

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Architettura Costruzione Città
Anno Accademico 2018/2019



TESI DI LAUREA MAGISTRALE

Modellazione parametrica e metodologia BIM
per la progettazione partecipata di una camera di degenza ospedaliera

Relatore:

Prof.ssa Anna Osello

Correlatore:

Ing. Pablo Angel Ruffino

Candidato:

Linda Falsini

Settembre 2019

“

*Faccio sempre ciò che non so fare,
per imparare come va fatto.*

Vincent Van Gogh

”

Ringraziamenti

Quest'ultimo anno accademico è stato carico di sfide ed emozioni, ma anche di grandi soddisfazioni personali. Per questo ci tengo ad esprimere la mia riconoscenza verso coloro che ne hanno fatto parte e che mi sono stati vicino.

Prima di tutto vorrei ringraziare la mia relatrice, la professoressa Anna Osello, per avermi introdotta alla disciplina del BIM ed avermi dato la possibilità di approfondire il mio interesse in materia attraverso questa tesi. Ringrazio Pablo Ruffino per avermi guidata in questa avventura contagiandomi col suo entusiasmo ed insegnandomi che «anche un fallimento, per la ricerca, è un buon risultato». Sono riconoscente a tutti i ricercatori e ai dottorandi del *DrawingTOthefuture*, che mi hanno accolta nel laboratorio, ma anche ai tesisti che hanno condiviso questa esperienza con me.

Il ringraziamento più grande va alla mia famiglia, che mi ha sempre sostenuta credendo in me e senza la quale non avrei potuto raggiungere questo importante traguardo. In particolare sono grata ai miei genitori, da cui ho imparato a rialzarmi a testa alta dopo ogni difficoltà e a non lasciarmi ostacolare dalle altrui opinioni negative. Ringrazio mio fratello Daniele per avermi fatto da personale supporto tecnico (gratuito!) ed essermi sempre stato accanto confortandomi con i suoi impareggiabili discorsi di incoraggiamento, anche nei momenti per me più bui.

Grazie a tutti quelli che hanno riposto fiducia in me, ma anche a quelli che hanno pensato, o sperato, che io non potessi farcela. Mi avete tutti resa più forte.

Abstract



The recent legislative impositions regarding the digitization of public works, together with the need to optimize resources to reduce production and maintenance costs and make the necessary health facilities available to the area as quickly as possible, solicit the use of the Building Information Modeling (BIM) as a methodology for the design, construction and management of such works. The aim of this research was to investigate the possibility of designing parametric rooms, usable on multiple BIM platforms, trying to accelerate and automate the process, based on the principle of repetition of the functions that distinguishes hospitals. In particular, methodological approaches about the creation of patient rooms were tested, so that they can be modified both in the geometric and informative aspects, in order to obtain, from a single parameterized object, an unlimited number of its varying possibilities, adaptable according to the needs of the project in which they have to be integrated. Thanks to the achievement of a single-function BIM model, the approach to participatory design, aimed at involving actors in the process, is simpler and allows the project to be modified in its preliminary stages, up to the agreement of all the parts, before the adaptations take place in the subsequent more critical phases of the process itself. This implies that the modeled rooms can be tested individually, even before their contextualization within the model of the healthcare facility for which they are intended, resulting in an improvement of time and resources required for the design.

Abstract

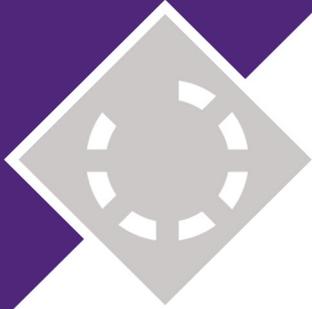


Le recenti imposizioni legislative in merito alla digitalizzazione delle opere pubbliche, unitamente alla necessità di ottimizzare le risorse per ridurre i costi di produzione e manutenzione e rendere disponibili, nel minor tempo possibile, le strutture sanitarie necessarie al territorio, sollecitano l'utilizzo del *Building Information Modeling* (BIM) come metodologia per la progettazione, realizzazione e gestione di tali opere. L'obiettivo di questa ricerca è investigare sulla possibilità di progettare delle stanze parametriche, impiegabili su più piattaforme BIM, cercando di velocizzarne ed automatizzarne il processo, basandosi sul principio di ripetitività delle funzioni che contraddistingue i complessi ospedalieri. In particolare vengono testati degli approcci metodologici incentrati sulla realizzazione di camere di degenza, editabili sia nell'aspetto geometrico che informativo, in modo da ricavare, da un unico oggetto parametrizzato, un numero illimitato di sue possibili varianti, adattabili a seconda delle esigenze del progetto in cui devono essere inserite. Grazie all'ottenimento di un modello BIM monofunzionale è più semplice l'approccio alla progettazione partecipata, volta al coinvolgimento di molteplici attori all'interno del processo, che permette di modificare il progetto nelle sue fasi preliminari, fino all'accordo di tutte le parti, prima che gli adattamenti abbiano luogo nelle successive fasi più critiche del processo stesso. Questo implica che le stanze modellate possano essere testate singolarmente, anche prima della loro contestualizzazione all'interno del modello della struttura sanitaria a cui sono destinate, determinando un efficientamento dei tempi e delle risorse richiesti per la progettazione.

Indice

1 Introduzione	13
1.1 BIM: <i>Building Information Modeling</i>	15
1.1.1 Tradizione VS innovazione	17
1.1.2 Caratteristiche del BIM	19
1.1.3 Affidabilità del dato	22
1.2 Progettazione parametrica	25
1.2.1 Modellazione algoritmica	26
1.2.2 <i>Visual editing</i> e BIM	28
1.3 Applicazione del BIM in ambito sanitario	31
1.3.1 Caso studio: parametrizzazione di una degenza nel Parco della Salute	33
1.3.2 Vincoli e dati progettuali	37
1.4 Progettazione partecipata	39
1.4.1 Tecnologie per lo sviluppo della progettazione partecipata nel settore edilizio: realtà virtuale, aumentata e mista	42
2 Metodologia	49
2.1 Fase 1: Test d'interoperabilità di un modello concettuale	53
2.1.1 Scelta della piattaforma BIM: <i>Revit</i> VS <i>ArchiCAD</i>	54
2.1.2 Parametrizzazione di un "oggetto-stanza" con <i>Revit</i>	61
2.1.3 Interoperabilità orizzontale diretta	69
2.1.4 Software parametrico "intermediario": <i>Grasshopper</i>	83
2.1.5 Parametrizzazione ed interoperabilità orizzontale indiretta	86
2.2 Fase 2: Generazione del modello parametrico di una degenza	99
2.2.1 Definizione dell'algoritmo e dei vincoli progettuali	100
2.2.2 Dallo <i>script</i> al BIM: revisione del modello	118
2.2.3 Dallo <i>script</i> al BIM: opzioni di visualizzazione per la creazione di un <i>template</i>	122
2.3 Fase 3: Approccio alla progettazione partecipata	129
2.3.1 Gestione delle librerie	130
2.3.2 Requisiti per la progettazione partecipata	133
2.3.3 Progettazione partecipata: <i>BIMx</i>	136

3 Risultati	143
3.1 Fase 1: La ricerca dell'approccio metodologico	145
3.2 Fase 2: Il paradosso dell'approccio parametrico	151
3.3 Fase 3: Un approccio olistico alla progettazione	155
4 Conclusione e sviluppi futuri	159
5 Allegati digitali	165
Riferimenti: bibliografia e sitografia	169



1

Introduzione

1 Introduzione

Negli ultimi anni l'industria delle costruzioni sta affrontando un processo di rinnovamento che implica il cambiamento dell'intero settore edilizio. Le costanti innovazioni tecnologiche si sono rese complici di tale metamorfosi, collaborando alla transizione fra tradizione e innovazione. In questo contesto sta prendendo sempre più importanza il **Building Information Modeling (BIM)**, grazie al quale è possibile dirigersi verso un efficientamento della progettazione e della gestione dell'intero ciclo di vita di edifici ed infrastrutture attraverso la digitalizzazione dei dati [1]. Questa metodologia sta operando non solo una rivoluzione nel campo dell'edilizia, specie nel settore pubblico, dove in Italia è già diventata obbligatoria dal 2019 per opere di importo pari o superiore a 100 milioni di euro [2], ma anche un **cambiamento culturale** volto a modificare modalità, attrezzature e tecnologie degli attori coinvolti.

Benché l'acronimo "BIM" sia stato coniato alla fine degli anni '70 da **Chuck Eastman**, professore presso il *Georgia Institute of Technology*, il concetto che ne sta alla base risale ad un periodo ancor precedente. È stato infatti **Douglas Carl Engelbart** (1925 - 2013), inventore statunitense nonché uno dei pionieri nell'interazione uomo-computer, a concepire, all'interno del suo articolo "*Augmenting Human Intellect*" del 1962, l'idea che la rappresentazione architettonica dovesse avere alla base una serie di dati ed informazioni, prima che una descrizione geometrica:

«Ignoring the representation on the display, the architect next begins to enter a series of specifications and data – a six-inch slab floor, twelve-inch concrete walls eight feet high within the excavation, and so on. When he has finished, the revised scene appears on the screen. A structure is taking shape. He examines it, adjusts it [...]. These lists grow into an evermore-detailed, interlinked structure, which represents the maturing thought behind the actual design.» [3]

Fa quindi riflettere come oggi, dopo quasi sessant'anni, questo sia ancora un argomento delicato e oggetto di continue polemiche ed osti-

lità all'interno del settore delle costruzioni.

Infatti, nonostante l'adozione del BIM sia destinata ad essere una scelta obbligata, che prevede il suo pieno utilizzo, per le opere pubbliche, a partire dal 2025 in Italia [2], la sfida più grande non è solo tecnologica, ma riguarda in primo luogo l'essere umano, e la capacità di mettersi in gioco lasciando da parte il metodo certo e consolidato per qualcosa di nuovo ed ignoto e che, in quanto tale, spaventa. L'obiettivo è perciò quello di realizzare un domani diverso dall'oggi non solo cambiando gli strumenti tecnologici ad oggi a disposizione, ma principalmente adottando un "**BIM state of mind**" che porti all'elaborazione dei concetti di collaborazione, coordinamento, condivisione e confronto [4].

