific integration approach they adopt (Isikdag & Zlatanova, 2009a). In this chapter a summary of the problems addressed in the various procedures is presented. In broad sense, we can identify three major problems, which are mainly related to the core differences of these standards: 1) Reference System, 2) 3D geometry, 3) LOD.

GIS and BIM adopt different approaches to represent the geometry as discussed earlier. IFC could use one of, or combination of, CSG, sweep volume, and B-Rep to represent a 3D geometry, while CityGML usually only uses B-Rep. The various natures of those different 3D geometry storing mechanisms set a barrier against easy data transformation. From the perspective of IFC to CityGML, the main obstacle is the transformation of B-Rep in one system to B-Rep in another, sweep volume to B-Rep, and CSG to B-Rep. While the B-Rep to B-Rep issue could be solved by the coordinate system transformation function, the sweep solid to B-Rep could be achieved by a customized function (Deng et al., 2016), and CSG to B-Rep could be completed by open source computational geometry library VTK; By far, the methods for transferring B-Rep to other shapes have been developed, while the transformation of other shapes to B-Rep have not, and it is the most essential step to finish geometry transformation from CityGML.

The spatial relationships between two models may differ as well. BIM adopts a local placement system in which objects are defined in local planar coordinate system (3D Cartesian Coordinate System). The local placement system of an object is referenced to that of another object. For example, in IFC, a door is given with its relative coordinates with respect to the wall it belongs to. This mechanism facilitates model modification, for instance, if the location of wall is to be changed, only the placement system of the wall is to be modified, and the locations of windows or doors attached to it will change automatically without the need to modify them individually. On the contrary, GIS generally uses a geographic coordinate system (GCS); each object inside it has absolute coordinates in the form of latitude, longitude, and altitude. CityGML would maintain a relation 'belong to' between door and a wall but the coordinates of the door will be absolute.

Apart from transformation between shape forms and the reference system, the LOD harmonisation is also an important aspect. LODs in GIS and LODs in BIM reflect the amount of detail contained in a city or building model. Both of them have five levels; however, they have different definitions for corresponding levels, as discussed in Sections (2.2.1.2 & 2.2.2.1). For example, CityGML defines a building model in LOD0 as the footprint or roof edge of the building, while an element defined as LOD100 in IFC may not even be a geometric representation. As a result, those levels cannot be simply mapped, which poses a huge barrier to the complete data interoperability between BIM and GIS. Progress is being made in this concern. In (de Laat & van Berlo, 2011) research, IFC models could be exported to CityGML LOD4, while (Donkers et al., 2016) developed a method to automatically generate CityGML LOD3 building models from IFC files for the construction of a city model and (Deng et al., 2016) successfully transformed IFC buildings to CityGML LOD1-LOD4 models. Reviewing the efforts have been carried out so far, the main focus is on the building models, while the research in the other infrastructures is very limited. Another limit is those studies could only achieve the transformation of IFC to CityGML Data exchange in the opposite direction is also important if complete data interoperability is to be achieved. The development of CityGML 3.0 and the restructure of the LODs will eliminate the distance between BIM and GIS world.

Geometry transformation between BIM and GIS could also be partly completed by the commercial software packages discussed in Section 3.4. All of them allow the users to convert IFC models to CityGML at different LODs. The users can in some cases choose which IFC objects should be used. However, all the features are converted, without any selection or post-processing to keep only the relevant ones. Different projects use such approach to obtain integrated datasets useful for visualisation-based analysis (some of them have described in detail in Section 3.3). However, none of these tools could fully transfer geometry and semantics between BIM and GIS (Donkers, 2013). By definition IFC has more classes than CityGML. For example, at least 7 classes, such as beam, column, and stair, could be matched to “BuildingInstallation” in CityGML. The mapping of “BuildingInstallation” to the corresponding class in IFC could be even more difficult, as one has to first decide to which class of IFC the “BuildingInstallation” is to be mapped.

Integration at this level is usually for 3D visualisation purposes, and an obvious disadvantage by far is the semantic information loss, which is mainly due to semantic mismatch between the two domains.

3.5.2 Semantic Level

The main problem at the semantic level it is the different classes between IFC and CityGML. One defines a component while the other does not, for instance, IFC defines beam, column, stair, and so on, while CityGML includes all these elements in “BuildingInstallation” class. The semantic loss often happens on the GIS side, as IFC contains much more information than CityGML.

Mainly two approaches are being used, to eliminate the semantic loss of data: 1) Developing new data models that are usually ontology-based using semantic web technology. 2) Modifying existing schema including schema simplification and extension; These two strategies have one thing in common that they both rely on existing schemas (IFC/CityGML).

In terms of the development of a new or an intermediate data model, the implementation is usually carried out with the semantic web technology, which is a set of technologies used to represent, publicate and browse structural data on the web (Hor et al., 2016). An ontology typically consists of a finite list of terms and the relationships between them, used to describe a domain of discourse, through a hierarch structure (Antoniou & van Harmelen, 2008).(Costa et al., 2016; Deng et al., 2016; El-Mekawy et al., 2011; Hor et al., 2016; Karan & Irizarry, 2015) used this approach in an attempt to extend BIM’s or GIS’s scope. The first developed a District Data Model (DDM), which combines information of both standards, to support the retrofitting design of energy-efficient districts, (El-Mekawy et al., 2011) developed the UBM, which was tested with BIMServer, while (Hor et al., 2016) established the Integrated Geospatial Information Model (IGIM).

The semantic web technology is implemented with the construction of ontologies for both fields, the ontology mapping between the two data models is following, which outputs an extended ontology containing all classes and properties and finally the combined elements are translated into semantic web standards, after which a query language, is being used to retrieve the information needed from the model.

This strategy is promising; however, it is often time-consuming to use these techniques, as they are still developing. Moreover, there are very few widely accepted ontologies for the AEC domain, and different projects independently develop their own ontologies, which impairs effective information exchange within this field (Karan & Irizarry, 2015).

In schema extension approach, usually the CityGML is to be extended, through ADE (de Laat & van Berlo, 2011). Sometimes, the IFC is also to be extended, for instance (Borrmann et al., 2015) extended the IFC model for incorporating multi-scale representation of shield tunnels, which was later transformed into CityGML. The semantic part of the model implements the multi-scale approach by providing explicit LOD objects and making use of the space aggregation hierarchy for modeling refinement relationships. The geometric part associates the individual semantic objects with a procedural description which is defined across multiple LODs. The resulting product model provides geometric-semantic coherence and at the same time mechanisms for automated consistency preservation. It thus responds to the particular demands of multi-scale representations in the context of highly dynamic planning processes. However, this strategy may encounter problems in terms of visualisation.

At the semantic level, more progress has been made, as in some projects, such as GeoBIM and Semantic City Model, bidirectional information exchange has been achieved; however, the methods proposed in these studies tend to be project-specific, which means that it could not be directly applied to another study, unless the necessary adaptations have not taken place. Therefore coordinated efforts are need to focus on the development of a more generic method for unidirectional information exchange.

4. METHODOLOGY & RESULTS

A practical approach carried out at the area of NTUA is described in this Chapter. The objective is to propose framework to better organise and group elements during the design process for the effective integration of IFC, CityGML and SketchUp. The evaluation and the results of the proposed methodology are also described in this Chapter.

4.1 Case Study

The scope of this thesis is to define the optimum way of data structuring, in order to produce semantically enriched models according to IFC and CityGML standards, by using commercial software. The study area was the National Technical University of Athens and more specifically the Veis building of the School of Rural and Surveying Engineering. The location was selected due to the availability of detailed architectural plans from the Department of Technical Services of the NTUA. The 3D model of this building was also created using different modelling techniques, during the Research Program «3D Visualisation of the National Technical University of Athens Zografos Campus - 3D Campus NTUA project». In this project, the modelling of the building was carried out using ESRI City Engine platform. The 3D building model was initially implemented using ESRI City Engine platform and then was automatically transformed into a CityGML model, by using the tools of 3DCIM CityGML toolbox (Floros et al., 2016; Pispidikis et al., 2018; Tsiliakou, 2013).

4.2 Software and Tools

Currently various modelling software facilitate the development of semantically enriched 3D City and Building Models (Section 2.2.4). The selection of tools is depending on the demands of every project. Part of this research work, is the evaluation of the selected tools.

For the preparation of 2D architectural plans, AutoCAD was used as the designing software. AutoCAD is a commercial software application for 2D and 3D CAD drafting. It is used across a wide range of industries, by architects, project managers, engineers and graphic designers. As Autodesk's flagship product, by March 1986 AutoCAD had become the most ubiquitous CAD program worldwide. In addition to its broad spectrum of users, AutoCAD is also very easy and straightforward to use and supports a wide range of file-based exchange formats. One of those is Trimble SketchUp. Utilising 2D AutoCAD drawings as a starting point for creating objects' geometry in SketchUp is a fast way to generate a detailed 3D model.

Buildings were modelled using the Trimble SketchUp Pro 2017, providing both 2D drawing generation from a model and interfaces to other applications through various file formats. It can read as background DXF, DWG, and IGES geometry input. It can also import IFC geometry—for some types. SketchUp Pro also supports export to 3DS, AutoCAD DWG, AutoCAD DXF, FBX, OBJ, XSI, and VRML. Some of these can be read into BIM platforms and the geometry recreated from the imported background. The development of models with semantic information is possible, through the use of different plugins.

The classification of entities according to IFC was carried out with the IFCManager[14]. SketchUp has developed a Classifier tool, which is embedded in the platform and gives the possibility to import, not only IFC classifications but also custom-made classifications. Any schema can be loaded into the software and then applied to any component. However, by utilising this tool it is concluded that it has many drawbacks mostly regarding the attributes and the export mechanism. Therefore, IFCManager was used. IFCManager was created around the Dutch "BIM basis ILS" initiative that tries to achieve a basic Information delivery specification for the Dutch building industry. Through this plug-in, the user can classify SketchUp components as IFC entities, with NL-SfB codes and edit SketchUp properties for name, material and layer.

The transformation modelling to CityGML was performed by CityEditor plugin, which has been thoroughly examined for the purpose of this thesis, highlighting its shortcomings and potential. Finally, the visualisation of the semantic data models, according to IFC and CityGML was held with the FZKViewer. The FZKViewer is a software tool for the visualisation of semantic data models in BIM and GIS domains. The graphical representation, properties and relationships between objects can be displayed textually. In addition, various data evaluations are possible. Thus, tables of the objects along with their properties and attributes can be created and partially displayed in a color-coded graphic representation.

4.3 Methodological Approach

The first step of the process focuses on the reorganisation of information contained in floor plans, taking advantages of the layering and blocking functions supported by CAD application. The original geometry and graphical representation in the raw floor plans is preserved as much as possible. The building was modelled in SketchUp environment, and the generated 3D models only correspond to 3D geometric representations. Using the grouping or component mechanism these shapes are transformed into objects, to which the user can assign semantic information. The most demanding task of this process was the appropriate grouping of elements, in terms of transforming them into meaningful information objects, not just representations. Using the IFCManager, the classification of different entities was performed according to the IFC documentation. Finally, the extraction of the model, according to CItyGML format, in different LoDs was achieved, using the CityEditor tool.

The workflow of the process is presented in Figure 4.1.

[14] http://www.bimloket.nl/BIMbasisILS

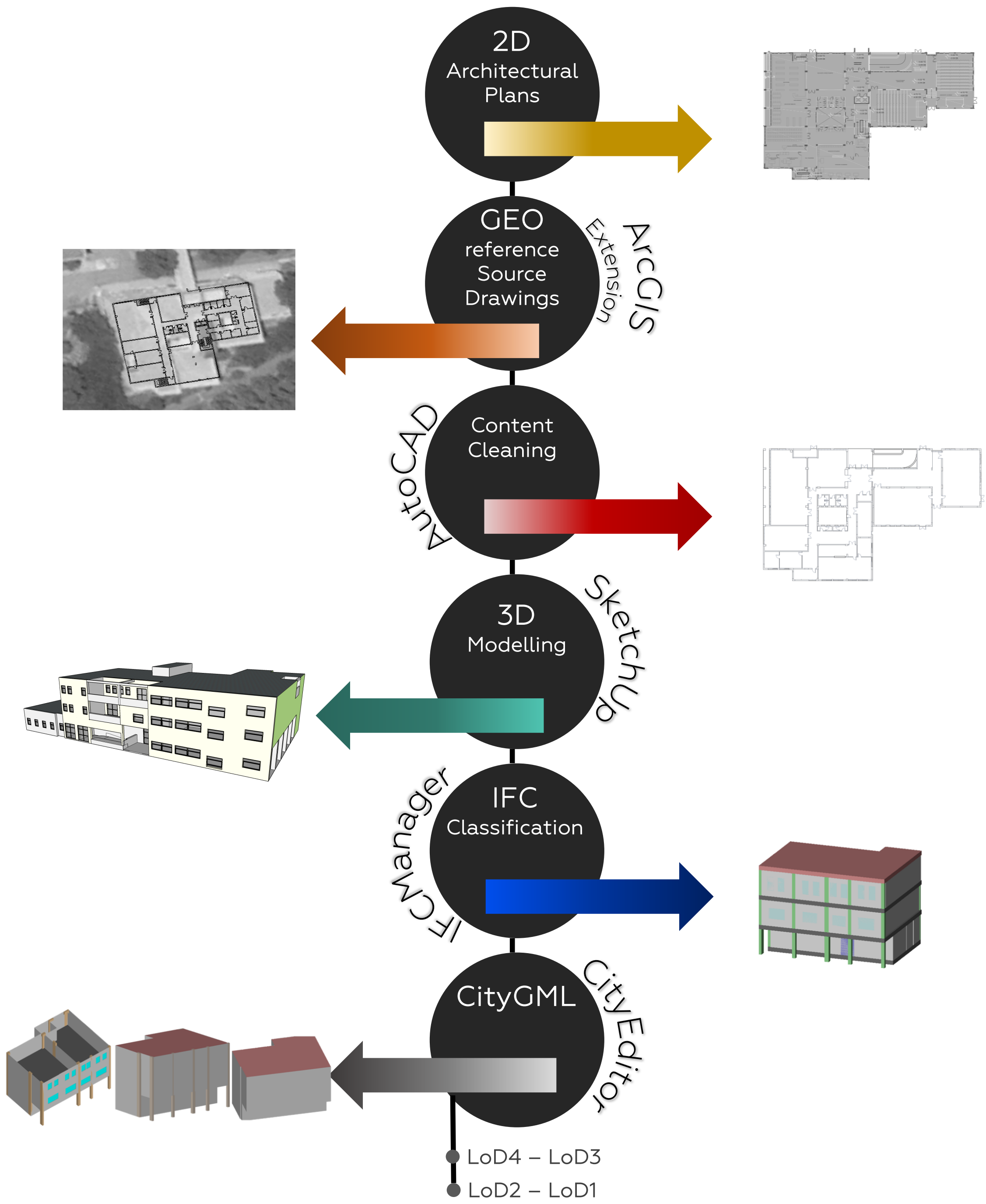


Figure 4.1: Methodological approach of the case study

4.4 Preparation of Input Data

SketchUp and AutoCAD can be used collaboratively to generate detailed site models. Utilising 2D AutoCAD drawings as a starting point for creating objects geometry in SketchUp is a fast way to generate a detailed 3D model.

The process followed for the preparation of floor plans in AutoCAD, is presented in Figure 4.2.

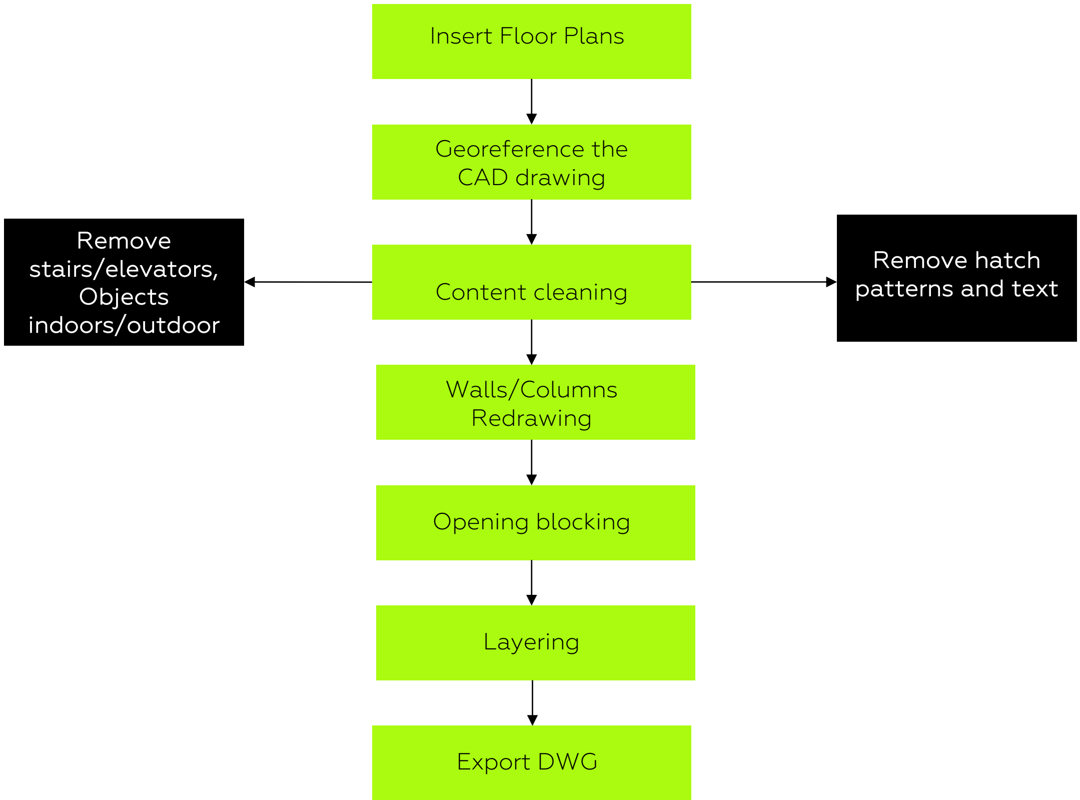


Figure 4.2: Workflow of floor plan proceessing in Autocad

The **first step** of the process was the georeference of the floorplans using the ArcGIS extension for Autocad [15]. ArcGIS for AutoCAD is a free plug-in that provides interoperability between AutoCAD and ArcGIS. Within the AutoCAD environment, there is a new tab, from which the user has access to GIS maps, map services, image services, and feature services hosted by ArcGIS Server. The way SketchUp handles the georeference is through the Geo-location tool, which adds a basemap for the selected area. If the model is created directly on SketchUp, with no input data from other sources, the SketchUp model will be georeferenced and it will be possible for the model to be exchanged with other programs. Since the modelling of the building will be based on an DWG file, the georeference in SketchUp environment through the Geo-location tool will cause a big lack in precision. Therefore, the input data need to be already georeferenced with the appropriate software and then imported into SketchUp. The Geo-location tool will be used only for the terrain model.