In questo contesto, la presente tesi applica la metodologia BIM con uno sguardo più ristretto, volgendosi verso la progettazione di un singolo elemento in **ambito sanitario** che, parametrizzato ed informatizzato, possa automatizzare un processo che, se pensato con metodologia tradizionale, sarebbe invece meno efficiente, con maggiori costi e probabilità di errori e ripetitivi tempi morti. Concentrandosi pertanto solo su una piccola porzione del processo edilizio si desidera arrivare, a piccoli passi, ad un grande risultato di integrazione e collaborazione verso la creazione di un più approfondito e complesso modello informativo.

1.1 BIM: *Building Information Modeling*

Negli ultimi anni il BIM è diventato oggetto di studio da parte di differenti discipline all'interno dell'industria delle costruzioni e, pertanto, ha sviluppato innumerevoli definizioni e sfumature. Esso rappresenta un'attività innovativa che offre la possibilità di attuare una svolta non solo a livello progettuale, ma anche, e soprattutto, metodologico mediante una serie di azioni che coinvolgono tutti gli attori del processo edilizio, a più livelli e in diversi tempi, al fine di **garantire qualità ed efficienza** nell'arco dell'intero ciclo di vita di un manufatto [5].

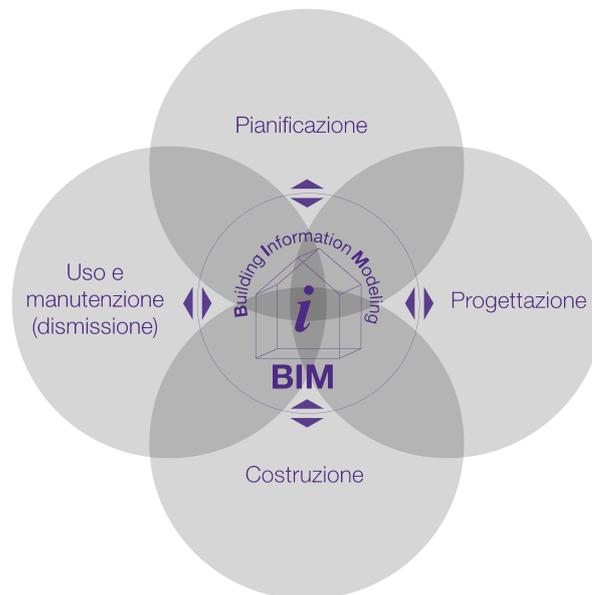
L'acronimo con cui il BIM verrà analizzato nell'ambito di questa ricerca è quello che lo identifica come "*Building Information Modeling*". Tra le tante accezioni, questa è di fatto quella consolidata da Chuck Eastman, che è considerato una delle pietre miliari del BIM. Dopo aver aperto un nuovo scenario nel mondo delle costruzioni, grazie all'introduzione, nel 1974, del "sistema descrittivo dell'edificio" [6], che ancora rimaneva accentrato sull'aspetto grafico, egli ha cominciato a volgere le proprie ricerche verso il possibile scambio di dati tra i modelli. Questo lo ha portato alla definizione di BIM come uno degli sviluppi più promettenti del settore edilizio, in grado di fornire un **modello virtuale** contenente sia le geometrie che le informazioni necessarie a tutte le attività che contraddistinguono la realizzazione di un edificio [7].

Analizzare pertanto il BIM esclusivamente come modello tridimensionale di un manufatto, per quanto possa comunque essere utile per la progettazione, non può considerarsi pienamente completo. Il secondo aspetto che necessariamente lo deve caratterizzare è quello dell'**integrazione di informazioni** e proprietà agli elementi grafici che compongono il modello stesso [8]. Tali dati possono spaziare da elementi specifici riguardo alle tecniche costruttive a dati di tipo economico, temporale o gestionale [9]. I modelli BIM non contengono perciò le sole nozioni architettoniche, ma l'insieme di tutte le informazioni sull'edificio, includendo fra queste anche quelle attinenti i differenti settori ingegneristici, tra cui gli studi strutturali, impiantistici, o anche relativi alla sostenibilità. Un modello BIM può quindi essere considerato come la rappresentazione, mediante dati, delle diverse di-

scipline che costituiscono un edificio.

È fondamentale inoltre enfatizzare come, partendo da un'idea di BIM basata su un utilizzo “verticale” del processo integrato, che prevedeva l'uso del modello per le sole fasi di progettazione e costruzione dell'edificio, si stia cercando sempre più di evolversi verso una soluzione “orizzontale”, che veda l'impiego della metodologia BIM anche nelle fasi di gestione e manutenzione, ma anche demolizione e dismissione. Per arrivare a tale risultato di applicazione del processo all'**intero ciclo di vita** dell'edificio è essenziale indirizzare l'integrazione delle informazioni del modello verso l'ottimizzazione dell'interscambio dei dati grafici e funzionali che supportino le operazioni [10]. La novità non consiste quindi solo nell'inserimento di un insieme di informazioni, ma nel modo in cui queste vengono gestite nell'arco dell'intero processo.

1 - Gestione dell'intero ciclo di vita del manufatto, grazie al BIM [4]



A porre l'accento sull'importante ruolo che il BIM può ricoprire grazie al controllo della totalità delle azioni che costituiscono un manufatto edilizio non sono solo istituzioni ed illustri professionisti del settore, ma anche le stesse *software house* che promuovono i propri programmi BIM. Fra queste emerge l'osservazione di *Graphisoft* che evidenzia inoltre la potenzialità di “previsione” spiegando che:

«un vero modello BIM deve comprendere l'equivalente virtuale delle parti dell'edificio e degli elementi necessari alla sua costruzione. Questi elementi possiedono tutte le caratteristiche - fisiche e logiche - delle loro controparti reali, quindi si tratta di elementi intelligenti che costituiscono il prototipo digitale degli elementi fisici (muri, pilastri, porte, finestre, scale, ecc.) che ci permettono di simulare l'edificio ed il suo comportamento al computer, prima di iniziarne la costruzione.» [11]

È di fatto questo il motivo principale per cui, secondo Chuck Eastman, come afferma in occasione della *Lectio Magistralis* tenuta il 27 aprile 2017 presso l'Università di Pisa [12], il BIM si sia così ampiamente diffuso nei paesi del nord Europa. Nei paesi scandinavi, dove i

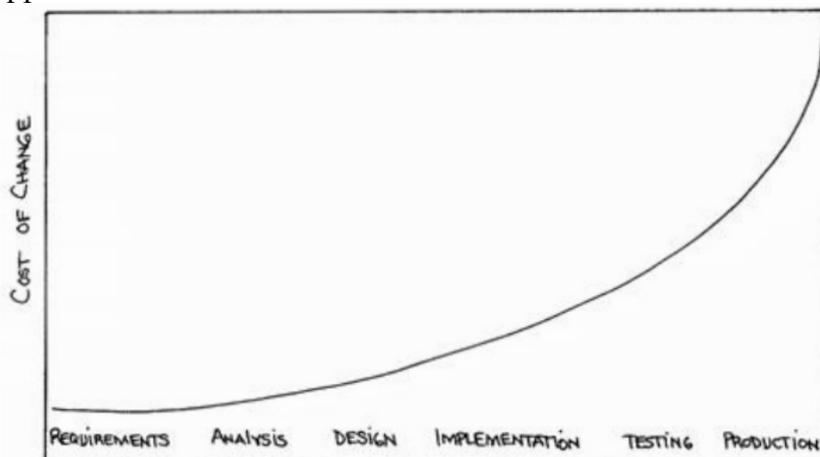
prolungati periodi di buio costituiscono un notevole ostacolo alle fasi di costruzione, il BIM si è presentato quasi come un'inevitabile necessità per **prevedere ed evitare eventuali errori** in fase di cantiere, che avrebbero invece costituito impiego di maggiori costi, tempi e risorse.

Il vero cambiamento, con l'impiego di questa nuova metodologia, consiste nel fatto che il processo non è più focalizzato su linee vettoriali, forme e caselle di testo, benché ancora in grado di produrre disegni 2D, ma su pacchetti di dati che, descrivendo gli oggetti virtuali, imitano il modo in cui essi verranno gestiti nel mondo reale. Si può pertanto affermare che il BIM sia un considerevole strumento per l'aumento della produttività del settore edilizio.

1.1.1 Tradizione VS innovazione

Il BIM si impone come una **rivoluzione metodologica** rispetto alla pratica tradizionale che da sempre prevede una mancanza di coordinazione sia tra la gestione delle informazioni che tra gli attori coinvolti nel processo. Per fare ciò, fin dalle fasi preliminari del processo edilizio, i professionisti devono partecipare alla condivisione dei dati, abbandonando la consueta circospezione e "gelosia" delle informazioni, in modo che l'intero processo possa basare in modo efficiente le proprie metodologie esclusivamente sugli obiettivi finali, senza inutili sprechi di risorse.

In questo ambito non può non essere citato Patrick MacLeamy, architetto statunitense e presidente di *BuildingSMART*, che nel 2004 ha rappresentato graficamente la difficoltà a cui un progetto va incontro con l'avanzamento delle sue fasi, ponendo a confronto la metodologia tradizionale e quella che invece prevede l'uso de BIM. Tale diagramma, che prende il nome dal suo ideatore, deriva dalla ben precedente **curva di Boehm** (1976) [13], che mostrava l'esponenziale crescita del costo del cambiamento del progetto, intendendolo in termini sia di sforzo che di sfruttamento di risorse, al progredire del suo stesso sviluppo.