Additionally, it is necessary to identify the information in AutoCAD that are needed for SketchUp modelling, having in mind firstly which of the elements SketchUp supports and secondly, the geometries that will be used for the production of the semantically enriched models according to CityGML and IFC standards. The supported CAD elements are the following: arcs, circles, entities with thickness, faces, 3D faces, layers, lines, polylines, nested blocks, AutoCAD regions, points, ellipse, spline, while the unsupported CAD elements are the proprietary ADT or ARX objects, dimensions, hatching, text and XREFs. If an element isn't supported, SketchUp simply ignores it

[15] https://www.esri.com/en-us/arcgis/products/arcgis-for-autocad

when the user imports the CAD file. In many cases the input floor plans should be redrawn in detail. The elements that will be used in the case study, will be the structural objects, (walls and columns), the content of the structural objects (openings – walls and windows), objects indoors outdoors (e.g. balconies and stairs). However, only the structural objects, (walls and columns) and the content of the structural objects (openings – walls and windows) will be reconstructed in AutoCAD.

Walls and columns are represented by closed polygons, which can be drawn by LINE, POLYLINE and LWPOLYLINE, the three most commonly used line entities in CAD files, or any combination of these three entities. POLYLINE is a connected sequence of segments created as a single entity, which can be 2D or 3D, and has been supported since very early version of AutoCAD; LWPOLYLINE is simply "lightweight" version of a POLYLINE, which is always 2D and supported in later versions.

The data types for opening geometry: LINE, POLYLINE, LWPOLYLINE and ARC. In addition to LINE, POLYLINE and LWPOLYLINE that have been introduced above, ARC entity is also used in opening geometry. An ARC is a portion of a circular arc, which can be created in AutoCAD in many ways. For sliding doors, pocket doors and bi-fold doors, only the arrows and annotating contents in the symbols should be removed. For casement windows and combined windows within which a casement window is included, they should be redrawn in same way as doors. For other windows, besides removing annotating contents, no extra redrawing is required, since the bounding boxes of these windows correctly indicate their locations.

An important step of this procedure is the check of the measurement units (inches, feet or a metric unit), in order to match the SketchUp model's units to the CAD file's units and thus maintain the scale and dimensions of the imported CAD geometry.

Thirdly, the AutoCAD linework will be reorganized on appropriate layers. Other linework will be isolated and discarded. The objects that will be completely discarded from the drawing will be the indicative information and symbols (e.g. texts, dimensions, auxiliary lines) and hatch patterns within walls. Interior and exterior walls, columns, window blocks, door blocks, stairs, rails should be separately stored in seven different layers, with the appropriate names. The initial floor plans were organised in 21 different layers, the final file from AutoCAD will contain only seven layers. When the AutoCAD layers are imported in SketchUp, the information on those layers will directly be transferred. Therefore, the basic organisation of layers is accomplished in AutoCAD prior to importing a model into SketchUp. The AutoCAD linework that composes the Flatwork Base is imported into SketchUp first. The imported linework is then used to generate faces, surfaces and solids. As with layers, AutoCAD blocks import directly into SketchUp. Once they are imported, AutoCAD blocks instantly become SketchUp components. This means that all versions of the block are now components; they can be edited to affect all the other similar blocks/components in the model.

Figure 4.3, Figure 4.4, Figure 4.7, Figure 4.8 show the initial floor plans provided by the Department of Technical Services of the NTUA. Figure 4.5, Figure 4.6, Figure 4.9, Figure 4.10 present the final product after the implementation of the procedure presented in Figure 4.2.

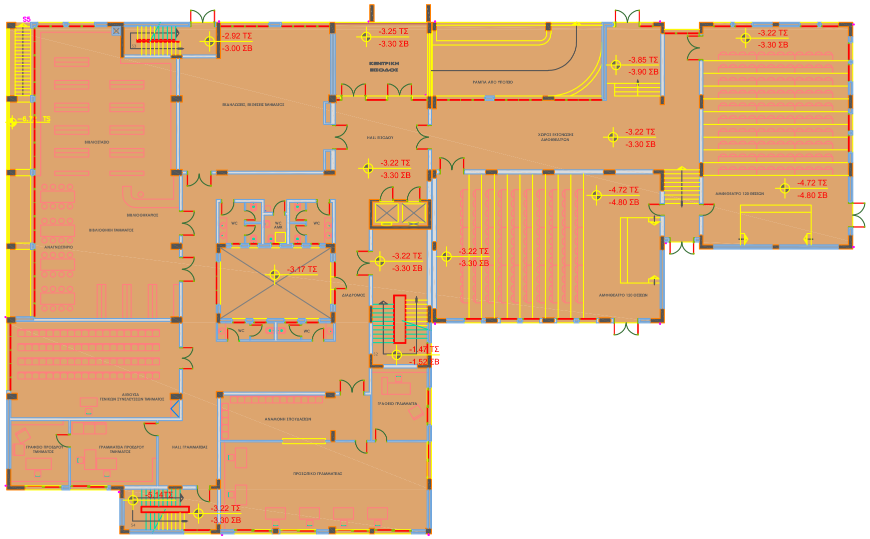


Figure 4.3: Initial architectural plan of ground floor

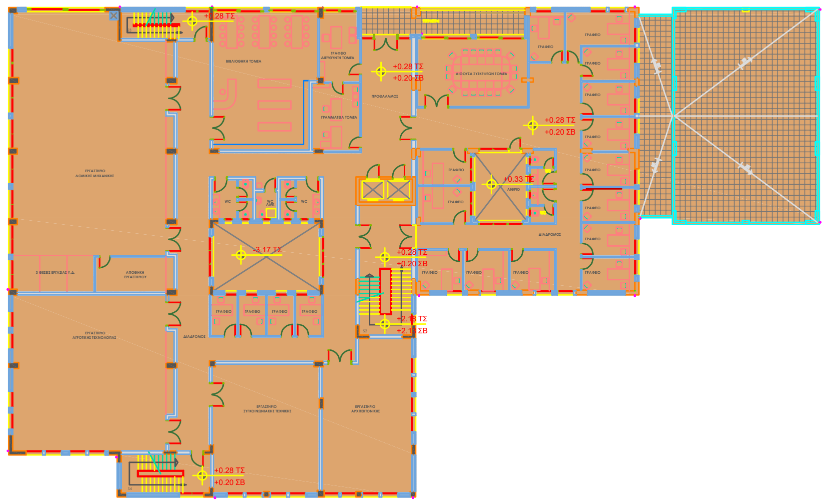


Figure 4.4: Initial architectural plan of first floor

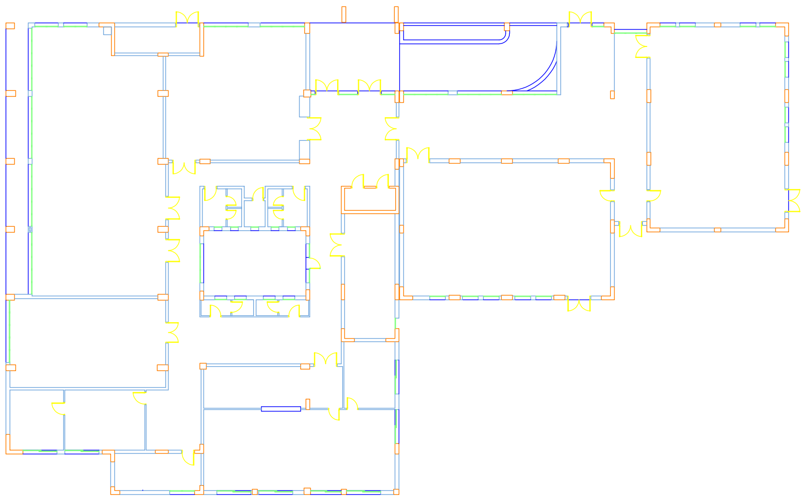


Figure 4.5: Architectural plan of ground floor after processing

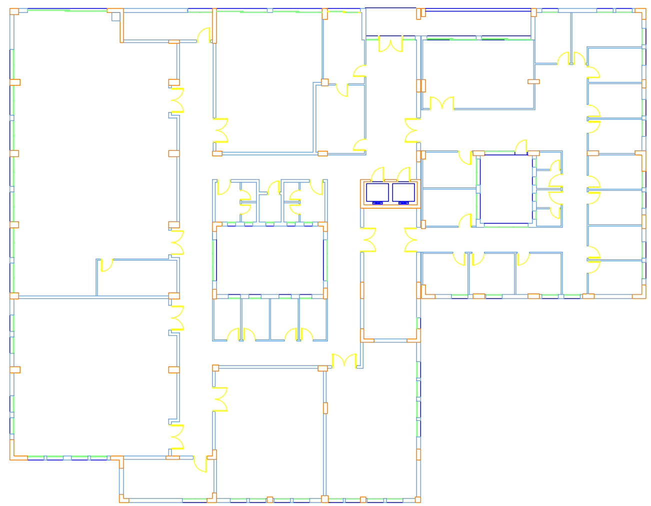


Figure 4.6: Architectural plan of first floor after processing

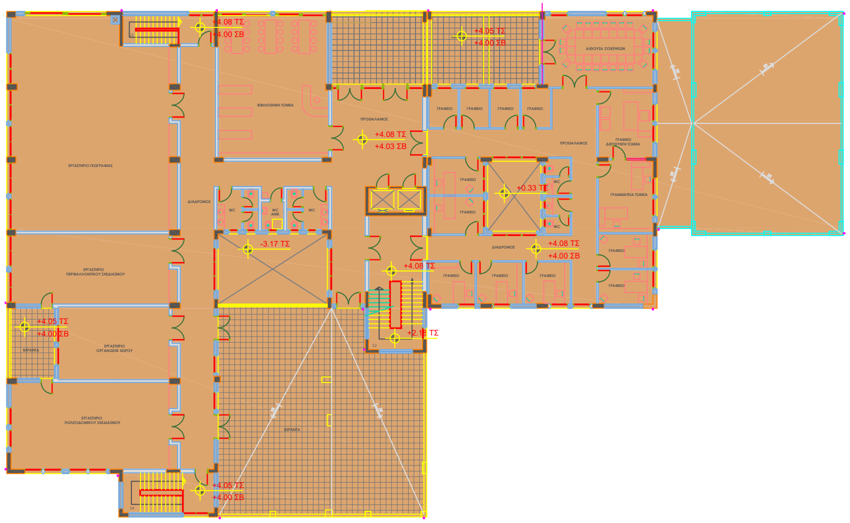


Figure 4.7: Initial architectural plan of second floor

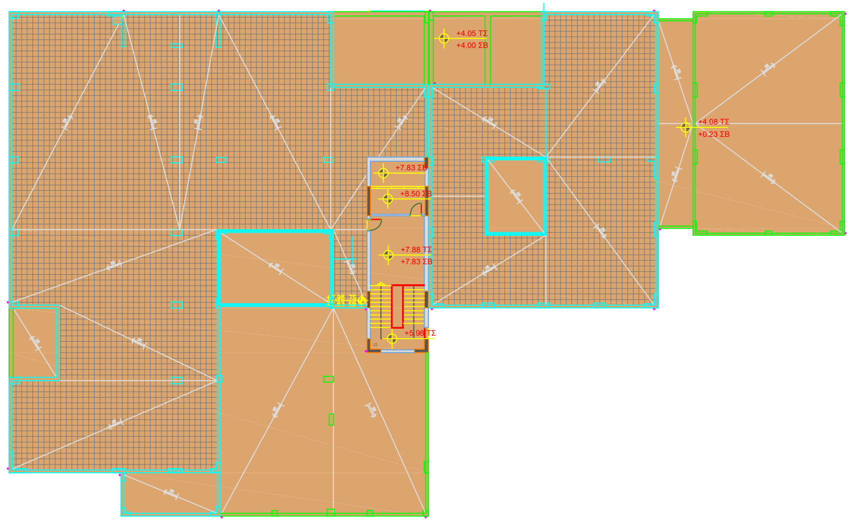


Figure 4.8: Initial architectural plan of terrace

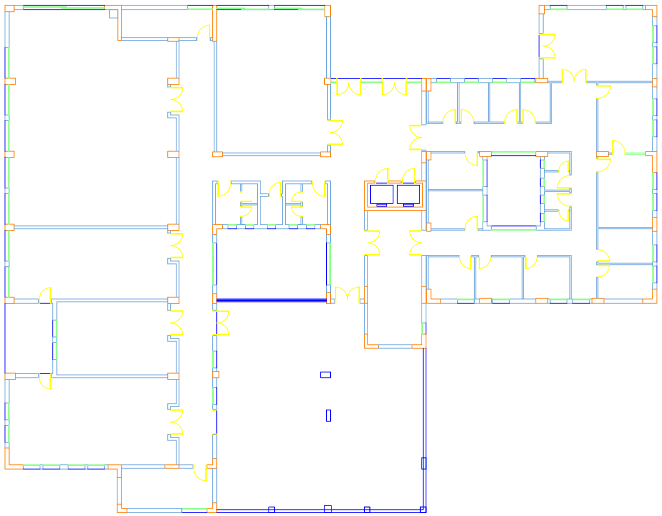


Figure 4.9: Architectural plan of second floor after processing

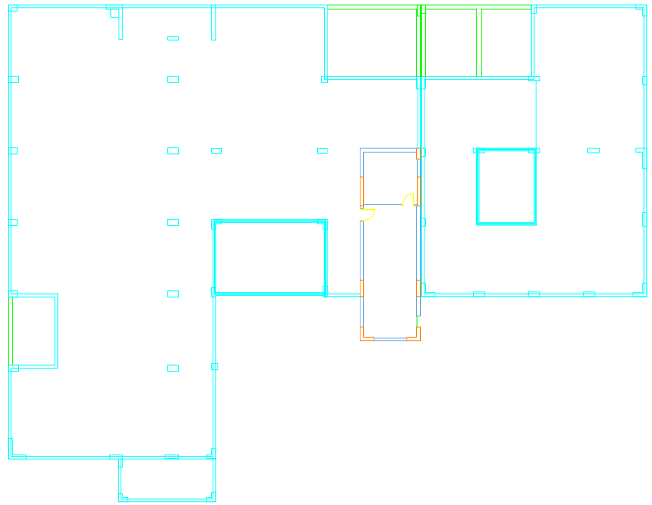


Figure 4.10: Architectural plan terrace after processing

4.5 Design Process of 3D Model using SketchUp

The process of proper modelling semantically enriched buildings requires internal order or structure to the 3D file. The structure of the 3D model is determined from the structure of the information models, according to which the building will be semantically defined. Therefore, the modelling should be in accordance with CityGML structure.

First step of the process was the import of 2D architectural plans in SketchUp environment (Figure 4.12). The different layers and blocks were directly transferred into SketchUp. Each floorplan was grouped in different layers. Making use of "S4U make faces" plug-in the polygons from the 2D architectural plans were transformed to faces.

The building was separated in four different building parts. According to CityGML documentation "A building part is a sub-division of a building that is homogeneous related to its physical, functional or temporal aspects and may be considered as a building". The subdivision of a building can be done by different criteria, e.g. structural like number of floors, roof type, height, construction method or administrative like building function, ownership, year of construction. The separation of Veis building was based on the criterion of height.

The next step was the 3D modelling of different elements and the creation of objects, through the grouping mechanism. Two very important and synergistic systems are used to organize model geometry in SketchUp. Components and groups consolidate geometry into bundles. These bundles can then be placed on the second system, called layers, providing users with the ability to toggle the visibility of the bundles. SketchUp models can contain loads of geometry. Having a lot of visible objects can impede the ability of the user to navigate within the SketchUp 3D environment because the geometry starts to get in the way. By placing edges and faces on layers and using those layers properly, the user can minimize or eliminate the challenges created by abundant geometry. Components and groups are the mainstays of constructing and organising geometry. They are almost identical. However, they have one important difference: Editing or altering a component affects every instance of that component. Although multiple copies of a group are identical, editing one group has no effect on other copies of that group. IFCManager tool works with the components mechanism, therefore the semantic objects will be grouped as Components.