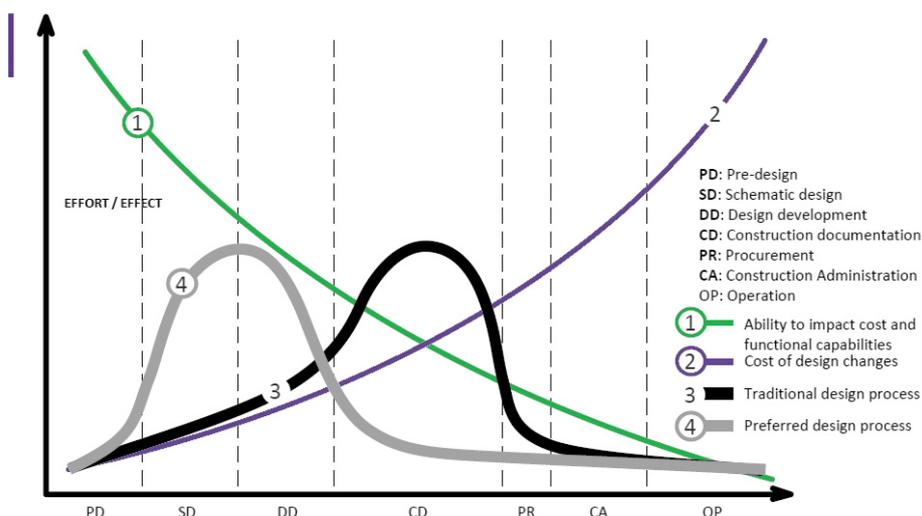
2 - Curva di Boehm [13]

La **curva di MacLeamy**, somigliante per forma e significato a quella di Boehm, evidenzia ancora di più la necessità di flessibili-

tà nell'arco di tutte le fasi progettuali e l'importanza di massimizzare l'impiego di risorse ed energie in quelle iniziali. Questo diagramma è di fatto esplicativo di quanto sia innovativa la metodologia BIM grazie all'accostamento diretto di quattro funzioni che si sviluppano lungo l'asse dello sforzo (ordinata) e quello delle fasi del processo (ascissa). Oltre alle due curve rappresentative della pratica tradizionale e di quella preferenziale del processo BIM, e a quella dei costi di cambiamento del progetto, derivante dalle osservazioni di Barry Boehm, è di fatto descritta anche una curva, quasi speculare alla precedente, che indica la capacità di influenzare costi e prestazioni.

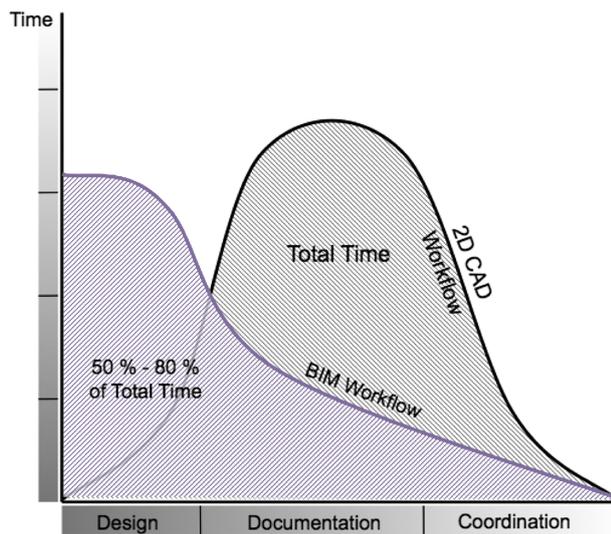
Confrontando la posizione delle due campane si può osservare come quella del CAD sia traslata rispetto a quella del BIM, evidenziando come il flusso di lavoro tradizionale abbia il suo massimo sforzo nel momento più critico, dove i costi di progetto cominciano ad incrementare esponenzialmente. Inoltre questo avviene nella fase di costruzione e documentazione, contrariamente al massimo utilizzo di risorse del BIM, che si trova già alla fine della fase di progettazione concettuale. Questo evidenzia non solo che le modifiche di progettazione risultano più facili, efficaci e meno costose nelle prime fasi del processo, ma anche che con la nuova metodologia si ha fin da subito grande capacità di influenzare i costi complessivi.

3 - Diagramma di MacLeamy [4]



Si può ben notare come, mettendo le due metodologie a confronto in funzione del tempo, lo sforzo massimo sia di differente intensità e anticipato dalla fase di documentazione a quella di progetto. Con il BIM infatti si ha un **costo extra in risorse iniziale**, che, come dimostra il grafico sottostante, permette però di avere un successivo **risparmio del tempo totale** pari a circa il 50-80 % rispetto alla pratica CAD tradizionale. Il primo richiede infatti l'impiego di più tempo nelle fasi di pianificazione sia del processo che della condivisione, mentre il secondo flusso di lavoro si impenna drasticamente nella fase di documentazione e reperimento dei dati, gravando anche sulla condivisione di informazioni ed elaborati. Non si tratta di un effettivo dato quantitativo, essendo lo sforzo progettuale comunque proporzionato al tipo di progetto, ma si intende dimostrare l'**abilità di anticipare** tali sforzi, in aggiunta ai vantaggi che ne derivano. Questi com-

portano infatti più libertà decisionale nelle fasi iniziali, in quanto è possibile, partendo dallo stesso modello, estrapolare differenti varianti, ma anche un notevole risparmio di tempo in tutte le fasi del processo successive alla progettazione. Maggiore è l'entità del progetto, più grandi saranno i vantaggi a lungo termine. Risulta quindi evidente la potenzialità di ottimizzazione del processo che si può ottenere adottando la metodologia BIM, grazie alla quale, massimizzando lo sforzo iniziale, è possibile velocizzare le fasi di sviluppo del progetto, documentazione, costruzione ed esercizio.



4 - Interpretazione del diagramma di MacLeamy in funzione del tempo [11]

1.1.2 Caratteristiche del BIM

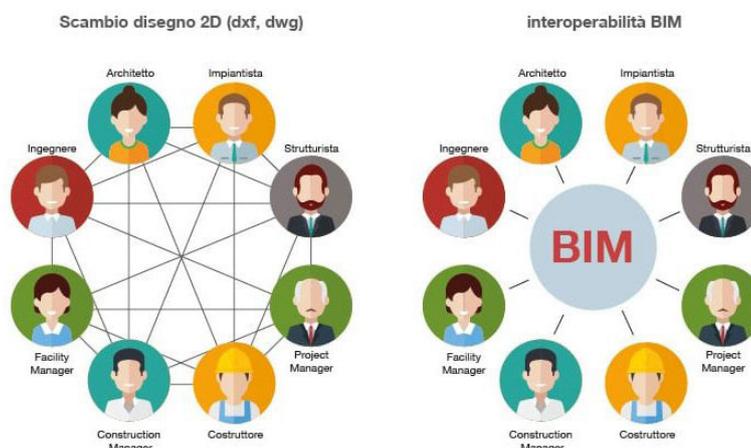
I grafici presentati nel capitolo precedente evidenziano la necessità di collaborazione e di ottimizzazione nella gestione delle informazioni che contraddistinguono l'approccio integrato del BIM. Si è già precedentemente discusso sull'importanza della creazione di **banche dati** al fine di garantire qualità ed efficienza nell'arco dell'intero ciclo di vita di un manufatto. Nel capitolo successivo questa caratteristica verrà ulteriormente approfondita in termini di affidabilità.

Si pone quindi ora l'attenzione su una delle più dirette conseguenze di questa tipologia di approccio, ovvero la modalità con cui gestire tali informazioni. Mediante la metodologia BIM si presuppone, infatti, che i professionisti condividano tra loro i dati immessi con il proprio apporto progettuale attraverso un **modello condiviso**. In questo modo ognuno di essi collabora all'aggiornamento e all'arricchimento delle informazioni digitali che compongono il processo edilizio stesso. Il conseguente vantaggio che ne deriva è la possibilità di evitare la ridondanza delle informazioni e riducendo, quindi, gli errori che inciderebbero sia in termini economici che temporali. A questo si aggiunge anche l'onere, ma soprattutto il beneficio, di avere un modello aggiornato in tempo reale durante la totalità delle fasi progettuali [14].

Per raggiungere questo scopo è fondamentale il ruolo ricoperto dall'**interoperabilità**, ovvero la capacità di trasferire informazioni

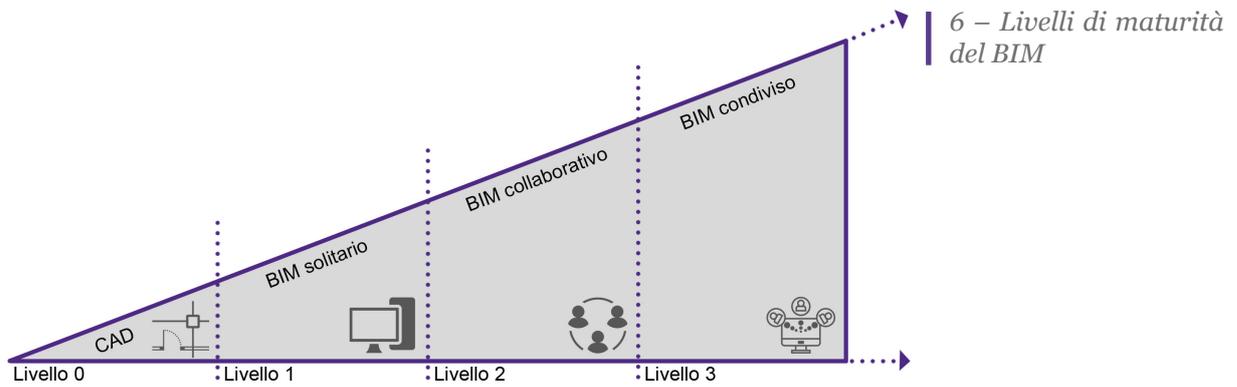
tra software per agevolare il flusso di lavoro e la sua potenziale automatizzazione [7]. Bisogna infatti sottolineare che non esiste un programma che possa gestire tutti gli *output* delle differenti discipline che partecipano all'intero processo progettuale. Per questo motivo, senza l'interoperabilità non ci sarebbe integrazione di informazioni né alcun tipo di collaborazione tra professionisti.