The basic principle followed during the design of the building was the modelling from the interior structures to the exterior. The workflow of the procedure is summarized below, while the representation of the model during the different modelling stages is demonstrated in Figure 4.13.

1) Modelling of interior installations (columns, beams, stairs)
2) Modelling of interior walls
3) Modelling of the openings of interior walls
4) Modelling of exterior walls
5) Modelling of the openings of exterior walls
6) Modelling of exterior building installations

During the modelling process every logical object is separated from the others as a different entity. The Outliner reveals that there is actually a hierarchy of objects in the model. This hierarchy was created by nesting components, which means that all the distinct components have been grouped and the resulting group defines the Building. After the formation of groups, the different entities were categorized in different layers. The geometry of boundaries surfaces can be represented only with individual faces, which cannot belong to a group or component, since the CityEditor tool doesn't allow the grouping of BoundarySurfaces. Therefore, different layers, were defined for the classification of boundary surfaces, according to CityGML format.

Figure 4.11 represents the organisation of the elements in AutoCAD and SketchUp Layers, as well the hierarchy of groups in Outliner.

One important aspect during the modelling procedure is the orientation of the faces, Materials can be imported and exported upon the front and back side of faces. In CityGML, materials on back sides are marked with the isFront element of a surfaceDataMember. The correct orientation of the faces is needed, for the proper definition of the CityGML appearance structure. Furthermore, in some circumstances, self-intersecting polygons may occur while modelling. Those are not recognized during the export; thus, the proper modelling will eliminate the possibility of error.

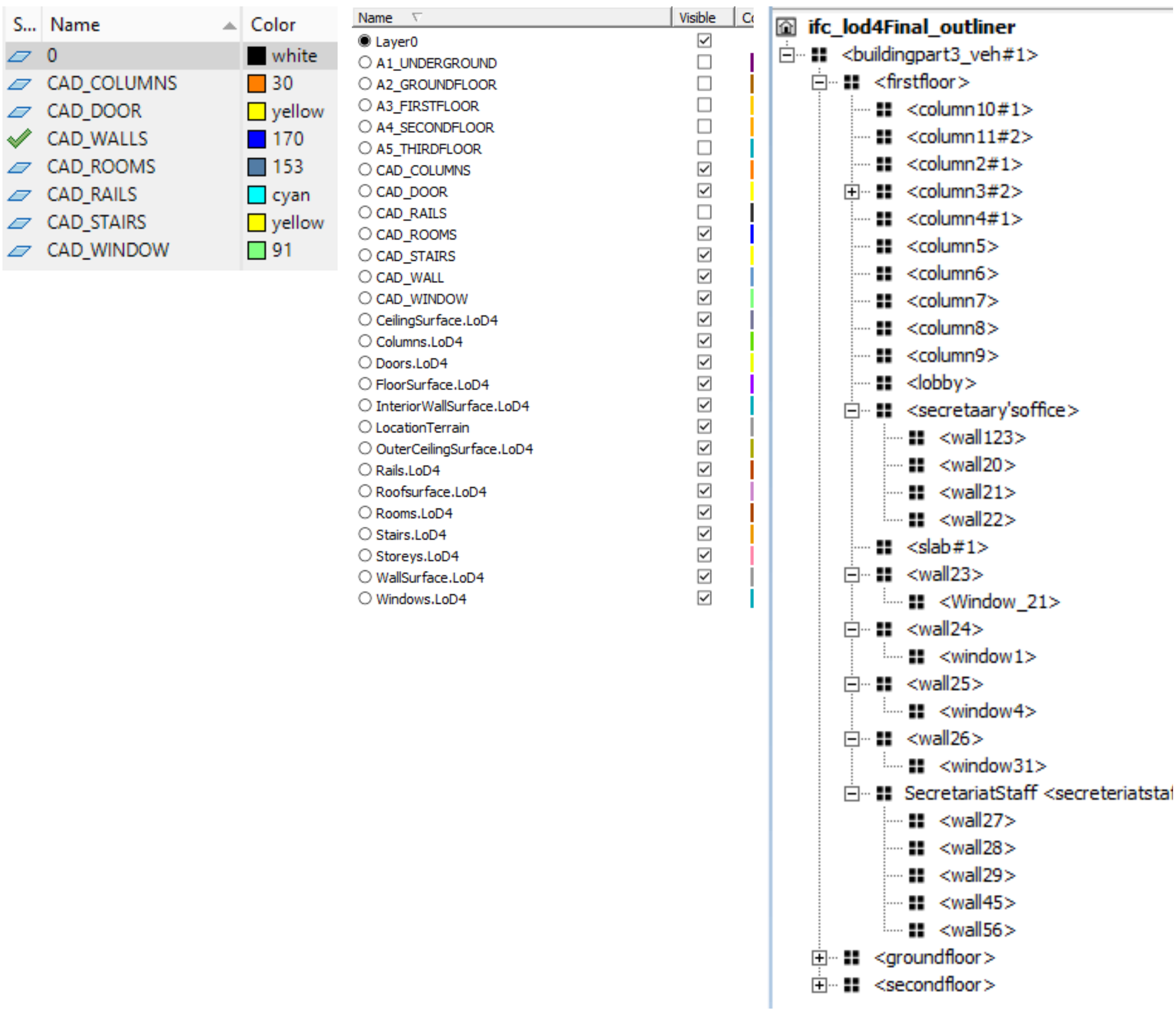


Figure 4.11: (left) AutoCAD Layering, (middle) SketchUp layering, (right) Grouping of components in Outliner

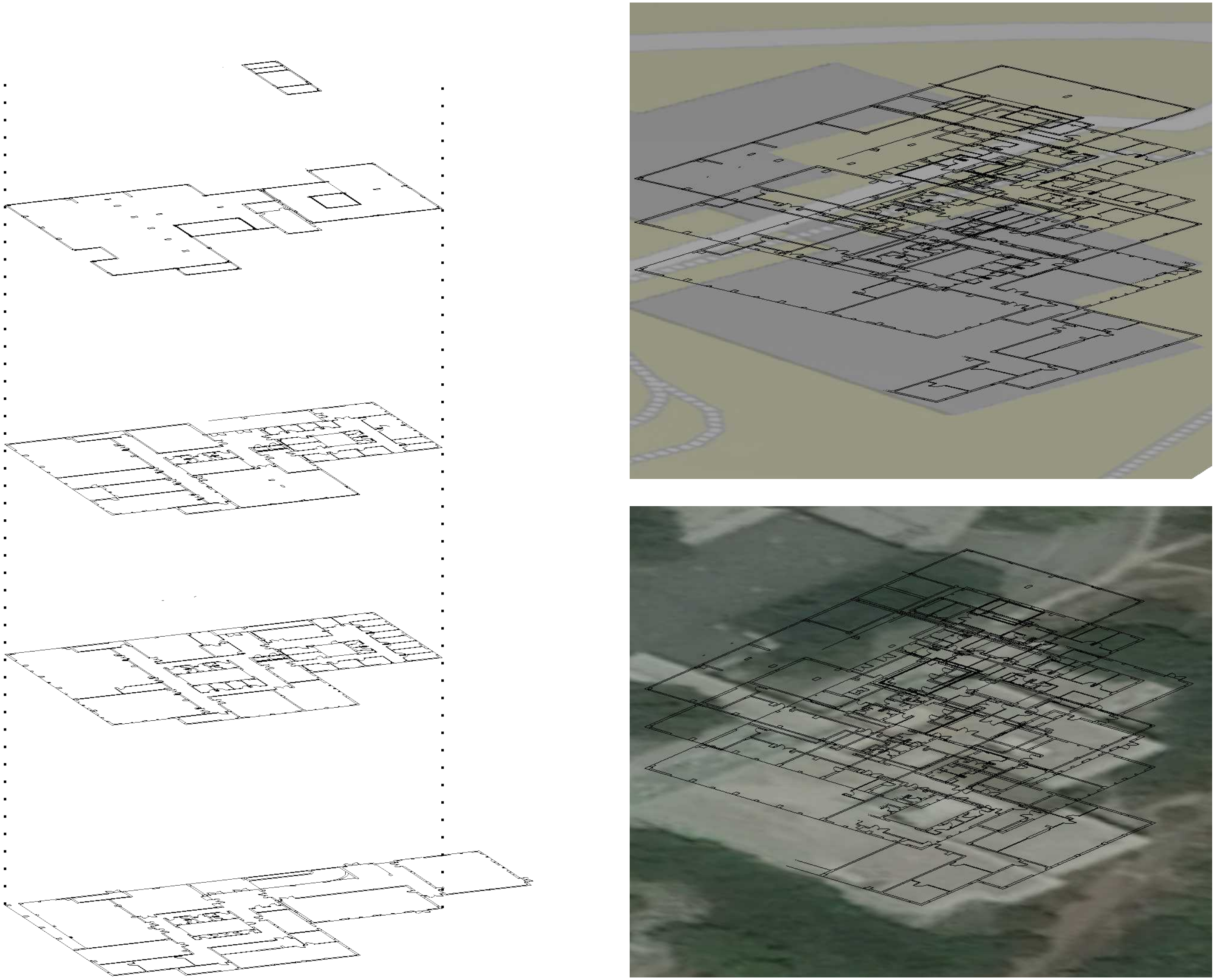


Figure 4.12: 2D architectural plans in SketchUp

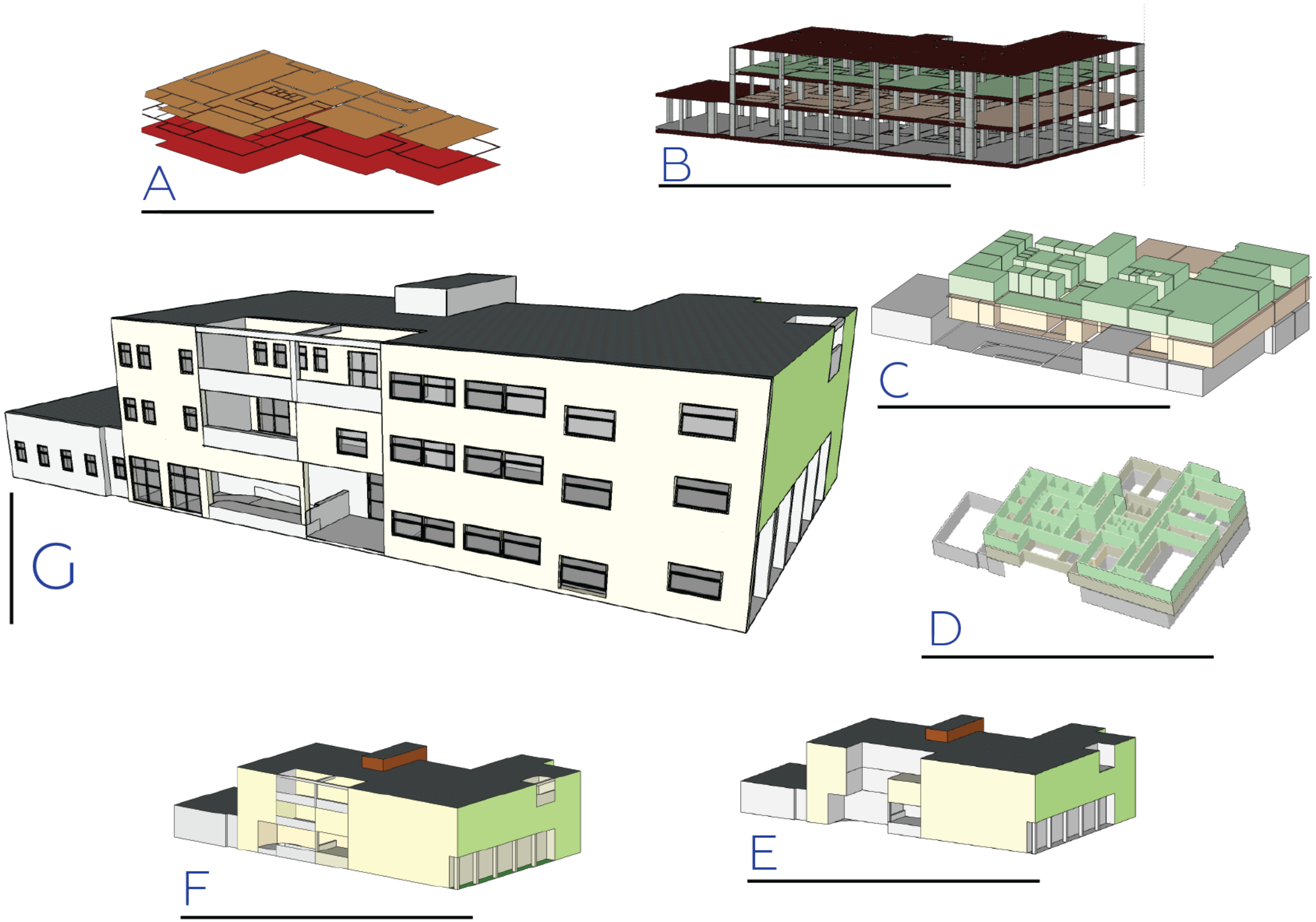


Figure 4.13: (A) Footprints of groundfloor, footprints of rooms and walls of groundfloor designed in SU from 2D architectural plan, (B) interior building installations, (C) Rooms of Veis building as solids, (D) Decomposition of rooms to walls, (E) building shell part of Veis building, (F) building shell part with the exterior building installations

4.6 IFC Classification

In SketchUp environment, the generated 3D models are only 3D geometric representation. Using the grouping or component mechanism, these shapes are transformed into objects, to which the user can assign semantically information.

As previously mentioned, there are around 900 classes defined in the IFC schema. However, the most relevant classes for CityGML are only a subset of these, IfcSpace and all the subtypes of IfcBuildingElement. All other classes either represent movable objects or are abstract classes without geometry. For each IfcObject in an IFC file, we verify whether it has a geometry and whether it is contained inside a building; Filtering these objects leaves us to deal with only objects having meaningful mappings in CityGML. These objects are described below.

According to IFC documentation, all spatial structure elements inherit the concepts associated to products, which are provided by IfcProduct. The spatial structure is the primary structure for building projects. The following four different concepts are subsumed under the IfcSpatialStructureElement entity:

- Project (uppermost container of all information)
- Site (also site complex and part of site)
- Building (also building complex and building section)
- Building story (also partial building story)
- Space (also partial space)

These different entities are contained by each other, providing a clear hierarchical structure for the building project. The four subtypes IfcSite, IfcBuilding, IfcBuildingStorey, IfcSpace are used to represent the levels of the spatial structure. Each has: a name (by IfcRoot.Name) and a description (by IfcRoot.Description). The spatial structure of a building project is created by using the fundamental decomposition relationship. The subtype IfcRelAggregates is used to link the instances of IfcProject, IfcSite, IfcBuilding, IfcBuildingStorey, IfcSpaceand establishes a hierarchical structure. Whereas the IfcProject, IfcBuilding and IfcBuildingStorey are mandatory levels for the exchange of complex project data, the IfcSite and IfcSpace represent optional levels. Each instance of IfcProject, IfcSite, IfcBuilding, IfcBuildingStorey, IfcSpace is connected to other instances of the spatial structure by an instance of IfcRelAggregates, where the single RelatingElement points to the element at the higher level and the 1 to many RelatedElements point to the elements at the lower level of the hierarchy. The spatial structure and the layout of Veis building is represented in Figure 4.15

All building elements are exchanged by an IFC file as instances of subtypes of the abstract entity IfcBuildingElement. The hierarchy chart of the different building elements is presented in Figure 4.14. Walls are exchanged by an IFC file as instances of IfcWall or IfcWallStandardCase. IfcWallStandardCase handles all cases, where a wall has a single material thickness (both in ground view and cross section) along the wall path, and IfcWall handles all other cases of walls.

Openings are exchanged in IFC files as instances of IfcOpeningElement. An IfcOpeningElement forms a void, which is created within another element. This is primarily independent from the kind of element, which is voided (like a wall opening, or a slab opening, or an opening in a roof) and also from the fact, whether the void goes through the whole thickness of the element are whether it is only partially voided.

Slabs are exchanged by an IFC2x file as instances of IfcSlab. The concept of slab comprises all building elements that are usually planar and non-vertical, such as: floor slabs, roof slabs, stair landings.

The IFC file structure is based on the structure of the Outliner in SketchUp Environment. The spatial structure of Building Part 3, according to IFC standard is represented in Figure 4.16. The IFC File was imported and displayed in FZK Viewer.