5 - Confronto tra il flusso di lavoro tradizionale e quello d'interoperabilità del Building Information Modeling [15]

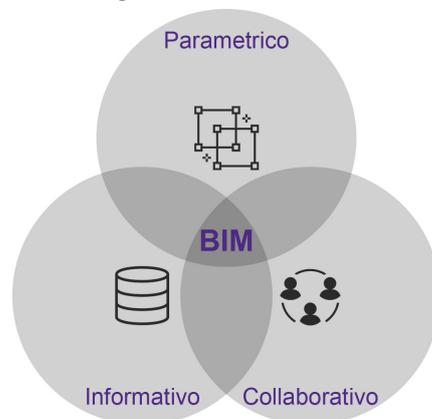


Ogni software ha la sua struttura interna, pertanto sono tuttora ancora in corso di sviluppo degli standard affinché l'interscambio di dati non si limiti alla sola geometria, ma comprenda anche le informazioni parametriche ed alfa-numeriche. Uno dei risultati di tali studi è il formato file **IFC** (*Industry Foundation Classes*), sviluppato da *BuildingSMART* [16]. Esso è nato come modello d'interscambio dati neutro ed aperto con lo scopo di facilitare l'interoperabilità tra i diversi attori coinvolti nel processo e consentire lo scambio di un modello informativo senza perdite né distorsioni di dati [15]. Per questo motivo, secondo *BuildingSMART*, l'IFC è il formato ideale per l'approccio *Open BIM* ed è di fatto ad oggi supportato da oltre 140 piattaforme software [17].

È possibile interpretare i diversi livelli di interoperabilità, chiamati **livelli di maturità** del BIM, come un percorso graduale crescente che parte dalla metodologia CAD, caratterizzata dall'assenza di collaborazione, fino ad un ideale di "BIM condiviso", in cui tutti i professionisti cooperano contemporaneamente allo stesso modello. Tuttavia il massimo grado di maturità raggiunto finora è descritto dal cosiddetto "BIM collaborativo", in cui ogni studio, o singolo professionista, opera fin dalle fasi iniziali su un modello che viene condiviso solo in prestabiliti momenti di confronto diretto. In questo contesto, si evidenzia che la presente tesi si focalizza sul precedente stadio definito come "BIM solitario", in cui non si instaurano reali collaborazioni con altre figure professionali. È di fatto un livello in cui l'attenzione si focalizza maggiormente sulla progettazione parametrica e sulla gestione dei dati [18].



Assieme agli aspetti informativo e collaborativo, si aggiunge quello **parametrico** [18], terzo aspetto fondamentale che distingue la metodologia BIM dall'approccio tradizionale. Questo principio, che già caratterizzava il precedente *CAD 3D Object-Oriented* (OOCAD), nella metodologia BIM ha assunto un ruolo fondamentale nel rendere il **modello intelligente**, differenziandolo perciò dalle tradizionali rappresentazioni 3D. Il concetto di oggetto parametrico, nel BIM, implica un insieme di regole, variabili dimensionali ed informazioni, che permette di descrivere complesse relazioni sia geometriche che funzionali tra i singoli oggetti. Tali definizioni geometriche sono in grado di non creare incongruenze e ridondanze né tra gli oggetti stessi né all'interno della totalità del modello. Inoltre le regole che definiscono l'intrinseca parametricità del BIM fanno sì che ciascun elemento, essendo associato a geometrie e dati, sia automaticamente aggiornato, in tutte le sue visualizzazioni, ad ogni modifica effettuata [7].



Questa ricerca si sofferma specialmente sull'analisi di parametri e dati, cercando di trovare una soluzione di **automatizzazione del flusso di lavoro**. Per questo motivo viene presa in considerazione soprattutto la fase di progettazione, lasciando momentaneamente da parte quelle di realizzazione e gestione, che comunque caratterizzano la metodologia BIM. Questo non implica tuttavia di trascurare nessuna delle caratteristiche alla base del BIM in quanto, nonostante non ci sia una vera condivisione dei dati tra differenti partecipanti del processo edilizio, la condivisione delle informazioni rimane fondamentale oggetto di studio. Si ipotizzerà di fatto un flusso di lavoro in cui professionisti diversi usano due delle più utilizzate piattaforme BIM in commercio per collaborare e condividere informazioni parametriche ed alfa-numeriche.

1.1.3 Affidabilità del dato

Come anticipato, uno dei caposaldi della metodologia BIM riguarda l'aspetto informativo, grazie al quale i dati progettuali di un manufatto possono essere gestiti, modificati ed aggiornati durante l'intero ciclo di vita dell'edificio. Pertanto, durante le fasi iniziali di pianificazione e definizione della strategia operativa, devono anche essere individuate la tipologia, la qualità e la quantità di informazioni che dovranno essere interscambiate fra i vari protagonisti del processo progettuale. In particolare bisogna definire a priori quali dati sono necessari, quando e da chi devono essere implementati e a che livello di dettaglio devono essere definiti. Per questo motivo è stato necessario introdurre il concetto di “**LOD**”, che ha però sviluppato differenti significati basati sull'acronimo e sulle molteplici sfaccettature che un argomento così delicato e complesso comporta [19].

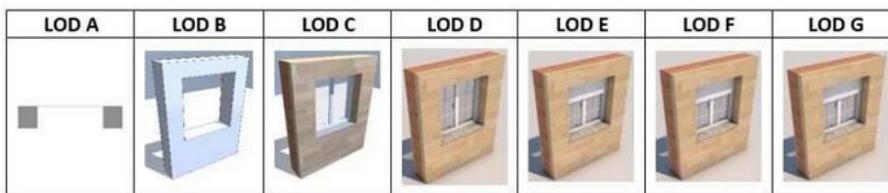
La parte 4 della **UNI 11337**, definisce il LOD come il «livello di approfondimento e stabilità dei dati e delle informazioni degli oggetti digitali che compongono i modelli» [20]. Essendo fra le più recenti normative sviluppate in questo ambito, quella italiana richiama i due più consolidati significati statunitense e inglese.

L'*American Institute of Architects* (AIA) in particolare dichiara due significati di LOD distinguendo tra “*level of detail*”, cioè la quantità di dettagli che sono inclusi in un elemento del modello, e “*level of development*”, interpretabile come l'affidabilità dei dati geometrici ed informativi annessi all'oggetto e su cui i professionisti possono basarsi nell'utilizzo del modello. Da questo si evince il grande distacco tra il primo, che può assumersi come *input* di dati, sia grafici che non, dell'elemento modellato, ed il secondo, da considerarsi come *output* di misura di sicurezza delle differenti fasi progettuali [21]. La *British Standards Institution* (BSI) ha invece volto l'attenzione sulla differenziazione tra “*level of model detail*” (LOD) e “*level of model information*” (LOI) limitando il campo del primo al solo contenuto grafico del modello e definendo col secondo tutto ciò riguarda dati ed informazioni [22]. La normativa italiana adotta ed incorpora perciò queste due direttive per regolamentare la gestione del processo informativo nel settore delle costruzioni sia dal punto di vista qualitativo che quantitativo.

La definizione dei LOD prevede che a ciascuna fase del progetto sia associato uno specifico livello di sviluppo. La classificazione statunitense prevede una scala numerica che spazia dalla rappresentazione grafica e simbolica del LOD100 ad un LOD500, i cui elementi vengono verificati in opera confermandone attendibilità grafica ed informativa [21]. La normativa italiana sviluppa una scala differenziata in base alla tipologia di intervento - nuova costruzione, restauro, territoriale, infrastrutturale e cantiere - e, sulle orme della sopracitata americana, in generale identifica gli oggetti attraverso una scala alfabetica:

- LOD A: oggetto simbolico;
- LOD B: oggetto generico;
- LOD C: oggetto definito;

- LOD D: oggetto dettagliato;
- LOD E: oggetto specifico;
- LOD F: oggetto eseguito;
- LOD G: oggetto aggiornato [20].



8 - Esempio di oggetto digitalizzato, sviluppato a LOD differenti [23]

Nell'ambito di questa tesi il LOD verrà definito in base alla diverse fasi di ricerca. Infatti, come si approfondirà meglio in seguito, ogni stadio avrà un suo specifico obiettivo, al quale è ovviamente connesso il livello di sviluppo del relativo *output*.

1.2 Progettazione parametrica

Come anticipato, i parametri sono una caratteristica intrinseca del *Building Information Modeling*, derivante dai suoi “precursori meno intelligenti” e che consente un’efficiente creazione e modifica degli elementi che compongono il modello.

Per “**parametro**” si intende una grandezza nota a cui si fa riferimento come criterio di valutazione per altre dimensioni. È essenziale specificare che tale enunciato sia un richiamo del linguaggio matematico in cui un parametro indica una costante arbitrariamente fissata da cui dipende l’andamento di una funzione [24]. Si evince quindi che, nell’ambito del BIM, sono parametrici tutti gli oggetti che, se accumulati da una variabile, al variare di tale valore generano una diversificazione all’interno della serie di elementi a cui appartengono [18]. Tale variabile può essere rappresentata da tutte le regole che possano definire l’aspetto formale dell’oggetto, come le componenti dimensionali e quantitative. Ad esempio un tavolo può avere come parametri la dimensione del piano, l’altezza e il numero di gambe, il tipo di rifinitura, il materiale e così via. In questo modo è chiaro che un singolo oggetto parametrico può dare origine ad un numero virtualmente infinito di alternative [25].

Pertanto, la **progettazione parametrica** si differenzia dalla tradizionale modellazione tridimensionale in quanto prevede lo studio delle differenti regole e relazioni che contraddistinguono i singoli componenti di ogni oggetto. Questo particolare tipo di approccio risulta molto pratico nel caso in cui si vogliano studiare molteplici alternative simultaneamente, nel minor tempo possibile. Tale caratteristica lo rende un metodo alquanto versatile in tutte le applicazioni che vertono sulla progettazione di elementi ripetibili simili fra loro, o anche che necessitano di modifiche rapide e precise, siano esse di grande o piccola entità.

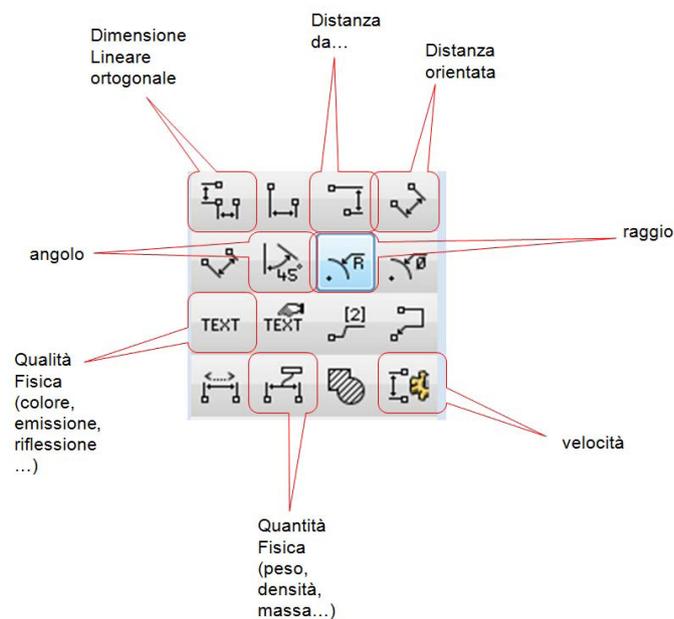
Benché la manipolazione di parametri sia un argomento molto vivo all’interno della progettazione contemporanea, non bisogna dimenticare che in verità, in architettura, questo non rappresenta un concetto del tutto nuovo. È lecito evidenziare che, per quanto possa sembrare un tipo di ragionamento che ben si accosta all’uso di software specializzati, la nascita di tale pensiero si è sviluppata ben prima

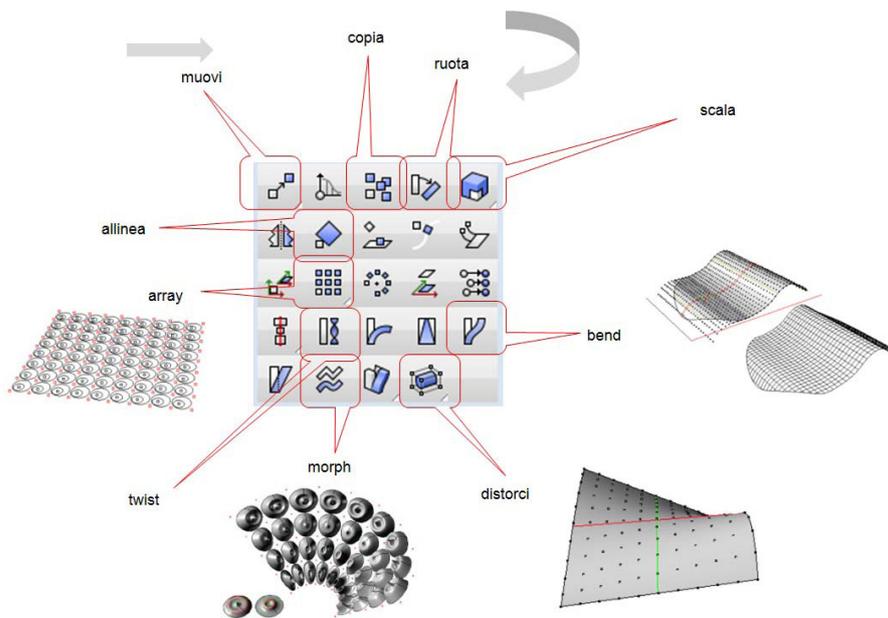
dell'avvento del computer. Questo non solo perché grandi maestri del passato come **Vitruvio** o **Leon Battista Alberti** avevano individuato equazioni matematiche e relazioni parametriche per definire le proporzioni degli ordini architettonici già nei loro trattati teorici, ma anche poiché è negli anni '40 che si può individuare la vera e propria nascita dell'architettura parametrica. L'espressione è stata coniata da **Luigi Moretti** (1906-1973), che, assieme al matematico Bruno De Finetti (1906-1985), ha ragionato sull'idea di applicare regole geometriche per la generazione di una forma parametricamente quantificata. Grazie a questa metodologia ha così sviluppato una serie di strutture sportive impiegando i parametri in progetti per stadi di calcio, tennis e nuoto, introducendo così un nuovo tipo di progettazione architettonica [18].

1.2.1 Modellazione algoritmica

Nell'ambito della progettazione parametrica bisogna tenere presente che il modello descrive processi ed azioni solo grazie all'impiego di algoritmi matematici. È necessario difatti distinguere tra il parametro, che, come già detto, rappresenta il valore al cui variare muta l'andamento di una funzione, e l'**algoritmo**, che è invece il procedimento vero e proprio che, indipendentemente dall'uso del computer, seguendo una serie ordinata di istruzioni elementari, permette di ottenere il risultato atteso [26].

9a - Esempi di parametri [26]





9b - Esempi di algoritmi [26]

In questo modo è ancora più accentuata la presenza della **matematica** come aspetto caratterizzante l'architettura del giorno d'oggi. Essa ha in realtà accompagnato da sempre l'uomo che, nella ricerca e nell'osservazione della natura, ha cercato di ricondurne i principi ad una struttura matematica. Questo profondo legame tra il linguaggio matematico e la generazione delle forme permette quindi di percepire un nuovo aspetto della progettazione architettonica, che pone al centro del modello non più la manipolazione delle forme stesse, bensì le regole che combinano ragionamenti logici, geometrici e matematici [27].

I vantaggi nell'adozione di un approccio algoritmico riguardano soprattutto l'affinità tra i meccanismi mentali di indagine e comprensione di un oggetto e la natura logica del processo, permettendo quindi un maggiore controllo degli elementi del modello e delle sue informazioni. È infatti possibile semplificare l'idea di "algoritmo" assimilandolo ad un **flusso di dati** in cui ogni componente parametrico collabora all'integrazione e alla modifica delle informazioni in entrata, ottenendo come risultato una nuova lista di dati in uscita [27]. Si assume pertanto che il limite della modellazione non termina più con le restrizioni dei singoli software, ma con la capacità di indagine formale dell'utente che li utilizza. Di contro, trovare un algoritmo efficace presuppone un'accorta conoscenza dell'oggetto da rappresentare. Bisogna quindi considerare un cambiamento nella metodologia di sviluppo del modello, in quanto le operazioni di **razionalizzazione delle forme** e scomposizione delle superfici complesse, che tradizionalmente vengono pensate come operazioni legate alle fasi più avanzate, devono diventare parte integrante del processo di definizione formale [28]. Questo implica, da parte dei progettisti, una continua revisione dei metodi e delle strategie di lavoro adottati per poter far fronte alle più incalzanti innovazioni sia sul sempre più competitivo versante del lavoro, sia per quanto riguarda il continuo aggiornamento sulla nascita di strumenti sempre più sofisticati.

Un diretto discendente della modellazione algoritmica, nato sta-

volta esclusivamente dopo l'avvento del computer e grazie all'utilizzo di appositi software parametrici, è il **design generativo**. Si tratta di un tipo di progettazione che, sfruttando algoritmi, delega al programma parametrico il compito di generare le forme. In particolare questa strategia viene adottata nel caso in cui sia necessario trovare un'opzione progettuale che offra la migliore soluzione o prestazione fra le molteplici varianti possibili, definite all'interno di una prestabilita gamma di parametri [18]. Una metodologia di questo genere non deve pertanto essere vista come una minaccia per l'autonomia del progettista. Al contrario, bisogna saperne comprendere le potenzialità, in quanto il progettista può predisporre le linee guida a cui il software dovrà attenersi e lasciare a quest'ultimo il calcolo ed il collaudo delle migliaia di ipotesi progettuali, che un essere umano non potrebbe da solo individuare in un arco di tempo limitato, per avanzare la soluzione migliore.

Prima dell'avvento dei programmi di disegno assistito la difficoltà nella descrizione delle forme più complesse era completamente nelle mani del designer. Grazie alla rivoluzione digitale, però, questa figura ha avuto maggiore libertà di azione potendosi concentrare esclusivamente sul progetto, controllato ora attraverso modelli tridimensionali, e affidando ai software il compito di ricavare piante, sezioni e dettagli propedeutici al processo produttivo. Allo stesso modo, i programmi di modellazione algoritmica, grazie alla loro base matematica, permettono di ottenere risultati **veloci** ed **accurati**, nonché un maggiore **controllo del prodotto finale**. Questo grazie al fatto che i dati inseriti dall'utente vengono direttamente trasferiti ai sistemi di produzione, che li traducono in un'immediata visualizzazione virtuale e realistica dell'oggetto [27].

1.2.2 Visual editing e BIM

Le potenzialità della modellazione algoritmica nel *Building Information Modeling* si traducono nell'opportunità, da parte del progettista, di programmare autonomamente un codice (uno **script**), che permetta di svolgere una determinata operazione ripetitiva, ma indispensabile al progetto, con l'obiettivo di **ottimizzare i tempi** di lavoro. Si sviluppa quindi la necessità di trasmutare la complessità del mondo reale nelle semplici regole ed operazioni che dovranno comporre l'algoritmo. Questo non implica inevitabilmente che il progettista debba diventare un esperto di informatica, bensì si evidenzia l'importanza di imparare ad adottare **soluzioni logiche e creative** che possano far fronte anche ai più complessi problemi progettuali [29].