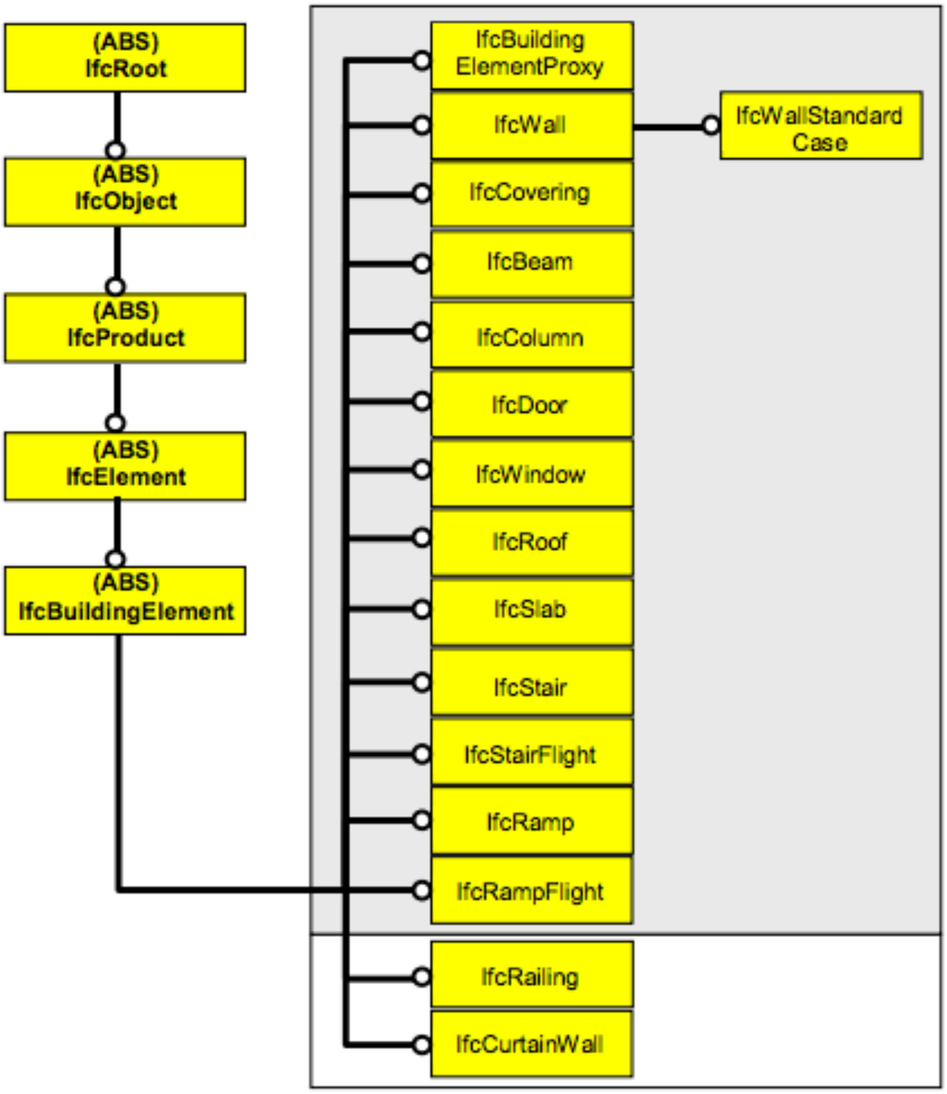


Figure 4.14: Hierarchy chart of building elements used in the model

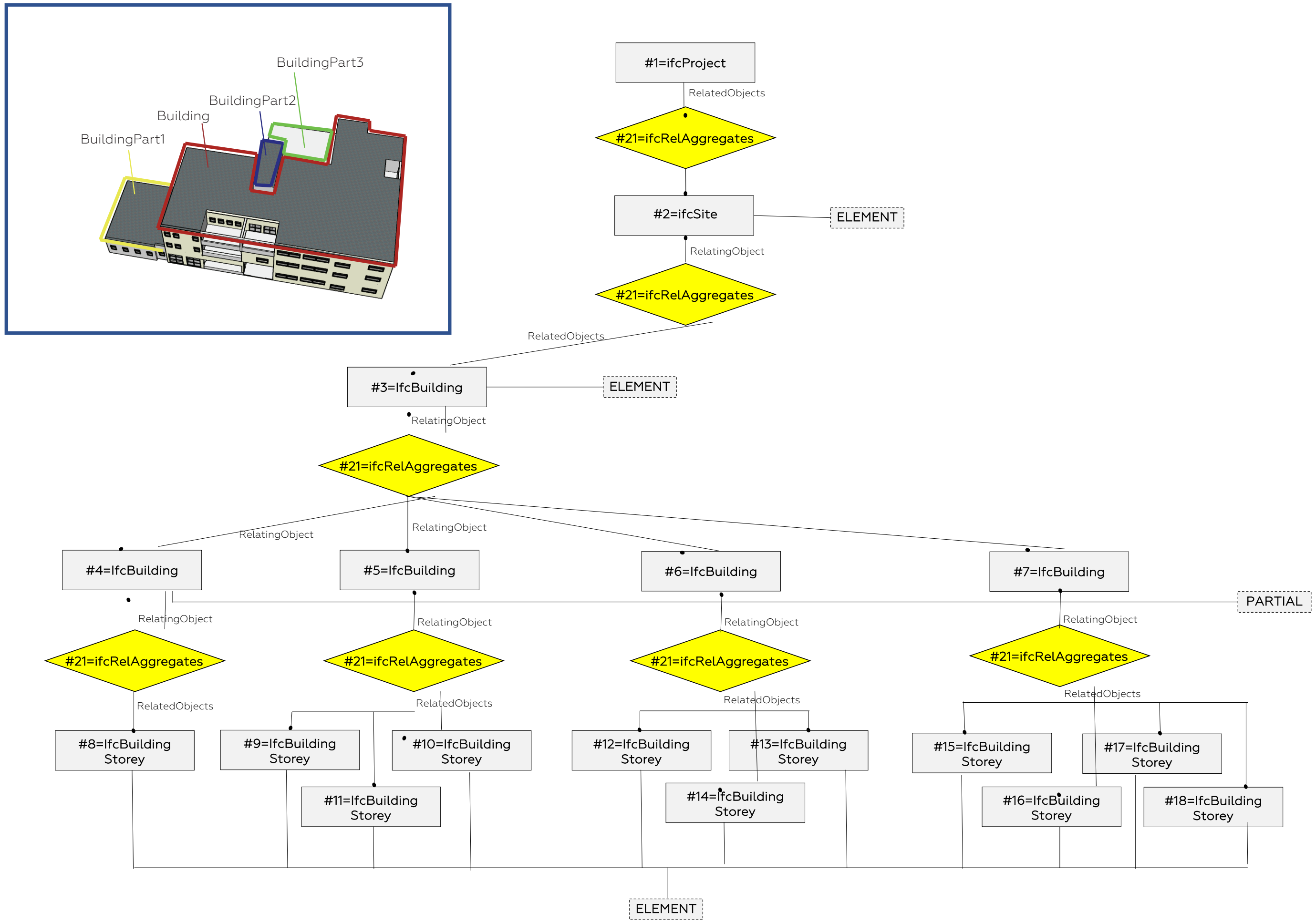


Figure 4.15: Decomposition of Veis Building

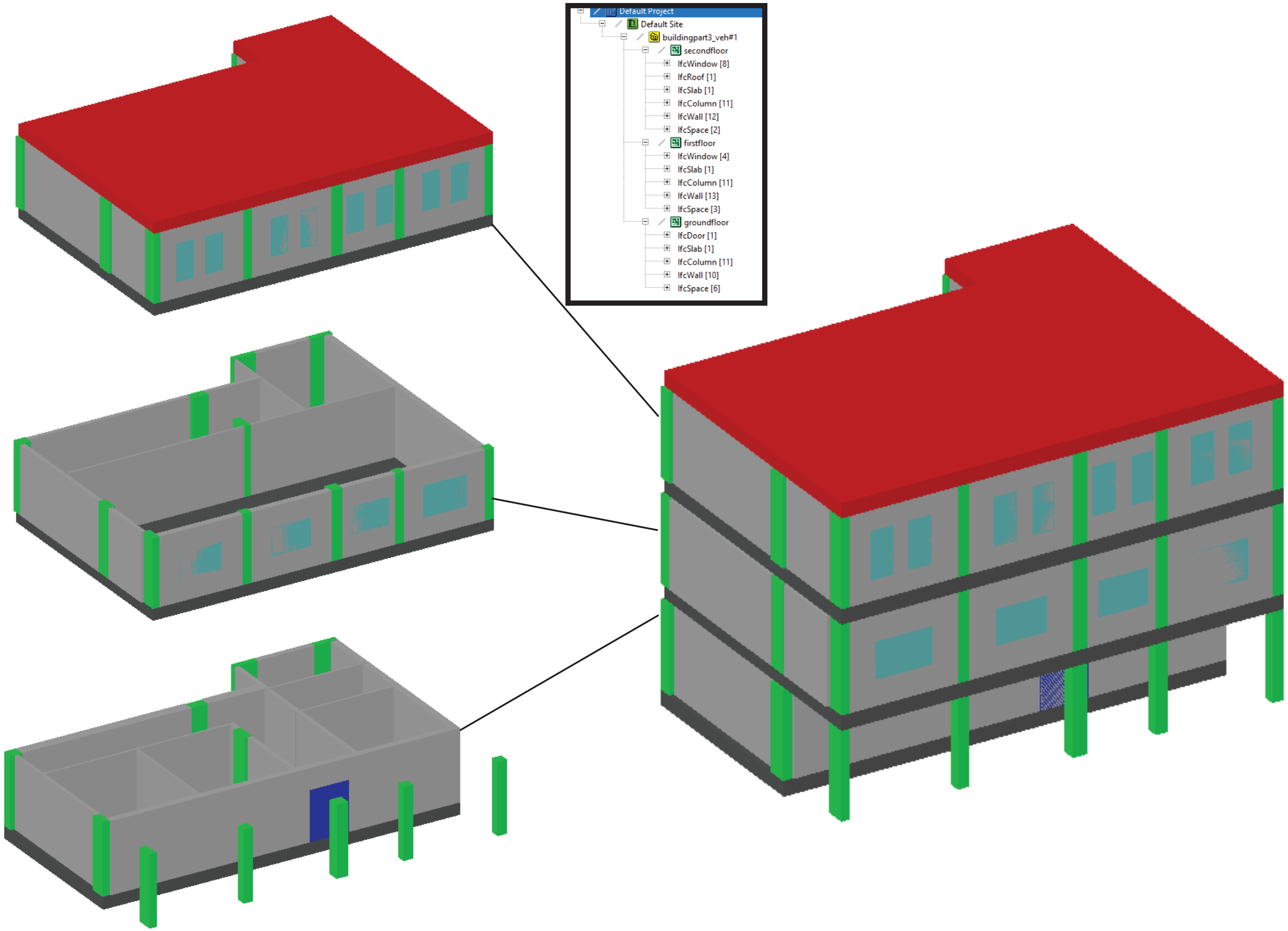


Figure 4.16: Generated IFC Model of Building Part 3, in FZK Viewer

4.7 Semantic Mapping between IFC and CityGML

Transforming information from IFC to CityGML requires a two-step approach: transforming semantic information and transforming geometries. Since the objects (classes) in these two models are very diverse, the two steps cannot be performed separately. An object in one of the models might be mapped to a group of objects (and vice versa), which requires a careful consideration on the order or converting geometries and semantics. Since the geometric components were created in SketchUp, using different kinds of the grouping mechanism for each format, the transformation of geometry was achieved within the SketchUp environment.

Regarding the semantic transformation the following rules were followed.

Building is a common object for both IFC and CityGML and it is represented through IfcBuilding and CityGML Building respectively. Subdivisions of IfcBuilding are matched with the BuildingParts of CityGML.

In IFC, a building is smoothly structured by breaking down the building into storeys and into spaces that form a specific storey. In CityGML 2.0, there is no explicit definition of spaces or stories. However, a storey can be represented as an explicit aggregation of all building features on a certain level using CityGML's notion of CityObjectGroups. This would include Rooms, Doors, Windows, IntBuildingInstallations and Building Furniture. At first place, the grouping of all semantic objects belonging to a specific storey is required and secondly the definition of attributes of the corresponding CityObjectGroup object as following:

- The class attribute shall be assigned the value "building separation".
- The function attribute shall be assigned the value "LoDXStorey" with X between 1 and 4 in order to de- note that this group represents a storey wrt. a specific LoD.
- The storey name or number can be stored in the gml:name property.

The implementation of this structure is not supported by the different applications. Thus, the introduction of AbstractBuildingSubdivision in the Building module of CityGML version 3.0 will allow the inclusion building storeys as well as building units like apartments or public spaces in buildings.

The IfcSpace objects correspond to the spaces that are bounded by the walls, roofs, floors and other structural components. But they can also be spaces outside of the building that are bounded by virtual boundaries (buidingSMART, 2013). Since this thesis, is focused on the interior structure of the building model, IfcSpace object will be matched with CityGML Room entity.

The door and window classes in the IFC model (IfcDoor and IfcWindow) are referred from the (IFC) opening element (IfcOpeningElement) but, in contrast to the CityGML object model, in IFC model, neither IfcOpeningElement is an abstract class, nor the IfcDoor and IfcWindow are the subclasses of the IfcOpeningElement. The IfcOpeningElement is an element that is used the describe the geometry and semantics of an opening which can contain multiple door and windows, thus an IfcOpeningElement can refer to multiple IfcDoor and IfcWindow elements. However, the Door and Window objects in CityGML can be generated by the information acquired from the IfcDoor and IfcWindow classes.

Building installations (ramps, chimneys, balconies, column, etc.) are defined differently in both IFC and CityGML. In IFC, they are defined as normal building elements (like walls, slabs, etc.) with the same geometric concepts. However, in CityGML building installations are specified in a separated object named building installation. In the latter case, geometries of these installations (ramps, chimneys, balconies, beams, column, etc.) are defined by multisurfaces that construct the objects and stored in CityGMLMultiSurface. Therefore, the IfcBuildingElements are related to BuildingInstallations and InteriorBuildingInstallations of CityGML. It is possible different IfcBuildingElements to be grouped under one BuildingInstallation.

The mappings between IFC and CityGML are summarized in Table 4.1 The Table demonstrates also the hierarchy of the different semantic objects, which was followed during the modelling process.

IFC class	CityGML class	SketchUp Grouping	AutoCAD Layering
IfcBuilding	Abstract Building	Building or BuildingPart (1st level)	N/A
IfcStorey	CityObjectGroup	Storey (2nd Level)	N/A
IfcSpace	InteriorBuildingInstallation/BuildingInstallation	Room (3rd Level)	N/A
IfcStandardCaseWall	InteriorWallSurface or WallSurface (depending on the boundaryType)	InteriorWallSurface or WallSurface	InteriorWalls
IfcWindow	Window	Windows (5th Level)	Windows
IfcDoor	Door	Doors (5th Level)	Doors
IfcSlab	Groundsurface, FloorSurface, CeilingSurface (depending on IfcSlab-TypeEnum)	Groundsurface, FloorSurface, CeilingSurface (ungrouped)	N/A
IfcRoof	Roofsurface	Roofsurface (ungrouped)	ExteriorWalls
IfcColumn	Column	Columns (5th Level)	
IfcFurnishingElement	BuildingFurniture	Furniture (5th Level)	
IfcStair	BuildingInstallation	Stair (5th Level)	Stairs
IfcRailing	BuildingInstallation	Railing (5th Level)	Rails

Table 4.1: IFC-CityGML mapping and demonstration of grouping 2D Architectural floor plan and in 3D model

4.8 Generating CityGML models using CityEditor

The CityEditor plugin was used for exporting the model in CityGML format. CityEditor is not an independent program but an extension for Trimble SketchUp allowing the import, editing and export of CityGML models and 3D geodata into SketchUp. The CityEditor provides tools for the displaying, editing, adding, and deleting of standard attributes for buildings and surfaces, as well as of generic attributes for buildings. With its installation, a separate sub menu is created within the menu Plugins. The current version of CityEditor is 2.7.

The objects modelled using SketchUp tools can be classified using the CityGML object types and BoundarySurface types. Different LoDs are also supported.

4.8.1 Main Characteristics of CityEditor 2.7

One of the most useful features of CityEditor is its ability to interface with data in various formats, increasing the interoperability level of SketchUp. Table 4.2 shows the kinds of data the user can import and export.

IMPORT	EXPORT
Building models from. gml, .dxf, .ply, .cco, .3mf, .stl and .shp files	CityGML models, as 3D-PDF documents, as CityBrowser projects
Terrain models from.asc, .ras, .xyz, .adf, .tif, terrain grids from .asc, .dem, .tif and .adf files, regularly and irregularly distributed terrain points from .ras and .xyz files that may be processed to a grid, triangulated terrain models from .dxf files	3D models in 3D Studio (.3ds), Alias Wavefront (.obj), OpenInventor (.iv), OpenSceneGraph Binary (.ive, .osgb), OpenSceneGraph ASCII (.osg), AC3D (.ac), 3D Manufacturing Format (.3mf) and Stereolithography (.stl).
Terrain textures from .jpg, .png, .bmp, .tif, .ecw and .jp2 files, OpenStreet- Map services or WMS servers,	
Vector data from .shp, .dxf, .gml, .json, .geojson and .tab files	
Point clouds from .xyz and .csv files that may optionally contain RGB color values	
Triangulated surface models from .ply and .obj files	

Table 4.2: CityEditor import/export mechanism

Furthermore, the main (CityGML related) editing functions provided by CityEditor plugin, during the modelling procedure, are listed below:

- Advanced attribute editing,
- Object typing
- Face typing
- LoD assignment
- Correction functions
- Rule based boundary surface assignment

More specifically, the main menu of CityEditor is structured according to the aforementioned functions. Through the **Model Explorer** the CityGML object hierarchy of the current model can be displayed. For each object a series of CityGML-specific metadata is listed. Objects that contain sub-objects can be folded out in a tree diagram-like structure so that sub-objects may be inspected as well.

A major aspect of the CityEditor tool is the application of a **rule-based classification engine**, which allows the automatic classification of surfaces based on their normal orientation. Before the classification process can be started, the classification of criteria has to be specified in the form of one or more ranges of surface normal elevation angles. The surface type that is to be assigned to surfaces whose normal falls within the defined elevation range can be chosen from a drop-down menu. The specified elevation ranges are visualized as colored areas in the half-circle diagram on the right side of the Surface Classifier dialog. Additional rows for defining elevation range criteria can be added by clicking the Add classification button.

The object types supported by the current version of CityEditor, that are related to the building module and were used in this case study, are: Building, BuildingPart, BuildingInstallation, IntBuildingInstallation, Window, Door, CityObjectGroup, Room, BuildingFurniture

The imported faces within the listed object types can be distinguished as follows: RoofSurface, WallSurface, GroundSurface, ClosureSurface, CeilingSurface, InteriorWallSurface, FloorSurface, OuterCeilingSurface and OuterFloorSurface

4.8.2 Semantical structure of the model using CityEditor

Having created the building components as described in Section 4.5 the classification of the entities was carried out according to CityGML documentation. The model was generated in four LoDs, starting from LoD4 and ending to LoD1, through the generalisation process, proposed by (S. U. Baig & Abdul-Rahman, 2013) and presented in Figure 4.17. Geometries and semantics are transferred from higher LoD to lower LoD with the aim to derive coarse LoD based on CityGML's generalisation specifications. LoD4 of building modelled in CityGML contains interiors along with LoD3. During the derivation of LoD3 from LoD4, movable and immovable objects of class at LoD4 such as IntBuildingInstallation, BuildingFurniture, roomInstallations, and InteriorRoom are removed. Immovable class objects, such as IntBuildingInstallations composed of interior stairs and railings which are permanently attached to the building structure cannot be removed directly so are considered to be part of LoD3. However, objects of the class IntBuildingInstallation can either be associated with a room, or with the complete building/building part. After derivation of LoD3 from LoD4, the resulting object classes at LoD3 composed of OuterBuildingInstallation, Wall, Roof, and Openings (Door, Window). With the aim to derive LoD2 from LoD3, openings are removed, followed by the filling of resulted holes, while other outer installations are projected onto the ground for simplification purpose. OuterBuildingInstallation class is related to outer components of a building therefore strongly affects the outer characteristic of the building. This class object contains chimneys, stairs, antennas, balconies, or attached roofs above stairs and paths. These components are removed with the aim to produce LoD1. Structural features drafts

of building models need to be clubbed and formed a simple block at LoD1. The resulting LoD1 become a building block without a roof in a proper shape, and simplified walls as a flat plane. The proper layering of different components was proved to be of high importance for the generation of the different LoDs.

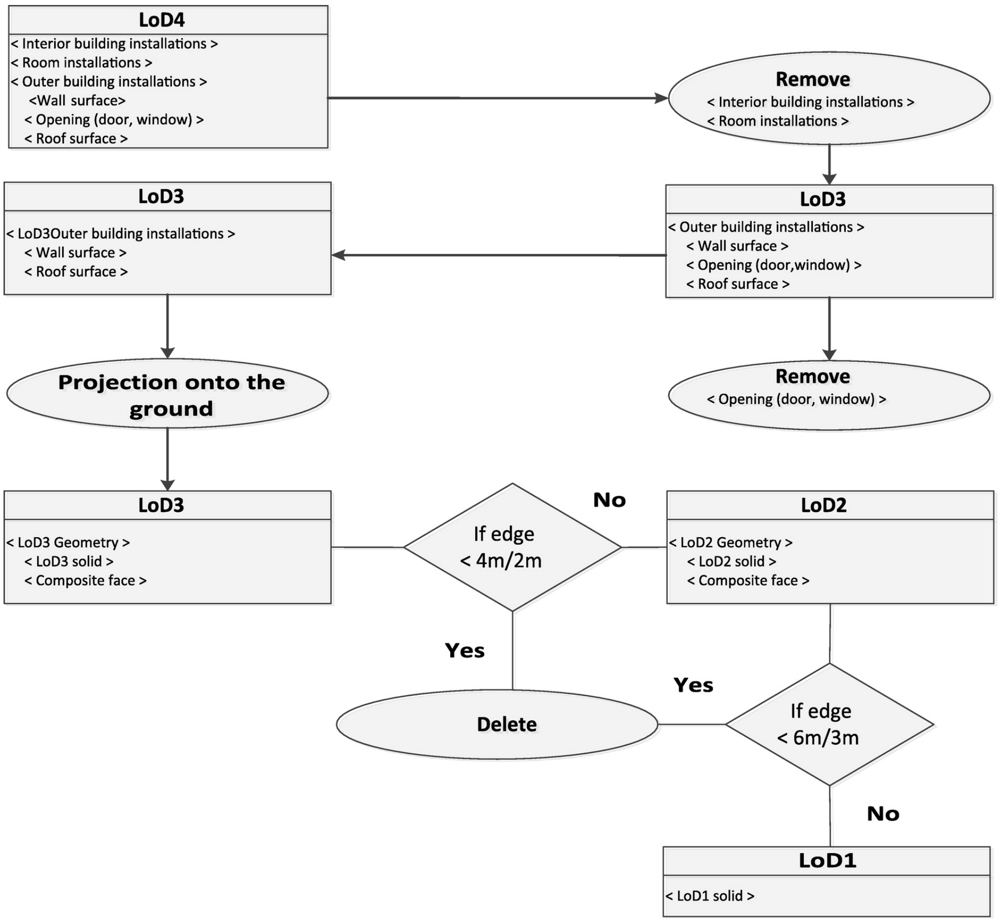


Figure 4.17: Workflow of the generalisation within the framework of CityGML (S. U. Baig & Abdul-Rahman, 2013)

Firstly, the **rule-based classification** engine was used to automate the process. Three successive kinds of classification were defined, as presented in Figure 4.18. The first one was implemented only to the exterior building shell. The second one was implemented only to the room shells, while the third one was performed to the exterior shell of each building storey. Then the manual **correction of the errors** was required.

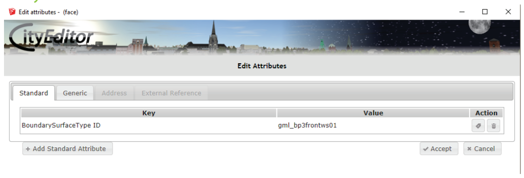
Next step of the procedure, was the **definition of the surface members** that comprise a Wallsurface or an InteriorWallSurface. This was achieved using the same boundarySurfaceID for each polygon, which belongs to the standards attributes of the boundarySurface. The polygonID and the LinearRingID were given automatically during the export of the model. This step was applied to the following cases:

- a boundary surface has different colours or textures
- a boundary is nonplanar (Figure 4.19)
- it is intended to indicate a storey structure (Figure 4.19)

Subsequently, the transformation of each component to a CityGML entity was completed manually, simultaneously with the selection of the **geometry selection**. For the Building and BuildingParts entities, the following Geometry can be used: LoDXMultiSurface, LoDXFootPrint, LoDXRoofEdge, LoDXSolid. For BuildingInstallation and BuildingFurniture LoDXGeometry was used. For the openings the LoDXMultiSurface was chosen, while for Rooms LoDXMultiSurface and LoDXSolid.

Figure 4.18: Implementation of rule-based classification in three stages.

```
<bldg:boundedBy>
<bldg:WallSurface gml:id="gml_bp3frontws01">
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="_surfacemembersgml_BD.3yExaZklVIgojExKoi2n_PG.Tze6KYusflSJsfplKJ1D">
<gml:exterior>
<gml:LinearRing gml:id="_surfacemembersgml_BD.3yExaZklVIgojExKoi2n_LR.FzC0Eevuy4zKOkcKfwws">
<gml:posList srsDimension="3">
12.81 12.509 4.0
10.46 12.509 4.0
10.46 12.509 7.0
12.81 12.509 7.0
12.81 12.509 4.0
</gml:posList>
</gml:LinearRing>
</gml:exterior>
<gml:interior>
<gml:LinearRing gml:id="_surfacemembersgml_BD.3yExaZklVIgojExKoi2n_LR.7TLAtDYoWYEnJiMoCWV9">
<gml:posList srsDimension="3">
10.725 12.509 6.109
10.725 12.509 4.891
12.442 12.509 4.891
12.442 12.509 6.109
10.725 12.509 6.109
</gml:posList>
</gml:LinearRing>
</gml:interior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
....
....
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:WallSurface>
</bldg:boundedBy>
<bldg:lod4Solid>
```



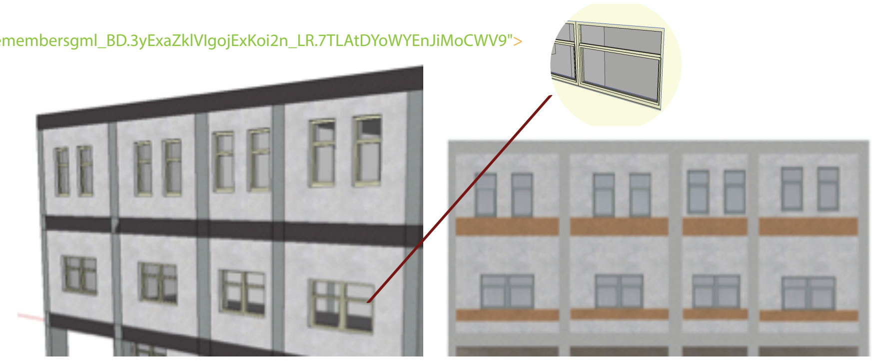


Figure 4.19: Front wall of first floor and second floor, as 1 wall surface consisting of 50 surfaces (revealing areas), (down) gml file structure

For the definition of the **openings**, two classifications are required in order to export a SketchUp group as an opening. The first one is the GroupType classification, which needs to be set to an opening type (Window or Door) and via the selection following a new menu entry will appear in the context menu: Opening Boundary Surface Type. This menu entry can be used to determine the Boundary Surface Type of the opening. Then the definition of the boundarysurfaceID attribute, to which the opening belongs to is required. The latter classification is required in order for the placement of the opening in the exported CityGML model to be determined.

Finally, the IfcBuildingElements that compose a building installation (balcony or loggia in our case) were grouped together as a **buildingInstallation**

Then, the **attribute editing** for each component was completed. The values of each attribute were given according to the Code lists proposed by the SIG 3D in Annex C of CityGML 2.0 documentation. The definition of the attributes was made only for LoD4, since CityEditor supports the maintenance of entities characteristics during the export of the other LoDs.

One more step of the procedure was the import of the **terrain model**. By the use of the Terrain optimisation option the essential terrain features were kept.

During the export of the different LoDs the following parameters were specified:

For the Solid Representation option, the "Composite Surface" element was chosen. The automatic generation of all elements IDs was selected. Regarding the georeference of the model, SketchUp is NOT suitable for editing models with geo coordinates. During an import process, the CityEditorImporter thus relocates a model to the origin of coordinates and saves the geo coordinate offset as a meta-datum in the imported model. If an imported model is opened, it consequently appears on the origin of the SketchUp coordinate system. The determined offset is preserved as a meta-datum attached to the model and can be viewed in the dialog Model Info at any time. During the export to CityGML the offset was automatically added to the exported coordinates.

The visualisation of Veis Building Part 3 (see Figure 4.15) in different LoDs is presented in Figure 4.21, while LoD1 & LoD2 of the whole building and the terrain model imported in FZK Viewer is shown in Figure 4.22.

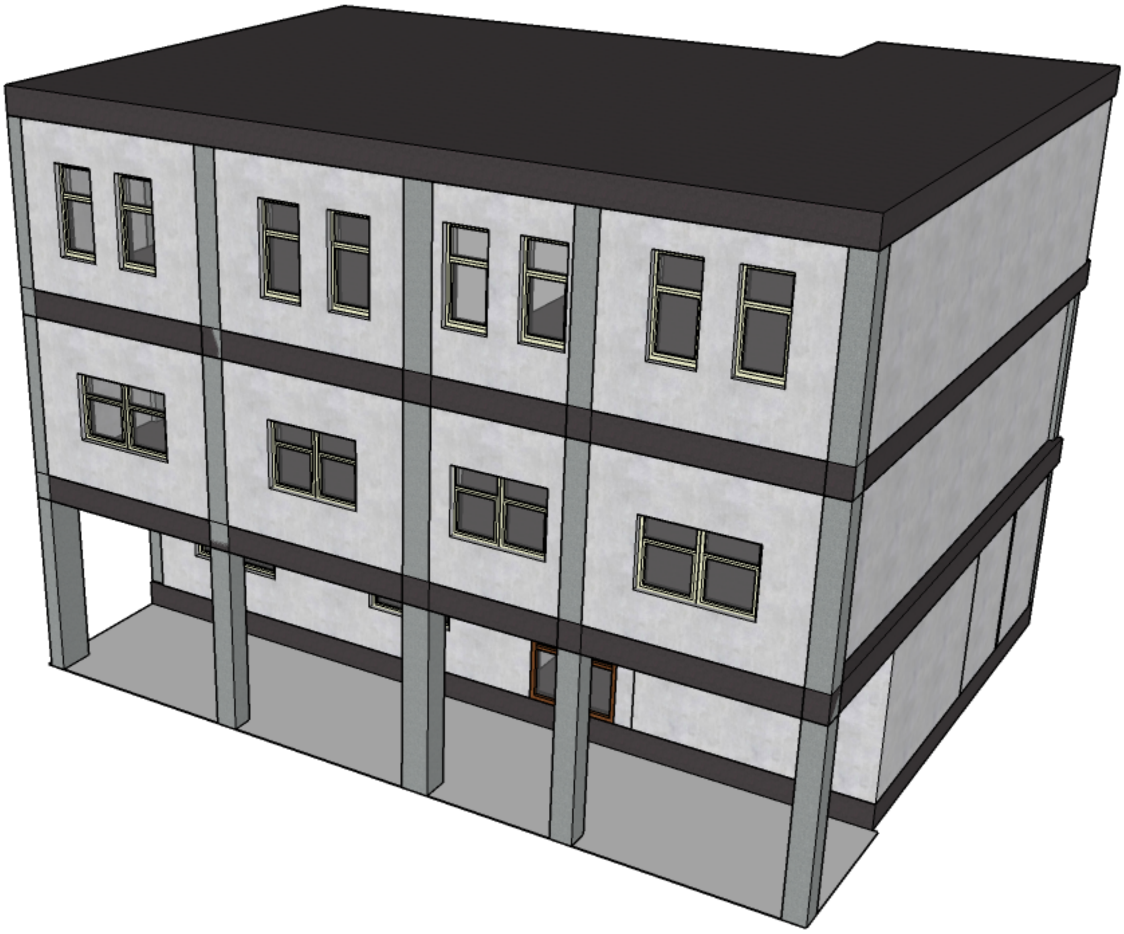


Figure 4.20: SketchUp 3D model of BuildingPart3

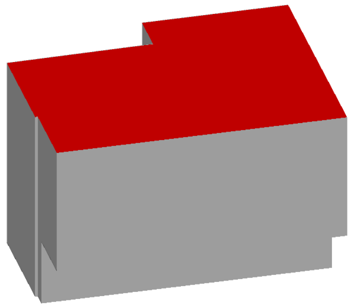
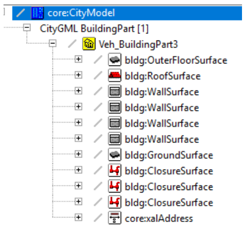
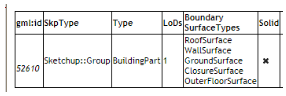
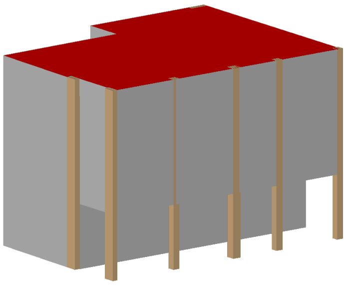
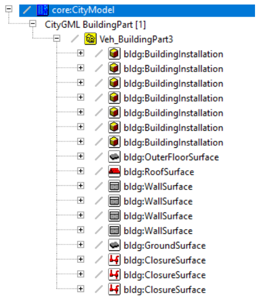
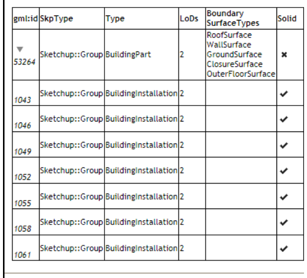
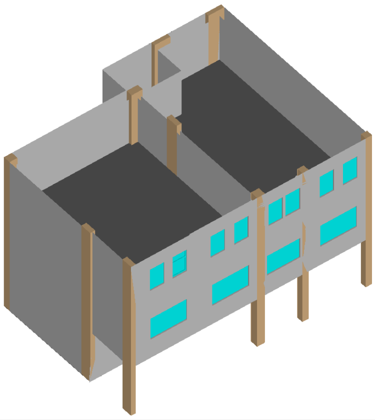
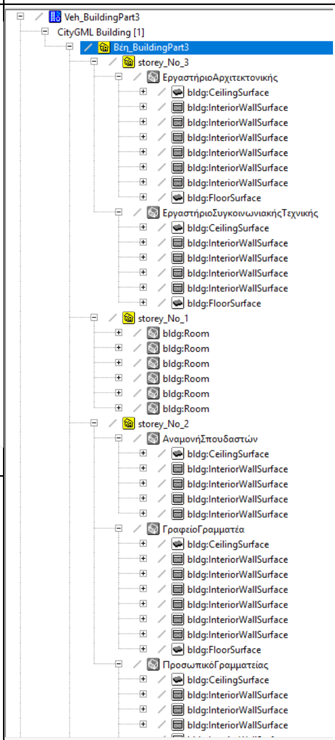
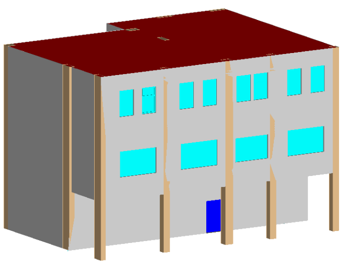
LoD	3D CityGML MODEL	Classification of Entities	Model Explorer (CityEditor)
1			
2			
3			
4			