Come si può immaginare, lo sviluppo dell'utilizzo dei computer, nonché il progresso tecnologico nell'amplificazione dei software parametrici, non ha fatto che accrescere la quantità e la qualità delle sperimentazioni in questo ambito progettuale. I programmi sono quindi diventati i protagonisti di tale diffusione, evolvendosi da semplici

strumenti produttivi, adottati inizialmente solo per velocizzare le operazioni, a **sistemi di controllo** per la riproduzione e generazione di forme vere e proprie. Il pregio dei software parametrici consiste nella possibilità di agire sull'algoritmo complessivo di un sistema grazie ad un'interfaccia grafica e ad un utilizzo intuitivo che non richiede conoscenze di linguaggi di programmazione. Inoltre, mediante i sistemi associativi che li caratterizzano, il progettista può di fatto agire sulla configurazione complessiva di un sistema semplicemente agendo sui singoli parametri precedentemente impostati [28]. La logica associativa permette quindi di "far propagare" automaticamente le modifiche attuate su una singola variabile dell'algoritmo all'intero *script* programmato. Bisogna sottolineare che questo approccio innovativo pone l'utente al centro del processo di definizione della forma, mentre il software resta un mezzo di comunicazione tra progettista e algoritmo [27].

L'esempio di uno dei più avanzati software per l'editor algoritmico visuale è **Grasshopper** ®, rilasciato per la prima volta nel 2007 come *plug-in* di *Rhinoceros* ®, programma per la modellazione 3D sviluppato da *Robert McNeel & Associates*. Il suo scopo è stato fin da subito quello di ovviare ai limiti di flessibilità di *Rhino* permettendo all'utente di apportare modifiche direttamente "dall'interno dell'applicazione". Come sopraccennato, il software permette inoltre di automatizzare procedure, definire geometrie attraverso funzioni matematiche, rendere rapide ed efficienti la modifica delle forme grazie all'applicazione di parametri e la possibilità di progettare geometrie molto complesse tramite la reiterazione di variabili controllate [27].

Tali vantaggi ben si accostano ai limiti della maggior parte dei programmi *BIM-based*, che possono quindi essere integrati con questo tipo di software di modellazione algoritmica per ottimizzare il flusso di lavoro. L'obiettivo è quello di ottenere un processo caratterizzato da una maggiore centralità, che unisca la **flessibilità** e la libertà espressiva della progettazione algoritmica al controllo delle informazioni e della gestione dei dati, tipici del BIM [30]. Un buon esempio di questo connubio è rappresentato da *Dynamo* ®, la piattaforma di programmazione visiva di *Autodesk* ® che consente di interagire con l'interfaccia di *Revit* ® [31]. Ciò non toglie che anche programmi sviluppati da *software house* differenti possano contare sul reciproco sostegno per raggiungere lo stesso obiettivo. È infatti questo il caso del sopracitato *Grasshopper* che, grazie al *plug-in* sviluppato da *Graphisoft* ®, riesce ad interfacciarsi con *ArchiCAD* ®, aprendo un mondo ricco di nuove possibilità di ottimizzazione ed interazione intelligente del flusso di lavoro.

Per rispondere alla crescente complessità dei problemi progettuali è possibile adottare "soluzioni ibride" che prevedano l'**integrazione tra la progettazione parametrica ed il BIM**. Un approccio tradizionale, infatti, non è più in grado di risolvere pienamente la complessità ed il numero di informazioni che un progetto di architettura contemporaneo richiede. Per questo motivo è necessario usufruire della modellazione algoritmica per integrare, e non semplicemente

esportare, geometrie ed informazioni da una piattaforma all'altra [32]. Grazie alla coordinazione tra *visual editing* e BIM, è inoltre possibile ottimizzare gli *output* di entrambe le metodologie. In questo modo si possono quindi indirizzare tempo e risorse a ragionamenti che portino a decisioni più ponderate, anziché perderne perseverando nel modellare completamente sul *BIM-based* software, che è invece più lento e limitato in questo campo [33].

1.3 Applicazione del BIM in ambito sanitario

Nel settore edilizio adottare la metodologia BIM ha il pregio di garantire una maggiore qualità ed efficienza nell'arco dell'intero ciclo di vita di un manufatto, riducendo inoltre i tempi ed i costi dovuti ad errori ed incongruenze tipici delle fasi più avanzate dei progetti realizzati in modo tradizionale. Nelle strutture sanitarie tale vantaggio si trasforma in una vera e propria esigenza se si considera la decisiva importanza ed indispensabilità che questi edifici ricoprono nelle vite dei cittadini. Si può infatti affermare che il BIM non solo permette di superare le fasi di pianificazione e progettazione stravolgendo i tradizionali sistemi progettuali e procedurali, ma consente soprattutto di oltrepassare l'immaginario limite dell'informazione singola favorendo una rete di dati finalizzata al miglioramento dell'ambiente vissuto [34].

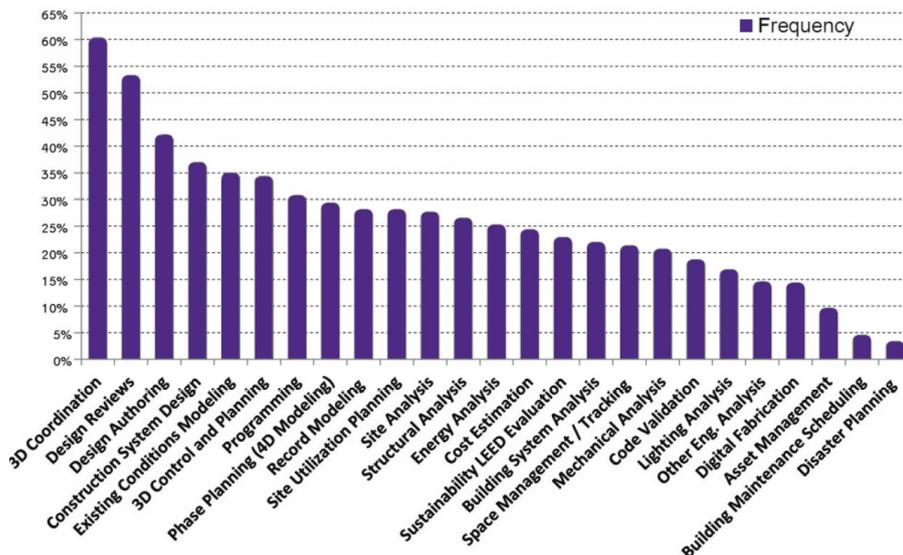
I complessi ospedalieri comportano una vasta quantità di funzionalità, complicati sistemi impiantistici ed **innumerevoli dati** da gestire all'unisono. Essi rappresentano infatti una delle tipologie più complesse da progettare ed è proprio in casi come questi che i benefici offerti dal BIM, in termini di **controllo della qualità** finale del fabbricato e di gestione accurata delle fasi progettuali, fanno davvero la differenza [35].

Come già accennato, in assenza di una metodologia che permetta di gestire tutti gli aspetti progettuali e di ottimizzare il dialogo informativo fra i differenti attori si rischia di incorrere in inutili prolungamenti dei tempi di progettazione dovuti alla necessità di correggere errori ed incongruenze, con conseguente aumento dei costi. Inoltre spesso ci si imbatte nell'irreparabile necessità di trovare un compromesso tra il superamento delle difficoltà riscontrate in corso d'opera ed il tentativo di non sprecare un imprevisto eccesso di risorse. Questo effetto domino rischia di tradursi infine nell'inadeguatezza di soddisfare l'idea iniziale di progetto, che, soprattutto nel caso di una struttura complessa e di grande entità, ne risente soprattutto in termini di qualità finale del manufatto.

È ormai assodato che il *Building Information Modeling* è applicabile alla maggior parte dei settori che costituiscono l'industria delle costruzioni, con più o meno successo, in base anche allo sviluppo della

capacità dei singoli software di interagire fra loro nella condivisione e scambio dei dati. Come evidenzia l'istogramma riportato, gli ambiti in cui il BIM riscontra maggior successo sono quelli del coordinamento 3D e della revisione progettuale. Ciò non toglie che, anche se in percentuale perlopiù inferiore al 30% di frequenza, i campi di utilizzo restino molteplici, e quindi anche i vantaggi che un progetto possa trarne [36].

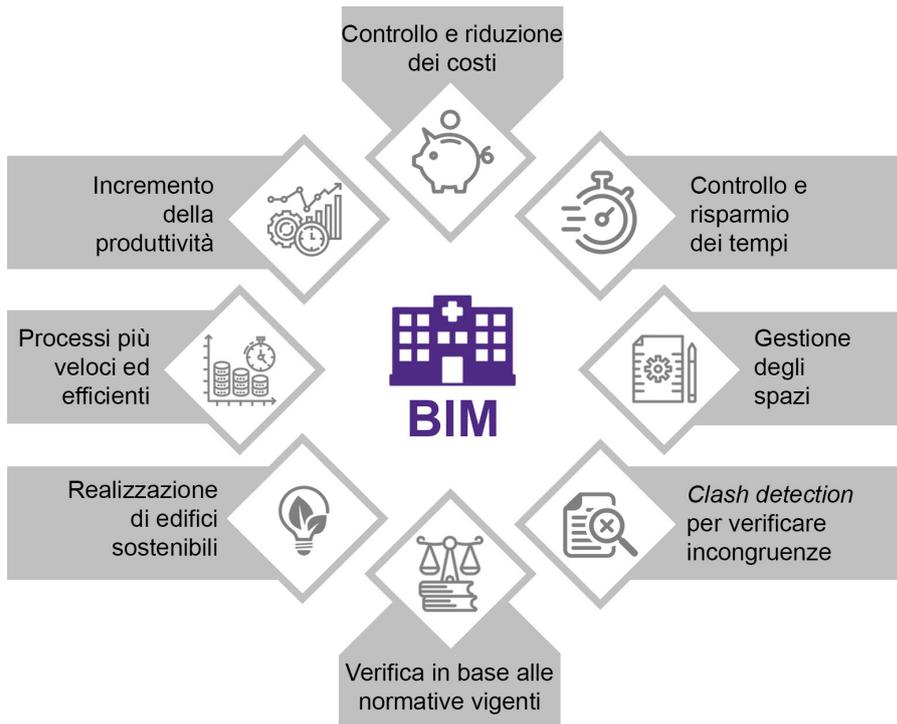
10 - Distribuzione della frequenza dell'applicazione del BIM [36]



L'applicazione del BIM in ambito sanitario, oltre ai già citati numerosi vantaggi generati dall'adozione di tale metodologia, comporta inoltre una più limpida **visione d'insieme** raggiungibile grazie all'opportunità di avvalersi di simulazioni ad ampio spettro e ad una efficiente progettazione partecipata. È infatti possibile coinvolgere, non solo le varie discipline tecniche che compongono un progetto edilizio, ma anche lo stesso personale medico – nel caso di una struttura sanitaria – per giovare del consulto di chi conosce in prima persona le necessità di un ospedale. Per fare ciò ci si può avvalere ad esempio di strumenti interattivi come la realtà aumentata, mista o virtuale, che verranno approfonditi in seguito. Grazie a tecnologie di questo calibro è possibile creare **scenari ipotetici**, i cosiddetti “*what if*”, in cui il progettista può coinvolgere l'utente nelle decisioni progettuali, dalla scelta dei materiali alla disposizione spaziale dell'equipaggiamento medico. Tali ipotesi possono inoltre riguardare **simulazioni** impiantistiche, dell'esodo in condizioni di emergenza, flussi di spostamento del personale o anche studi solari, coinvolgendo attivamente sin dal principio i vari professionisti che collaborano al progetto. Ne derivano così una riduzione del rischio di interferenze ed un aumento della consapevolezza e della **prevedibilità degli sviluppi** del progetto. [37]

La possibilità infine di mantenere il modello BIM aggiornato in tutte le fasi di progetto consente di rendere più semplici le operazioni di **gestione e manutenzione** che, generalmente per mancanza di dati, sono piuttosto impegnative, e per questo anche dispendiose in termini sia economici che temporali. Questo è tuttavia uno dei tasselli più spinosi nell'accettazione del BIM, in quanto si può incontrare molta opposizione nell'uso di risorse per l'aggiornamento costante dello stato

di fatto di un modello, soprattutto in paesi come l'Italia in cui si lavora molto sull'esistente.



11 - Principali vantaggi del BIM applicati al settore sanitario [34]

1.3.1 Caso studio: parametrizzazione di una degenza nel Parco della Salute

La presente ricerca non si focalizza su un progetto ospedaliero o su una specifica struttura sanitaria, ma ha come obiettivo quello di elaborare una **degenza parametrizzata** che possa in essi essere inserita. La premessa che riguarda la complessità caratteristica di questa tipologia di progetti serve pertanto ad anticipare la necessità di avere degli elementi, adattabili al caso studio in analisi, che possano semplificarne, almeno in parte, il flusso di lavoro.

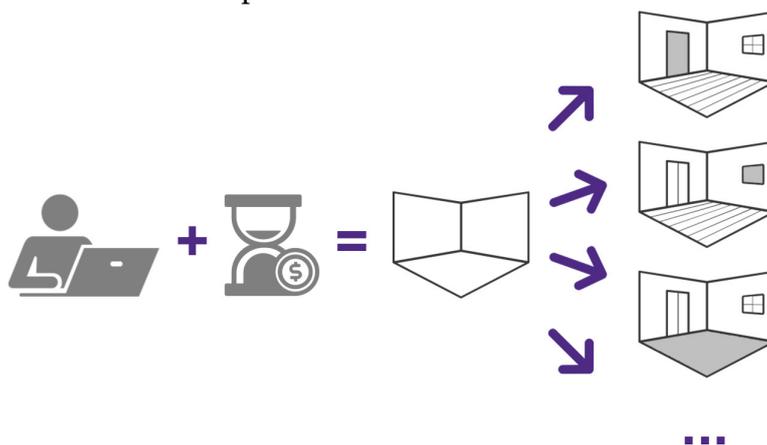
Come anticipato, i progetti delle strutture sanitarie, siano essi di piccola o grande entità, sono accumulati dalla compresenza di numerose informazioni, la cui gestione e controllo della qualità possono diventare una vera e propria “questione di vita o di morte” per i futuri pazienti. Tale responsabilità rappresenta uno dei punti cruciali per cui è importante ridurre al minimo la possibilità di errori ed incongruenze, non solo in fase di progettazione, ma anche in tutte le fasi che accompagneranno l'edificio nel suo intero ciclo di vita. Questo, assieme al fatto che un ospedale racchiude in sé un elenco di **azioni ripetute** da parte dei fruitori, rende conveniente l'adozione del BIM per la gestione di questa complicata tipologia di struttura [35]. In particolare, nell'ambito di questa tesi si è applicata tale metodologia sulla singola funzione di degenza, scelta fra le tante possibili destinazioni d'uso all'interno di un progetto ospedaliero.

Questo preambolo ha portato a ragionare sulla possibilità di ottimizzare il processo di lavoro per riuscire a ricavare un'unica camera

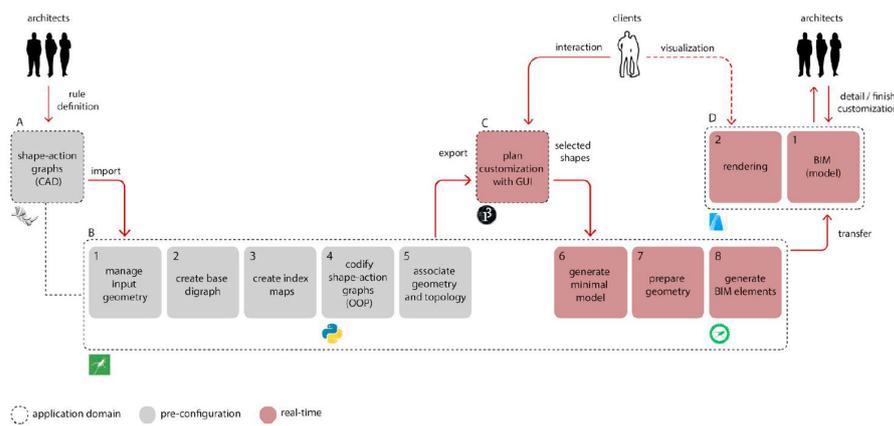
di degenza ospedaliera, alla quale includere una serie di dati parametrici e dalla quale estrapolare abachi ed informazioni di varia natura. Una logica di questo tipo consente infatti di moltiplicare questo unico “**oggetto stanza**”, nelle sue varie istanze, anziché doverne creare tanti tutti simili fra loro, incrementando a loro volta la possibilità di duplicare errori. Tale metodologia, atta non solo a semplificare il *workflow* ma anche a ridurre i tempi totali, e di conseguenza anche i costi, richiama non solo il BIM, ma anche la modellazione algoritmica. Quest’ultima infatti consente, per sua natura, di relazionare i parametri fra loro in modo da generare una rete di “**correzioni automatiche**” ogniqualvolta un’azione modifichi dei valori.

Solitamente la progettazione parametrica viene applicata o a singoli oggetti che compongono un modello, quali elementi d’arredo o impiantistici, o, più recentemente, alla progettazione preliminare di interi edifici dalle forme complicate e intenzionalmente contorte. Si può infatti dedurre che un approccio di questo genere, applicato ad una stanza di degenza, costituisca una soluzione ibrida grazie alla quale, a scapito di un grande impiego di tempo e risorse iniziali, si possono ottenere numerose e rapide soluzioni finali.

12 – Rappresentazione schematica del flusso di lavoro di parametrizzazione, grazie al quale spendendo molte risorse e ore di lavoro si possono ricavare, da un unico oggetto parametrizzato, un numero illimitato di sue possibili varianti



Sono comunque già presenti degli studi che applicano questa combinazione di metodologie per un approccio volto a coinvolgere il cliente per la progettazione degli spazi di cui fruirà in prima persona. Un esempio significativo è la ricerca svolta da Pedro Veloso, Gabriela Celani e Rodrigo Scheeren (2018) [38]. Il loro obiettivo è stato quello di assicurare la qualità del prodotto finale e al contempo di ridurre i costi del processo attraverso la sua automatizzazione. La metodologia da loro seguita ha visto in primo luogo la definizione, tramite CAD e programmi di *visual scripting*, delle regole parametriche che guidano la produzione delle geometrie. Dopodiché è l’utente stesso a dover prendere parte al processo selezionando le opzioni desiderate per la progettazione del proprio appartamento residenziale. Infine il progettista riprende in mano le redini implementando le geometrie così ottenute con informazioni, dettagli, costi e producendo tavole ed elaborati grafici finali grazie all’introduzione della metodologia BIM nel processo.



13 – Workflow del processo progettuale elaborato nello studio “From the generation of layouts to the production of construction documents: an application in the customization of apartment plans” [38]

L'impiego delle metodologie BIM e parametrica-algoritmica è tendenzialmente associato ad un tipo di **progettazione ex novo**, che quindi abbia libertà nell'ideazione delle geometrie nonché l'intera gamma di dati necessari a disposizione. Ciò non toglie però che possano essere utilizzate anche nel caso di edifici e strutture preesistenti. Nel campo del BIM questo approccio è già di largo impiego, specie in territori come quello italiano dove molti progetti riguardano il patrimonio esistente. In questi casi si adottano le medesime soluzioni di una nuova costruzione, ma spesso capita, per assenza di materiale dell'*as-is*, di dover lasciare largo spazio alle ipotesi per quanto riguarda l'aspetto informativo della struttura. Nonostante questo inconveniente infatti i vantaggi che derivano dall'adozione del BIM sono comunque, come già ribadito precedentemente, di grande agevolazione per le fasi successive alla progettazione. Più raro è invece il caso in cui venga adottata la metodologia algoritmica per la modellazione di una preesistenza, proprio perché i vantaggi che derivano dal grande *effort* iniziale riguardano soprattutto la possibilità di studiare in contemporanea molteplici soluzioni e di identificare quella migliore per il proprio caso studio.