Figure 4.21: LoD1, LoD2, LoD3, LoD4 of Veis Building using CityEditor

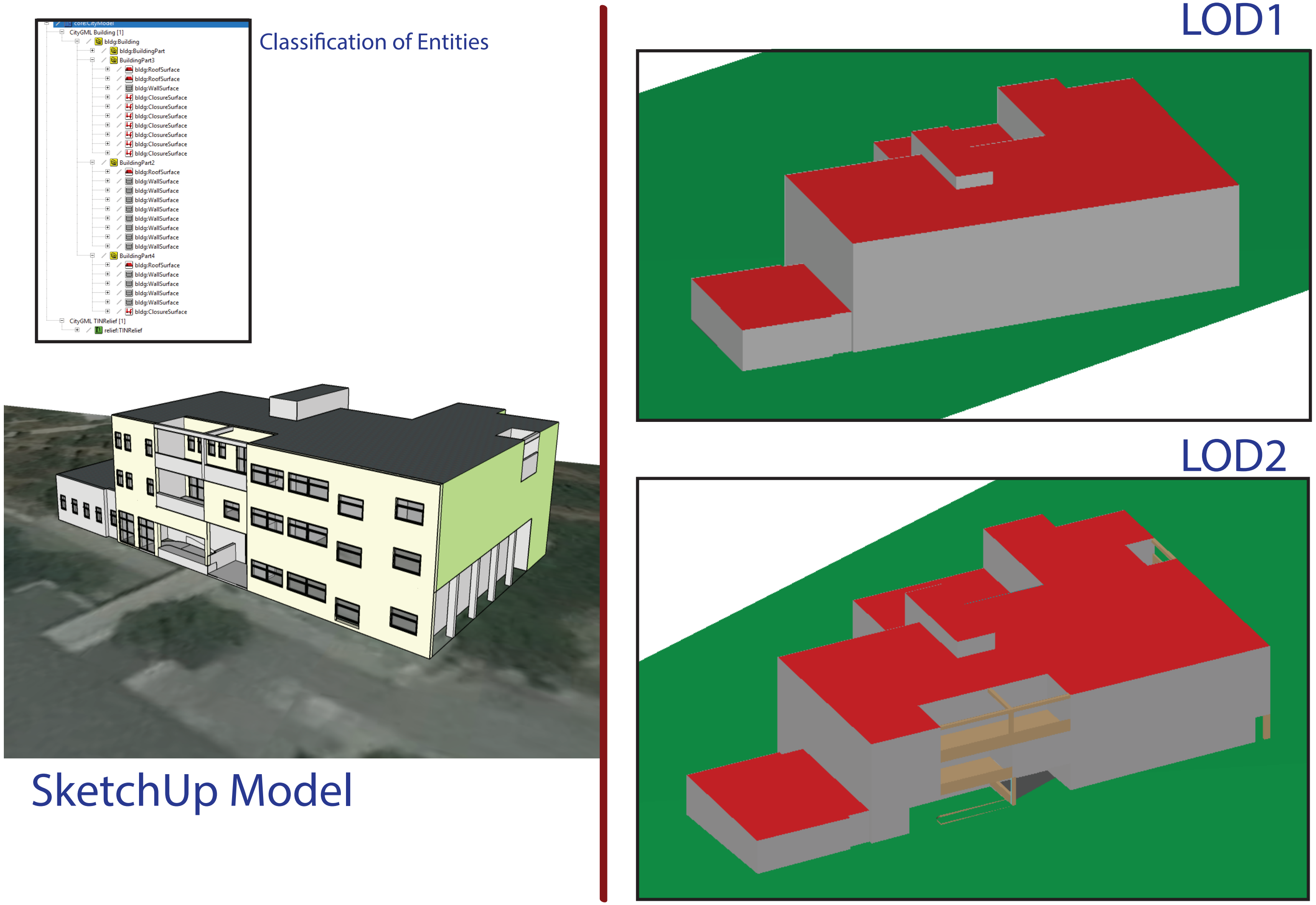


Figure 4.22: LoD1 & LoD2 of Veis Building

5. CONCLUSIONS & FUTURE WORK

The purpose of this chapter is to record the main conclusions of the bibliographic research carried out, as well as to evaluate the methodology. Finally, suggestions for further research on the integration and interoperability of BIM and GIS according to the literature review and the conclusions of the current thesis are provided.

5.1 Conclusions

5.1.1 General conclusions from the literature review

Although the GIS development has a long history, while BIM has also developed for over 10 years, their integrated seems to be quite challenging and is still at an early exploration stage. BIM and GIS originate from different domains and for different purposes and scales. BIM is for detailed 3D building model creation and sharing, while GIS is for geospatial and non-geospatial data management and analysis. However, the benefits by their integration are emphasized by the current developments and the intensive research in this field.

There is an extensive bibliography not only about the integration methods of BIM and GIS, but also the review of these integration methods which are described in Chapter 3. The main findings from the literature review are summarized below.

GIS and BIM cannot replace each other for quite a long time, and they will continue to operate as independent but complementary systems. At present, the priority is to achieve full and effective data interoperability between the two domains. From a data perspective the obstacles in the integration of two domains, is mainly because existing BIM models contain many geometrical and topological inconsistencies which need to be properly handled and often fixed. These may not be problematic when used in a BIM environment. However, these issues are very problematic in GIS applications because we need to perform spatial analyses which often involve complex operations such as Boolean set operations. Thus, functionalities to validate the geometry of a BIM model are required.

IFC and CityGML are representative data formats for BIM and GIS, respectively. Even though there are other formats involved, such as shapefile, these are the most dominant, due to ability of explicit declaration of semantic information. Therefore, the key for the integration of those domains is considered to be the transformation of IFC to CityGML (or vice versa). The core differences between these data models consist the main obstacles for the transformation procedure. IFC and CityGML differ widely in terms of geometrical and semantic model of information, as well as in the concept of LoD. Semantic heterogeneity presents a major challenge when the focus is on the interoperability. Transforming information from IFC to CityGML requires a two-step approach: transforming semantic information and transforming geometries.

From a semantic perspective, a transformation needs to tackle problems due to semantic mismatches between the classes in two models. For example, CityGML model does not provide a class for representing the storey of a building. On the other hand, an entity of the IFC model can correspond to different objects of CityGML (i.e. the

CityGML model provides two different objects for representing the FloorSurface and CeilingSurface of a room, but these surfaces are represented with a single entity in the IFC model (IfcSlab)). However, the mapping of the classes between two models is possible, as described in Chapter 4. IFC model contains all necessary information for representation of buildings in different LoDs of CityGML model. It is possible to define rules for transforming geometrical information from IFC entities to CityGML objects. It is possible to define rules for facilitating semantic matching between two models.

Regarding the application level of the integration approaches., the current solutions exchange is likely to be project-specific. A more generic approach is needed. Additionally, this may largely rely on the extension of CityGML and the standardisation of ontologies of these two areas.

Furthermore, there is an obvious gap of BIM and GIS integrated application in infrastructures and urban districts, compared to buildings, the development of IFC5 and CityGML version 3.0 is expected to fulfil this gap. The extension of the thematic coverage of IFC5 to other domains (tunnel, bridge, transportation), will transform it from a building-oriented standard to an infra-oriented model. In addition, the introduction of the construction module in CityGML and more specifically the new feature type AbstractConstructiveElement and its concrete building, bridge, and tunnel specific subtypes, will allow for mapping constructive elements from BIM datasets given in the IFC standard onto CityGML. The intensive ongoing research and the technology process on these fields, will shortly lead to significant achievements.

5.1.2 Evaluation of the Proposed Methodology

The use of architectural plans as a basis for the 3D modelling of the building is an efficient tool for extracting semantic information. The modification of the floorplans is necessary, in most of the cases, due to the different purposes of the creation of the initial floorplan. Some basic rules are accordingly proposed for the redrawing. The proposed rules for redrawing floor plans focus mainly on the segmentation of the information included in the floor plans, taking advantages of the layering and blocking supported by CAD application. The proposed structuring of data by using the layering mechanism in AutoCAD and SketchUP platforms, eliminates the time for the extraction of a semantically enriched model to both IFC and CityGML formats.

SketchUp application is a useful tool for rapid development of building schematic designs. The main advantage of SketchUp Is that you can easily define a 3D line and stretch it into a surface that aligns with other points in space, supporting easy-to-use direct manipulation. Lines can be used to define a polygon on a surface and can be extruded into or above the surface, to punch holes or define new shapes. Dimensioning feedback allows a user to be precise or imprecise. SketchUp allows 3D shapes and buildings to be defined quite simply, with minimal training. Until last year, SketchUp was described as a modelling solution, that did not utilize BIM design technology since its main purpose was the visualisation. Nowadays, given the availability of different plugins, the creation of semantically enriched models according to IFC and CityGML using SketchUp, is possible.

However, the development of those plugins Is at an initial stage. More specifically, in the CityEditor the declaration is based on the SketchUp's geometry group, where each geometry group must be declared as a semantic object. The main disadvantage is that the semantic declaration process is inevitably workflow specific and, thus the manual declaration of each geometry's semantic content can be a time-consuming and laborious task when the designer has multiple design options. Even though the use of Its rule-based classification engine accelerates the procedure, the addition of further criteria is still considered necessary.

The specific problems encountered during the use of CityEditor tool are listed below: the current version of CityEditor does not support the definition of openings within the room entity yet. Furthermore, the tool does not support the use of cityobjectgroups with parent objects. Therefore, the manual editing of the .gml file is required.

Additionally, it is possible that when importing CityGML models, single faces (particularly roof surfaces) are automatically triangulated. This is due to the fact that the sensitivity regarding the planarity of faces in SketchUp is rather low. In some circumstances, self-intersecting polygons may appear while modelling.

Finally, SketchUp tool cannot load and edit large-scale city models. It is therefore proposed that the use of 2D architectural plans in conjunction with a refined rule-based classification tool for the semantic enrichment of the entities would be a quick and effective method for the production of semantically enriched 3D city models.

5.2 Future Work

The long term goal is to achieve a high degree of interoperability between BIM and GIS. GIS extends the value of digital BIM design data through visualisation and analysis of structures in the context of the natural and built environment. The integration of BIM and GIS enables deeper insight for better decision making, communication, and understanding. By using BIM and GIS together with time information, project participants can better understand the impacts of decisions before, during and after the construction of a project. This not only requires better alignment of the standards, but also methodologies for the conversion to other city objects like, tunnels and bridges. By doing so, instead of making converters for each LoD separately, the different LoD models can be made in a geometrically consistent and semantically connected way.

IFC and CityGML are seen today to be the most prominent and recognised standards for BIM and GIS respectively. To fulfil the demands of urban planning applications and construction analysis, integration of IFC and CityGML is necessary. Considering the great differences in semantics between the BIM and geospatial worlds, developing a formal framework for semantic interoperability is a promising alternative for data exchange between the two worlds. Following the background literature review in this thesis, different approaches to the integration of IFC and CityGML have been presented. Existing approaches, however, do not provide complete integration as they tend to offer only unidirectional transfer, mainly from IFC to CityGML. Additionally, the number of semantic models that support this integration is relatively small compared to the number of geometric models.

It is not hard to believe that high levels of interoperability between BIM and GIS could be achieved, with the advance in information technology and the efforts from individuals and organisations in both domains. At that time, new issues may arise. One is the impact of huge data handling. The data size of a whole city model comprising hundreds or even thousands of buildings would be enormous and can hardly be handled by present technologies. New techniques are needed to handle this issue, such as innovative methods for reducing data size while keeping semantic information intact. Another challenge might be the development of efficient methods for model generation. The current process for creating city models is still time-consuming, especially for those with high level of detail. Even the construction of a simple bridge model would take days, not to mention the creation of a whole city. New efficient methods are needed to facilitate this process if a city model is to be built quickly and efficiently.

The future of BIM/GIS is promising, due to the increasing demand for detailed 3D city models in the area of smart city/digital city studies. However, the aforementioned problems have to be settled before the full data interoperability between BIM and GIS can be realized.

REFERENCES

3DCityDB. (2018). The CityGML Database – 3DCityDB. Retrieved from https://www.3dcitydb.org/3dcitydb/3dcitydbhomepage/. Last accessed on 15 June 2018.

Agugiaro, G., Benner, J., Cipriano, P., & Nouvel, R. (2018). The Energy Application Domain Extension for CityGML: enhancing interoperability for urban energy simulations. *Open Geospatial Data, Software and Standards, 3*(1), 2. doi:10.1186/s40965-018-0042-y

AIA. (2008). *AIA Document E202 BIM Protocol Exhibit*. T. A. I. o. Architects. US. Last accessed on.

Akbarnezhad, A., Ong, K. C. G., & Chandra, L. R. (2014). Economic and environmental assessment of deconstruction strategies using building information modeling. *Automation in Construction, 37*, 131-144. doi:https://doi.org/10.1016/j.autcon.2013.10.017

Akinci, B., Karimi, H. A., Pradhan, A., Wu, C. C., & Fichtl, G. (2010). CAD and GIS Interoperability through Semantic Web Services. In H. A. Karimi & B. Akinci (Eds.), *CAD and GIS Integration* (pp. 199-222): Taylor and Francis (CRC Press).

Albert, J., Bachmann, M., & Hellmeier, A. (2003). *Zielgruppen und Anwendungen für Digitale Stadtmodelle und Digitale Geländemodell*. SIG3D:

Amirebrahimi, S., Rajabifard, A., Mendis, P., & Ngo, T. (2015). A data model for integrating GIS and BIM for assessment and 3D visualisation of flood damage to building. *Locate*.

Amirebrahimi, S., Rajabifard, A., Sabri, S., & Mendis, P. (2016). Spatial Information in Support of 3d Flood Damage Assessment of Buildings at Micro Level: A Review. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W1*, 73-81. doi:10.5194/isprs-annals-IV-2-W1-73-2016

Antoniou, G., & van Harmelen, F. A. (2008). A Semantic Web Primer [Press release]MIT Press, London, UK.

Arroyo Ohori, K., Biljecki, F., Diakité, A., Krijnen, T., Ledoux, H., & Stoter, J. (2017). Towards an Integration of Gis and Bim Data: What Are the Geometric and Topological Issues? *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-4/W5*, 1-8. doi:10.5194/isprs-annals-IV-4-W5-1-2017

Baig, S. U., & Abdul-Rahman, A. (2013). Generalisation of buildings within the framework of CityGML. *Geo-spatial Information Science, 16*(4), 247-255. doi:10.1080/10095020.2013.866617

Baig, U. S. (2013). *A three-step strategy for generalisation of three-dimensional buildings modelled in City Geography MarkUp Language.* (PhD Thesis), Universiti Teknologi Malaysia, Malaysia.

Batty, M., Chapman, D., Evans, S., Haklay, M., Kueppers, S., Shiode, N., Smith, A., & Torrens, M. P. (2000). *Visualising the City: Communicating Urban Design to Planners and Decision-Makers* (Technical Report Paper 26). Centre for Advanced Spatial Analysis (UCL): London, United Kingdom.

Becker, T., Nagel, C., & Kolbe, H. T. (2009). A Multilayered Space-Event Model for Navigation in Indoor Spaces. In J. Lee & S. Zlatanova (Eds.), *3D Geo-Information Sciences* (pp. 61-77). Berlin, Heidelberg: Springer Berlin Heidelberg.

Beetz, J., van Berlo, L. A. H. M., de Laat, R., & van den Helm, P. (2010). *BIMserver. ORG—An open source IFC model server.* In: Proceedings of the CIP W78 Conference, Haifa, Israel. 14-16 July 2010.

Beil, C., & Kolbe, H. T. (2017). CityGML and the streets of New York - A proposal for detailed street space modelling. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV(4/W5)*, 9-16.

Benner, J., Geiger, A., Gröger, G., Häfele, K. H., & Löwner, M. O. (2013). Enhanced LoD Concepts for Virtual 3D City Models. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-2/W1*, 51-61.

Benner, J., Geiger, A., Häfele, K. H., & Joerg, I. (2010). *Interoperability of Geothermal Data Models.* In: Proceedings of the World Geothermal Congress 2010 Bali, Indonesia. 25-29 April 2010

Benner, J., Geiger, A., & Leinemann, K. (2005). *Flexible Generation of Semantic 3D Building Models*. In: Proceedings of the 1st International Workshop on Next Generation 3D City Models, Bonn.

Bentley. (1998). ProjectWise Initiative. Retrieved from http://www.bentley.com/en-US/Products/ProjectWise+Integration+Server/. Last accessed on 11 June 2018.

Biljecki, F. (2013). *The concept of level of detail in 3D city models.* (PhD Research Proposal), TU Delft, Delft, The Netherlands.

Biljecki, F. (2017). *Level of Detail in 3D City Models.* (PhD Thesis), Delft University of Technology, The Netherlands. Retrieved from http://repository.tudelft.nl

Biljecki, F., Ledoux, H., & Stoter, J. (2016a). Generation of Multi-Lod 3d City Models in Citygml with the Procedural Modelling Engine Random3dcity. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-4/W1*, 51-59. doi:10.5194/isprs-annals-IV-4-W1-51-2016

Biljecki, F., Ledoux, H., & Stoter, J. (2016b). An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems, 59*, 25-37. doi:10.1016/j.compenvurbsys.2016.04.005

Biljecki, F., Ledoux, H., Stoter, J., & Zhao, J. (2014). Formalisation of the level of detail in 3D city modelling. *Computers, Environment and Urban Systems, 48*, 1-15.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015). Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information, 4*(4), 2842-2889. doi:10.3390/ijgi4042842

Biljecki, F., Zhao, J., Stoter, J., & Ledoux, H. (2013). Revisiting the concept of level of detail in 3D City Modelling. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-2/W1*, 63-74.

BIMForum. (2017). *LOD Specification Guide*. Last accessed on.

BIMServer. (2015). BiMserver: Open source Building Information Modelserver. Retrieved from http://bimserver.org/. Last accessed on 11 June 2018.

Blut, C., Blut, T., & Blankenbach, J. (2017). CityGML goes mobile: application of large 3D CityGML models on smartphones. *International Journal of Digital Earth*, 1-18. doi:10.1080/17538947.2017.1404150

Bolpagni, M., & Ciribini, A. L. C. (2016). *The Information Modeling and the Progression of Data-Driven Projects.* In: Proceedings of the CIB World Building Congress, Tampere, Finland.

Borrmann, A. (2010). *From GIS to BIM and back again – A Spatial Query Language for 3D building models and 3D city models*. In: Proceedings of the 5th International 3D Geoinfo Conference, BerlinV.

Borrmann, A., Kolbe, H. T., Donaubauer, A., Steuer, H., Jubierre, J. R., & Flurl, M. (2015). Multi-scale geometric semantic modeling of shield tunnels for GIS and BIM applications. *Computer-Aided Civil and Infrastructure Engineering, 30*(4), 263-281.

Boyes, G., Thomson, C., & Ellul, C. (2015). *Integrating BIM and GIS: Exploring the use of IFC space objects and boundaries*. In: Proceedings of the GISRUK 2015, Leeds, UK.

Breunig, M., Kuper, P., Butwilowski, E., Thomsen, A., Jahn, M., Dittrich, A., Al-Doori, M., Golovko, D., & Menninghaus, M. (2016). The story of DB4GeO – A service-based geo-database architecture to support multi-dimensional data analysis and visualisation. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. doi:http://dx.doi.org/10.1016/j.isprsjprs.2015.12.006

Brutto, M. L., & Spera, M. G. (2011). Image-Based and Range-Based 3D modelling of Archaeological Cultural Heritage: The Telamon of the Temple of Olympian Zeus in Agrigento (Italy). *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXVIII-5/W16*.

buidingSMART. (2007). Industry Foundation Classes IFC2x3. http://www.buildingsmarttech.org/specifications/ifc-releases/summary

buidingSMART. (2013). Industry Foundation Classes Specification 4 Add2 - Addendum 2. http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/

buildingSmart. (2017). *Documentation and guidelines: IFC Infra Overall Architecture Project*.

C. Clarke, K., & Chen, J. (2017). *Modeling Standards and File Formats for Indoor Mapping*. Paper presented at the Proceedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management.

Çağdaş, V. (2013). An Application Domain Extension to CityGML for immovable property taxation: A Turkish case study. *International Journal of Applied Earth Observation and Geoinformation, 21*, 545-555. doi:https://doi.org/10.1016/j.jag.2012.07.013

Canada, N. D. o. (2012). Open BIM and the future: BIM and GIS integration - Building the DND Real Property Spatial Data Framework.

Chaturvedi, K. (2014). *Web based 3D analysis and visualisation using HTML5 and WebGL*. (Master), ITC University of Twente, Enschede, The Netherlands.

Chaturvedi, K., Smyth, C. S., Gesquière, G., Kutzner, T., & Kolbe, T. H. (2017). Managing Versions and History Within Semantic 3D City Models for the Next Generation of CityGML. In A. Abdul-Rahman (Ed.), *Advances in 3D Geoinformation* (pp. 191-206). Cham: Springer International Publishing.

Chatuverdi, K., & Kolbe, H. T. (2017). Future City Pilot 1 Engineering Report. *OGC Document 16-098*.

Chen, L.-C., Wu, C.-H., Shen, T.-S., & Chou, C.-C. (2014). The application of geometric network models and building information models in geospatial environments for fire-fighting simulations. *Computers, Environment and Urban Systems, 45*, 1-12. doi:https://doi.org/10.1016/j.compenvurbsys.2014.01.003

Cheng, J., Deng, Y., & Du, Q. (2013). *Mapping Between BIM Models and 3D GIS City Models of Different Levels of Details*. In: Proceedings of the 13th International Conference on Construction Applications of Virtual Reality, London, UK.

CityGML3.0. (2018). Retrieved from https://www.citygml.org/ongoingdev/v3/. Last accessed on 7 June 2018.

Clemen, C., & Gründig, L. (2006). *The Industry Foundation Classes (IFC)–ready for indoor cadastre?* Paper presented at the XXIII International FIG Congress, Munich.

Costa, G., Sicilia, A., Lilis, G., Rovas, D., & Izkara, J. (2016). *A comprehensive ontologies-based framework to support retrofitting design of energy-efficient districts*. In: Proceedings of the European Conference on Product and Process Modelling (ECPPM), Limassol, Cyprus. 7–9 September 2016.

Costamagna, E., & Spanò, A. (2013). CityGML for Architectural Heritage. In A. Abdul Rahman, P. Boguslawski, C. Gold, & M. N. Said (Eds.), *Developments in Multidimensional Spatial Data Models* (pp. 219-237). Berlin, Heidelberg: Springer Berlin Heidelberg.

Dale, L. (2009). Oracle Spatial Extraction, Transform and Load (ETL).

de Laat, R., & van Berlo, L. (2011). Integration of BIM and GIS: The Development of the CityGML GeoBIM Extension. In *Advances in 3D Geo-Information Sciences* (pp. 211-225).

Deng, Y., Cheng, C. P., & Anumba, C. (2016). Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction, 6*(7), 1-21.

Dimyadi, J., Spearpoint, M., & Amor, R. (2008). Sharing Building Information using the IFC Data Model for FDS Fire Simulation. *Fire Safety Science, 9*, 1329-1340. doi:10.3801/iafss.fss.9-1329

Döllner, J., & Hagedorn, B. (2007). *Integrating urban GIS, CAD, and BIM data by service based virtual 3D city models*. London: Taylor & Francis Group.

Donkers, S. (2013). *Automatic generation of CityGML LoD3 building models from IFC models.* (MSc), TUDelft,

Donkers, S., Ledoux, H., Zhao, J., & Stoter, J. (2016). Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. *Transactions in GIS, 20*(4), 547-569. doi:10.1111/tgis.12162

Dore, C., & Mrphy, M. (2012). *Integration of historic building information modeling and 3D GIS for recording and managing cultural heritage sites*. Paper presented at the 18th International Conference on Virtual Systems and Multimedia: "Virtual Systems in the Information Society", Milan, Italy.

Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors* (Second ed.). Hoboken, New Jersey: John Wiley & Sons, Inc.

El-dabiry, T., & Osman, H. (2010). Ontologies for Linking CAD/GIS. In H. A. Karimi & B. Akinci (Eds.), *CAD and GIS Integration* (pp. 171-198): Taylor and Francis (CRC Press).

El-Mekawy, M., Östman, A., & Hijazi, I. (2012). A Unified Building Model for 3D Urban GIS. *ISPRS International Journal of Geo-Information, 1*(2), 120-145. doi:https://doi.org/10.3390/ijgi1020120

El-Mekawy, M., Östman, A., & Shahzad, K. (2011). Towards Interoperating CityGML and IFC Building Models: A Unified Model Based Approach. In *Advances in 3D Geo-Information Sciences* (pp. 73-93).

Elbeltagi, E., & Dawood, M. (2011). Integrated visualized time control system for repetitive construction projects. *Automation in Construction, 20*(7), 940-953.

ESRI. (2006). ArcToolbox Conversion Tools. Retrieved from http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=ArcToolbox_window_basics. Last accessed on 11 June 2018.

ESRI. (2015). ArcSDE. Retrieved from www.esri.com/software/arcgis/arcsde. Last accessed on 11 June 2018.

ESRI. (2017). ArcGIS Data Interoperability Extension Supported Formats. Retrieved from http://www.esri.com/library/fliers/pdfs/data-interop-formats.pdf. Last accessed on.

Floros, G., Solou, D., Pispidikis, I., & Dimopoulou, E. (2016). A roadmap for generating semantically enriched building models according to CityGML model via two different methodologies. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W2*, 23-32. doi:https://doi.org/10.5194/isprs-archives-XLII-2-W2-23-2016

Ford, A. (2004). *The visualisation of integrated 3D petroleum datasets in ArcGIS.* In: Proceedings of the 24th ESRI User Conference, San Diego.

Furukawa, Y., & Hernández, C. (2015). Multi-View Stereo: A Tutorial. *Foundations and Trends® in Computer Graphics and Vision, 9*(1-2), 1-148. doi:10.1561/0600000052

Geiger, A., Benner, J., & Haefele, K. H. (2015). Generalisation of 3D IFC Building Models. In *3D Geoinformation Science* (pp. 19-35).

Github. (2018). Retrieved from https://github.com/opengeospatial/CityGML-3.0. Last accessed on 7 June 2018.

Goetz, M. (2013). Towards generating highly detailed 3D CityGML models from OpenStreetMap. *International Journal of Geographical Information Science, 27*(5), 845-865. doi:10.1080/13658816.2012.721552

Goodchild, M. F. (2011). Scale in GIS: An overview. *Geomorphology, 130*(1-2), 5-9.

Góźdź, K., Pachelski, W., & van Oosterom, P. (2014). *The possibilities of using CityGML for 3D representation of buildings in the Cadastre.* In: Proceedings of the 4th International FIG 3D Cadastre Workshop, Dubai, United Arab Emirates. 9-11 November.

Gröger, G., Benner, J., Dörschlag, D., Drees, R., Gruber, U., Leinemann, K., & Löwner, M. O. (2005). *Das interoperable 3D-Stadtmodell der SIG 3D*. Zeitschrift für Geodäsie, Geoinformation und Landmanagement:

Gröger, G., Kolbe, H. T., Drees, R., Kohlhaas, A., Müller, H., Knospe, F., Gruber, U., & Krause, U. (2004). *Das interoperable 3D-Stadtmodell der SIG 3D der GDI NRW.* Initiative Geodaten Infrastruktur NRW:

Gröger, G., & Plümer, L. (2012). CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing, 71*, 12-33. doi:10.1016/j.isprsjprs.2012.04.004

He, S. (2012). *Production and Visualisation of Levels of Detail for 3D City Models.* (PhD Thesis), École Centrale de Nantes, Nantes, France.

Hijazi, I., Ehlers, M., & Zlatanova, S. (2010). *BIM for Geo- Analysis (BIM4GEOA): Set Up of 3D Information System With Open Source Sofrware and Open Specification (OS).* In: Proceedings of the 5th International 3D GeoInfo Conference, Berlin, Germany.

Hijazi, I., Ehlers, M., Zlatanova, S., Becker, T., & van Berlo, L. A. H. M. (2011). Initial investigations for modeling interior Utilities within 3D Geo Context: Transforming IFC-interior utility to CityGML/ UtilityNetworkADE. In H. T. Kolbe, G. König, & C. Nagel (Eds.), *Advances in 3D Geo-Information Sciences. Lecture Notes in Geoinformation and Cartography* (pp. 95-113). Berlin, Heidelberg: Springer.

Hijazi, I., Ehlers, M., Zlatanova, S., & Isikdag, U. (2009). *IFC to CityGML transformation framework for geo-analysis: a water utility network case*. Paper presented at the 4th International Workshop on 3D Geo-Information, Ghent, Belgium.

Hor, A. H., Jadidi, A., & Sohn, G. (2016). BIM-GIS integrated geospatial information model using semantic WEB and RDF graphs. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 3*, 73-79.

Howell, I., & Batcheler, B. (2005). Building Information Modeling Two Years Later – Huge Potential, Some Success and Several Limitations. Retrieved from http://www.laiserin.com/features/bim/newforma_bim.pdf. Last accessed on 21 May 2018.

Hughes, J., van Dam, A., Mcguire, M., Sklar, F. D., Foley, J., Feiner, K. S., & Akeley, K. (2014). *Computer Graphics: Principles and Practice* (3rd ed.): Addison-Wesley.

IAI. (2005). IFC for GIS (IFG). Retrieved from http://www.iai.no/ifg/Content/ifg_use_cases.htm. Last accessed on 11 June 2018.

Irizarry, J., & Karan, E. P. (2012). Optimising location of tower cranes on construction sites through GIS and BIM integration. *Journal of Information Technology in Construction, 17*, 351-366.

Irizarry, J., Karan, E. P., & Jalaei, F. (2013). Integrating BIM and GIS to improve the visual monitoring of construction supply chain management. *Automation in Construction, 31*, 241-254. doi:https://doi.org/10.1016/j.autcon.2012.12.005

Isikdag, U. (2006). *Towards the implementation of building information models in geospatial context.* (PhD Thesis), University of Salford, Salford, Manchester. Retrieved from http://usir.salford.ac.uk/id/eprint/26731

Isikdag, U., Aouad, G., Underwood, J., & Wu, S. (2007). *Building Information Models: A review on storage and exchange mechanisms.* In: Proceedings of the CIB W78.

Isikdag, U., & Zlatanova, S. (2009a). A SWOT analysis on the implementation of Building Information Models within the geospatial environment. In: Taylor & Francis Group.

Isikdag, U., & Zlatanova, S. (2009b). Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models. In J. Lee & S. Zlatanova (Eds.), *3D Geo-Information Sciences* (pp. 79-96). Berlin, Heidelberg: Springer Berlin Heidelberg.

ISO. (2012). ISO/TS 19152:2012 Geographic information – Land Administration Domain Model.

ISO. (2013). ISO 16739:2013 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.

ISO. (2014). ISO 10303-11:2014 Industrial automation systems and integration - Product data representation and exchange.

ISO. (2016). ISO 29481-1:2016. Building Information Modeling — Information Delivery Manual — Part 1: Methodology and Format.

Johansson, M., & Roupé, M. (2009). *Efficient Real-Time Rendering of Building Information Models* In: Proceedings of the International conference oncomputer graphics and virtual reality (CGVR09), Las Vegas, Nevada, USA.

Julin, A., Jaalama, K., Virtanen, J.-P., Pouke, M., Ylipulli, J., Vaaja, M., Hyyppä, J., & Hyyppä, H. (2018). Characterising 3D City Modeling Projects: Towards a Harmonized Interoperable System. *ISPRS International Journal of Geo-Information, 7*(2). doi:10.3390/ijgi7020055

Jusuf, S., Mousseau, B., Godfroid, G., & Hui, S. J. V. (2017). *Integrated modeling of CityGML and IFC for city/neighborhood development for urban microclimates analysis.* In: Proceedings of the CISBAT 2017 International ConferenceFuture Buildings & Districts – Energy Efficiency from Nano to Urban Scale, Lausanne, Switzerland. 6-8 September 2017.

Jusuf, S., Mousseau, B., Godfroid, G., & Soh, J. (2017). Path to an Integrated Modelling between IFC and CityGML for Neighborhood Scale Modelling. *Urban Science, 1*(3). doi:10.3390/urbansci1030025

Kang, T. W., & Hong, C. H. (2015). A study on software architecture for effective BIM/GIS-based facility management data integration. *Automation in Construction, 54*, 25-38. doi:10.1016/j.autcon.2015.03.019

Karan, E. P., & Irizarry, J. (2015). Extending BIM Interoperability to Preconstruction Operations Using Geospatial Analyses and Semantic Web Services. *Automation in Construction, 53*, 1-12. doi:http://dx.doi.org/10.1016/j.autcon.2015.02.012

Karan, E. P., Irizarry, J., & Haymaker, J. (2015). BIM and GIS integration and interoperability based on semantic web technology. *Journal of Computing in Civil Engineering, 30*(3). doi:04015043

Karimi, H. A., & Akinci, B. (2010). *CAD and GIS Integration* (1st ed.): Taylor and Francis Group (CRC Press).

Kavisha, K., & Saram, S. (2015). *CityGML based interoperability for the transformation of 3D data models.* (Master), Andhra University, India.

Khemlani, L. (2004). The IFC Building Model: A Look Under the Hood. *AECbytes*.

Kim, Y., Kang, H., & Lee, J. (2014). Developing CityGML Indoor ADE to Manage Indoor Facilities. In U. Isikdag (Ed.), *Innovations in 3D Geo-Information Sciences* (pp. 243-265). Cham: Springer International Publishing.

Kolbe, H. T. (2009). Representing and Exchanging 3D City Models with CityGML. In J. Lee & S. Zlatanova (Eds.), *3D Geo-Information Sciences* (pp. 15-31). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kolbe, H. T., & Bacharach, S. (2006). CityGML: An Open Standard for 3D City Models.

Kolbe, H. T., Gröger, G., & Plümer, L. (2008). CityGML–3D city models and their potential for emergency response. *Geospatial information technology for emergency response, 257*.

Kolbe, H. T., Nagel, C., & Stadler, A. (2009). CityGML–OGC standard for photogrammetry. *Photogrammetric Week, 9*, 265-277.

Kossiakoff, A., Sweet, N. W., Seymour, J. S., & Biemer, M. S. (2011). *Systems Engineering Principles and Practice* (S. P. Andrew Ed. 2nd ed.). Hoboken, New Jersey: John Wiley & Sons, Inc. .

Krüger, B. A., & Kolbe, H. T. (2012). *Building Analysis for Urban Energy Planning Using Key Indicators on Virtual 3d City Models - the Energy Atlas of Berlin.* In: Proceedings of the XXII ISPRS Congress, Melbourne, Australia. 25 August – 01 September 2012.

Kühne, T. (2006). Matters of (Meta-) Modeling. *Software & Systems Modeling, 5*(4), 369-385. doi:10.1007/s10270-006-0017-9

Kumar, K., & Saran, S. (2015). *CityGML based Interoperability for the Transformation of 3D Data Models*. Transactions in GIS. Manuscript ID 2015-Mar-TGIS-ORA-1209 (In press).

Kutzner, T., & Kolbe, H. T. (2018a). *CityGML 3.0: Sneak Preview.* In: Proceedings of the Wissenschaftlich-Technische Jahrestagung der DGPF und PFGK.

Kutzner, T., & Kolbe, H. T. (2018b). Notes on the CityGML 3.0 Conceptual Model.

Lamberti, F., Sanna, A., & Ramirez, E. A. H. (2011). *Web-based 3D visualisation for intelligent street lighting*. Paper presented at the Proceedings of the 16th International Conference on 3D Web Technology, Paris, France.

Lancelle, M., & Fellner, W. D. (2010). *Current issues on 3D city models*. Paper presented at the 25th International Conference in Image and Vision Computing, Queenstown, New Zealand.

Lattuada, R. (2006). Three-dimensional representations and data structures in GIS and AEC. In S. Zlatanova & D. Prosperi (Eds.), *Large-scale 3D data integration–Challenges and Opportunities* (pp. 57-86). London: Taylor & Francis.

Lemmens, M. (2011). Geo-information: technologies, applications and the environment. *Springer Science & Business Media, 5*.

Li, L., Wu, J., Zhu, H., Duan, X., & Luo, F. (2016). 3D modeling of the ownership structure of condominium units. *Computers, Environment and Urban Systems, 59*, 50-63. doi:https://doi.org/10.1016/j.compenvurbsys.2016.05.004

Li, M., Xiao, J., Wang, J., Liu, H., Luan, X., & Zhou, Z. (2007). *The design of a map database*. Paper presented at the Geoinformatics 2007: Cartographic Theory and Models.

Li, Y., & He, Z. (2008). *3D Indoor Navigation: a Framework of Combining BIM with 3D GIS*. Paper presented at the 44th ISOCARP Congress 2008, Dalian, China.

Liu, X., Wang, X., Wright, G., Cheng, J., Li, X., & Liu, R. (2017). A State-of-the-Art Review on the Integration of Building Information Modeling (BIM) and Geographic Information System (GIS). *ISPRS International Journal of Geo-Information, 6*(2). doi:10.3390/ijgi6020053

Löwner, M. O., Benner, J., Gröger, G., & Häfele, K.-H. (2013). *New Concepts for Structuring 3D City Models – An Extended Level of Detail Concept for CityGML Buildings*. In: Proceedings of the Computational Science and Its Applications – ICCSA 2013, Berlin, Heidelberg. 2013//.

Löwner, M. O., Gröger, G., Benner, J., Biljecki, F., & Nagel, C. (2016). Proposal for a New Lod and Multi-Representation Concept for Citygml. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W1*, 3-12. doi:10.5194/isprs-annals-IV-2-W1-3-2016

Luan, H., Fan, Y., Zhou, M., & Wang, X. (2014). *Towards Effective 3D Model Management on Hadoop*. In: Proceedings of the Advances in Computer Science and its Applications, Berlin, Heidelberg. 2014//.

Luebke, D., Reddy, M., Cohen, J., Varshney, A., Watson, B., & Huebner, R. (2002). *Level of Detail for 3D Graphics*: Morgan Kaufmann.

Ma, Z., & Ren, Y. (2017). Integrated Application of BIM and GIS: An Overview. *Procedia Engineering, 196*, 1072-1079. doi:10.1016/j.proeng.2017.08.064

Mäntylä, M. (1988). An introduction to solid modelling [Press release]Computer Science Press, New York, USA.

Mao, B., Harrie, L., Cao, J., Wu, Z., & Shenc, J. (2014). NOSQL based 3d city model management system. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-4*.

Mignard, C., Gesquière, G., & Nicolle, C. (2011). *SIGA3D: A Semantic BIM Extension to Represent Urban Environment*. Paper presented at the 5th International Conference on Advances in Semantic Processing, Lisbon, Portugal.

Morton, J. P., Horne, M., Dalton, C. R., & Thompson, M. E. (2012). *Virtual City Models: Avoidance of Obsolescence*. In: Proceedings of the 30th eCAADe Conference, Prague, Czech Republic. September 2012.

Nagel, C. (2014). *Proposal for a revision of the CityGML LOD concept*. Paper presented at the 5th Meeting of OGC Working Package 3 for the revision of the LoD concept for CityGML 3.0.

Nagel, C., Stadler, A., & Kolbe, H. T. (2009). Conceptual Requirements for the Automatic Reconstruction of Building Information Models from Uninterpreted 3D Models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 34*.

OGC. (2007). OGC Web Services Architecture for CAD GIS and BIM.

OGC. (2010a). Draft for Candidate OpenGIS: Web 3D Service Interface Standard. http://www.w3ds.org/doku.php

OGC. (2010b). OGC Web View Service: Discussion Paper. http://www.webviewservice.org/doku.php

OGC. (2012a). OGC 3D Portrayal Interoperability Experiment.

OGC. (2012b). OGC City Geography Markup Language (CityGML) Encoding Standard. Version 2.0.0. 12-019. Retrieved from http://www.opengis.net/spec/citygml/2.0

OGC. (2017). 3D Portrayal Service 1.0. . http://docs.opengeospatial.org/is/15-001r4/15-001r4.html#_3dps_service_model

OGC. (2018). OGC IndoorGML. Version 1.0.3. 14-005r4. Retrieved from http://docs.opengeospatial.org/is/14-005r5/14-005r5.html

Park, T., Kang, T. W., Lee, Y., & Seo, K. (2014). *Project Cost Estimation of National Road in Preliminary Feasibility Stage Using BIM/GIS Platform*. In: Proceedings of the 2014 International Conference on Computing in Civil and Building Engineering, Orlando, Florida, United States. 23-25 June 2014.

Peachavanish, R., Karimi, H. A., Akinci, B., & Boukamp, F. (2006). An ontological engineering approach for integrating CAD and GIS in support of infrastructure management. *Advanced Engineering Informatics, 20*, 71-88.

Pfund, M. (2002). 3D GIS Architecture A Topological Data Structure. *GIM International, 16*, 35-37.

Pispidikis, I., Tsiliakou, E., Kitsakis, D., Athanasiou, K., Kalogianni, E., Labropoulos, T., & Dimopoulou, E. (2018). *Combining Methodological Tools for the optimum 3D Modelling of NTUA Campus*. National Technical University of Athens.

Pouliot, J., Roy, T., Fouquet-Asselin, G., & Desgroseilliers, J. (2010). 3D Cadastre in the province of Quebec: a first experiment for the construction of a volumetric representation. In H. T. Kolbe, Gerhard, C. Nagel, & Claus (Eds.), *Advances in 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography* (pp. 149-162). Berlin, Germany: Springer.

PracticalBIM. (2013). Practical tips on making BIM work. Retrieved from http://practicalbim.blogspot.com/2013/03/what-is-this-thing-called-lod.html. Last accessed on 10 June 2018.

Pu, S., & Zlatanova, S. (2006). *Integration of GIS and CAD at DBMS level*. Paper presented at the UDMS '06 25th Urban Data Management Symposium, Delft, Netherlands.

Rafiee, A., Dias, E., Fruijtier, S., & Scholten, H. (2014). From BIM to geo-analysis: View coverage and shadow analysis by BIM/GIS integration. *Procedia Environ.mental Sciences, 22*, 397–402.

Reinhardt, J., & Bedrick, J. (2016). Level of Development Specification. *BIMForum*.

Reitz, T., & Schubiger-Banz, S. (2014). The Esri 3D city information model. *IOP Conference Series: Earth and Environmental Science, 18*(1).

Requicha, A. A. G. (1982). Representation of rigid solids—theory, methods and systems. *ACM Computing Surveys, 12*(4), 437-464.

Saygi, G., Aguriaro, G., Hamamcioglu-Turan, M., & Remondino, F. (2013). Evaluation of GIS and BIM roles for the information management of historical buildings. *ISPRS Ann. Photogrammetry Remote Sensing Spatial Information Sciences, 2*, 283-288.

Schneck, D., & Wilson, P. (1994). Information Modeling the EXPRESS Way. *Oxford University Press*.

Shahzad, M., & Zhu, X. X. (2015). Robust Reconstruction of Building Facades for Large Areas Using Spaceborne TomoSAR Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing, 53*(2), 752-769. doi:10.1109/TGRS.2014.2327391

Singh, S. P., Jain, K., & Mandla, R. (2013). Virtual 3D City Modelling: Techniques and Applications. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-2/W2*.

Software, S. (2016). FME. Retrieved from http://www.safe.com/. Last accessed on 1 June 2018.

Song, Y., Wang, X., Tan, Y., Wu, P., Sutrisna, M., Cheng, J., & Hampson, K. (2017). Trends and Opportunities of BIM-GIS Integration in the Architecture, Engineering and Construction Industry: A Review from a Spatio-Temporal Statistical Perspective. *ISPRS International Journal of Geo-Information, 6*(12). doi:10.3390/ijgi6120397

Stadler, A., & Kolbe, H. T. (2007). *Spatio-Semantic Coherence in the Integration of 3D City Models*. In: Proceedings of the 5th International ISPRS Symposium on Spatial Data Quality ISSDQ, Enschede, The Netherlands.

Stadler, A., Nagel, C., König, G., & Kolbe, H. T. (2008). Making interoperability persistent: A 3D geo database based on CityGML. In J. Lee & S. Zlatanova (Eds.), *3D Geo-Information Sciences, Selected papers from the 3rd International Workshop on 3D Geo-Information* (pp. 175-192). Seoul, Korea: LNG&C series, Springer Verlag.

Stoter, J., Ledoux, H., Zlatanova, S., & Biljecki, F. (2016). *Towards sustainable and clean 3D Geoinformation*.

Sugumaran, R., Burnett, J., & Armstrong, M. (2014). Using a Cloud Computing Environment to Process Large 3D Spatial Datasets. In *Big Data: Techniques and Technologies in Geoinformatics*.

Suveg, I., & Vosselman, G. (2004). Reconstruction of 3D building models from aerial images and maps. *ISPRS Journal of Photogrammetry and Remote Sensing, 58*(3), 202-224. doi:https://doi.org/10.1016/j.isprsjprs.2003.09.006

Tah, J. H. M., Oti, A. H., & Abanda, F. H. (2017). A state-of-the-art review of built environment information modelling (BeIM). *Organisation, Technology and Management in Construction: an International Journal, 9*(1). doi:10.1515/otmcj-2016-0030

Tashakkori, H., Rajabifard, A., & Kalantari, M. (2015). A new 3D indoor/outdoor spatial model for indoor emergency response facilitation. *Building and Environment, 89*, 170-182. doi:https://doi.org/10.1016/j.buildenv.2015.02.036

Thompson, M., Horne, M., Lockley, S., & Cerny, M. (2011). *Towards an information rich 3D city model: Virtual NewcastleGateshead GIS integration*. In: Proceedings of the 12th International Conference on Computers in Urban Planning and Urban Management, Alberta, Canada.

Tomlinson, R. F. (1969). A Geographic Information System for Regional Planning. *Journal of Geography (Chigaku Zasshi), 78*(1), 45-48. doi:10.5026/jgeography.78.45

Tomljenovic, I., Höfle, B., Tiede, D., & Blaschke, T. (2015). Building Extraction from Airborne Laser Scanning Data: An Analysis of the State of the Art. *Remote Sensing, 7*(4), 3826-3862.

Tsiliakou, E. (2013). *Κανονιστική μοντελοποίηση στο 3D Κτηματολόγιο – Εφαρμογή στην Πολυτεχνειούπολη Ζωγράφου*. National Technical University of Athens,

TUDelft. (2018). 3D Geoinformation Group. azul - CityGML viewer for Mac. Retrieved from https://github.com/tudelft3d/azul Last accessed on.

UNIDO. (2006). *Role of Standards. A Guide for Small and Medium-sized Enterprises*. Retrieved from https://www.unido.org/role-standards-guide-small-and-medium-sized-enterprises. Last accessed on 21 May 2018.

van Berlo, L. A. H. M., & Bomhof, F. (2014). *Creating the Dutch national BIM levels of development.* In: Proceedings of the 2014 International Conference on Computing in Civil and Building Engineering.

van Berlo, L. A. H. M., & de Laat, R. (2011). *Integration of BIM and GIS: The development of the CityGML GeoBIM extension.* In: Proceedings of the 5th International 3D GeoInfo Conference, Berlin, Germany,. 3–4 November 2010.

van Oosterom, P. (2013). Research and development in 3D cadastres. *Computers, Environment and Urban Systems, 40*, 1-6. doi:https://doi.org/10.1016/j.compenvurbsys.2013.01.002

van Oosterom, P. J. M., & Stoter, J. (2006). Bridging the Worlds of CAD and GIS. In S. Zlatanova & D. Prosperi (Eds.), *Large-Scale 3D Data Integration: Challenges and Opportunities* (pp. 9-36). Boca Raton, FL: CRC Press, Taylor & Francis Group.

Vilgertshofer, S. (2016). *IfcTunnel – A proposal for a multi-scale extension of the IFC data model for shield tunnels under consideration of downward compatibility aspects*. Paper presented at the 11th European Conference on Product and Process Modelling, Limassol, Cyprus.

Vilgertshofer, S., Amann, J., Willenborg, B., Borrmann, A., & Kolbe, T. H. (2017). *Linking BIM and GIS Models in Infrastructure by Example of IFC and CityGML*. Paper presented at the Computing in Civil Engineering 2017.

Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings — Literature review and future needs. *Automation in Construction, 38*, 109-127. doi:10.1016/j.autcon.2013.10.023

W3C. (2014). RDF 1.1 specifications. https://www.w3.org/standards/techs/rdf#w3c_all

Wang, W., & Wang, K. (1986). Geometric modeling for swept volume of moving solids. *Computer Graphics and Applications, IEEE, 6*(12), 8-17.

Wu, I. C., & Hsieh, S. H. (2007). Transformation From IFC Data Model to GML Data Model: Methodology and Tool Development. *Journal of Chinese Institute of Engineers, 30*(6), 1085-1090.

Wu, W., Yang, X., & Fan, Q. (2014). *GIS-BIM Based Virtual Facility Energy Assessment (VFEA) – Framework Development and Use Case of California State University, Fresno.* In:

Proceedings of the Computing in Civil and Building Engineering, Orlando, Florida, United States.

Xie, X., Xu, K., Mitra, N. J., Cohen, O. D., Gong, W., Su, Q., & Chen, B. (2013). Sketch-to-design: Context-based part assembly. *Computer Graphics Forum 32, 32*(8), 233-245.

Yang, B., & Lee, J. (2017). Improving accuracy of automated 3-D building models for smart cities. *International Journal of Digital Earth*, 1-19. doi:10.1080/17538947.2017.1395089

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., & Kolbe, H. T. (2018). 3DCityDB - a 3D geodatabase solution for the management, analysis, and visualisation of semantic 3D city models based on CityGML. *Open Geospatial Data, Software and Standards, 3*(5). doi:https://doi.org/10.1186/s40965-018-0046-7

Yin, X., Wonka, P., & Razdan, A. (2009). Generating 3D Building Models from Architectural Drawings: A Survey. *IEEE Computer Graphics and Applications, 29*(1), 20-30. doi:10.1109/MCG.2009.9

Zhu, J., Wright, G., Wang, J., & Wang, X. (2018). A Critical Review of the Integration of Geographic Information System and Building Information Modelling at the Data Level. *ISPRS International Journal of Geo-Information, 7*(2). doi:10.3390/ijgi7020066

Zlatanova, S., & Holweg, D. (2004). *3D Geo-information in emergency response: a framework.* In: Proceedings of the 4th International Symposium on Mobile Mapping Technology: a framework. , Kunming, China. 29-31 March.

Zlatanova, S., Stoter, J., & Isikdag, U. (2012). *Standards for Exchange and Storage of 3D Information: Challenges and Opportunities for Emergency Response.* In: Proceedings of the 4th International Conference on Cartography & GIS, Albena.