Sebbene questo sia un argomento di tesi perlopiù teorico sullo sviluppo di una generica camera di degenza parametrizzata, si può ipotizzare di avere come obiettivo finale quello dell'inserimento dell'oggetto-stanza in analisi all'interno del caso studio di **Parco della Salute** di Torino (progetto che, ad oggi, risulta in fase di sviluppo). Il presidente della Regione Piemonte Sergio Chiamparino a gennaio del 2019 ha evidenziato infatti che il complesso, benché sia fonte di discussione da almeno 15 anni, ha davanti a sé ancora un percorso molto lungo [39]. L'obiettivo a cui si mira è quello di ottenere una struttura sanitaria che racchiuda al suo interno le più grandi **innovazioni tecnologiche** finora sviluppate. Questo concerne, tra le altre questioni, il fatto che i tempi di degenza ospedaliera siano più brevi rispetto al passato e che, per tal motivo, i posti letto, il cui reparto ricopre solitamente circa il 35% della superficie di una struttura ospedaliera [40], possano essere in numero inferiore rispetto al passato [41]. Oltre ad essere motivo di non poche polemiche e razionale inquietudine, ciò fa emergere a maggior ragione l'importanza di una camera di degenza ottimamente pensata per adempiere al suo scopo e quindi anche la necessità di adottare metodologie, come il BIM e l'approccio algoritmico,

che permettano di soddisfare tali premesse.

Il futuro Parco della Salute è stato presentato il 17 aprile 2018 come un complesso definito da un doppio ruolo acutamente combinato. Da una parte infatti è pensato per essere un **polo ospedaliero** integrato in cui convergeranno le attuali strutture ospedaliere di Molinette, Sant'Anna, Regina Margherita e CTO. Dall'altra si prevede un **polo di ricerca e didattica** che comprenda le attività che ad oggi concernono la Scuola di Medicina e Chirurgia dell'Università degli Studi di Torino [42]. A questo proposito l'assessore alla Sanità Antonio Saitta ha fatto presente che anche se a livello nazionale il progetto di Parco della Salute sarebbe una vera innovazione, non rappresenterebbe comunque il primo caso in Europa. Il riferimento che fa da capostipite al pensiero di Parco della Salute è l'ospedale **Karolinska** di Stoccolma [41] che, oltre a possedere le medesime caratteristiche di molteplicità funzionale, è stato interamente sviluppato attraverso la metodologia BIM [43].

Nonostante i 176.000 m² di Parco della Salute [44] non siano paragonabili con i 330.000 m² del complesso svedese [45] esso rappresenta a livello nazionale un importante risvolto. Questo è uno dei tanti motivi per cui questo progetto costituirebbe un valido caso studio da sviluppare in BIM. Le giustificazioni maggiori però riguardano non solo i già citati vantaggi portati dal BIM riguardo alla diminuzione degli sprechi di costi e risorse in corso d'opera, ma soprattutto quelle attinenti al cercare di limitare il rischio di corruzione nelle gare d'appalto che spesso contraddistinguono i bandi dei progetti edilizi. A tal proposito, nell'articolo del 25 febbraio 2016, pubblicato su La Stampa, è riportata l'opinione della professoressa Anna Osello, relatrice di questa tesi, che evidenzia l'importanza del BIM come modalità per introdurre un **modello anticorruzione** e grazie al quale è possibile rendere il processo condiviso con tutti gli attori e tracciabile [46].

14 – Render di progetto del Parco della Salute [47]



1.3.2 Vincoli e dati progettuali

Come riferimento tecnico per l'attinenza a vincoli e parametri definiti e prestabiliti si è scelto di adottare, tra le possibili soluzioni, la sezione legata alle strutture sanitarie de "il Nuovissimo Manuale Dell'architetto" [40]. Nello specifico, essendo questa ricerca limitata alla sola **funzione di degenza** ospedaliera, quest'ultima è la tematica del testo che è stata approfondita. Tale porzione del manuale tratta diversi aspetti che possono essere riassunti in quattro macro-argomenti.

1. Vincoli dimensionali

- Dimensione minima della camera di degenza singola (al netto di eventuali servizi): 12 m²
- Dimensione minima delle camere di degenza a più letti (al netto di eventuali servizi): 9 m² per posto letto
- Numero massimo di posti letto per camera: 4 (per le strutture esistenti: max. 6)
- Camera di degenza con servizio igienico: almeno ogni 4 posti letto
- Altezza minima dell'ambiente camera: 2,70 m
- Altezza minima di bagni e disimpegni: 2,40 m

2. Caratteristiche costruttive

- Adozione di tecniche e materiali per la salvaguardia del comfort acustico
- Applicazione di paraspigoli arrotondati alle pareti
- Impiego di rivestimenti per pavimenti con materiali resistenti a lavaggio, disinfezione ed azione meccanica
- Uso di rivestimenti verticali impermeabili, lavabili e disinfettabili (altezza min. 2,00 m)
- Dotazione di zoccolatura ai pavimenti (altezza min. 10 cm)
- Equipaggiamento di fasce protettive alle pareti, all'altezza delle maniglie di lettighe e carrelli
- Installazione delle porte dei servizi igienici con apertura verso l'esterno

3. Attrezzature ed arredi

Dotazione minima per posto letto:

- Trave testaleto
- Letto mobile (preferibilmente a tre snodi)
- Comodino
- Armadio
- Seduta a un tavolo da soggiorno-pranzo
- Sedia con braccioli
- Uso preferenziale di elementi d'arredo mobili, dotati di rotelle o facilmente spostabili

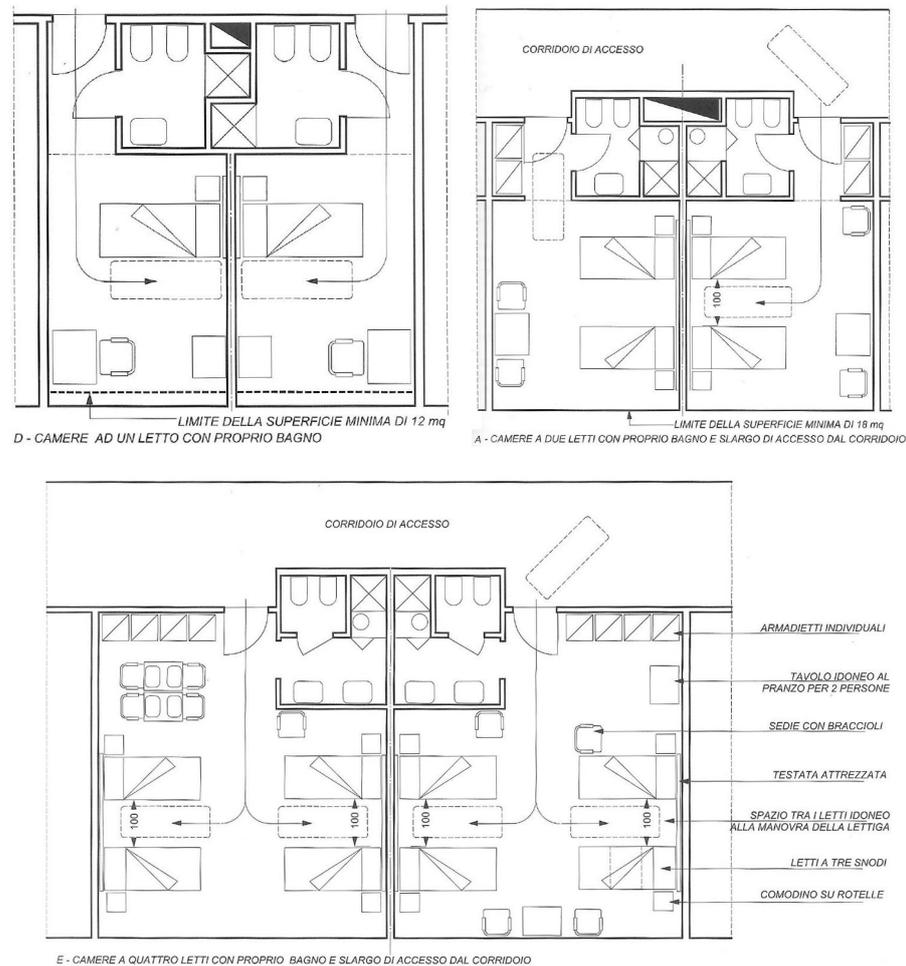
Apparecchi igienico-sanitari:

- Vaso, lavabo e bidet
- Piatto doccia con box e sgabello

4. Schemi di distribuzione

Sono di seguito riportati alcuni schemi distributivi di camere di degenza del manuale, rispettivamente con uno, due e quattro posti letto, con soluzioni differenti per quanto riguarda il servizio igienico.

15 – Alcuni schemi distributivi estratti dal manuale [40]



In generale si evince quindi che il manuale evidenzia l'aspetto organizzativo della funzione di degenza in modo tale da garantire al paziente ospedalizzato sia **privacy** che **comfort** di tipo alberghiero.

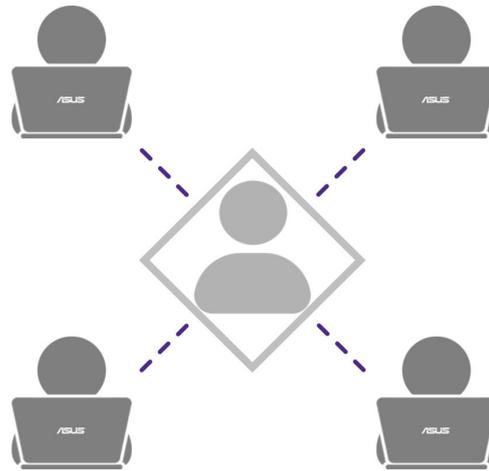
La sezione che tratta il sistema impiantistico della degenza all'interno di questa porzione di manuale è molto breve e raccomanda sommariamente la progettazione di appositi pannelli per l'ispezione e la manutenzione degli impianti elettrici nelle nuove costruzioni. Sono inoltre forniti valori standard per l'illuminazione artificiale ed il comfort termico, entrambe tematiche che in questa ricerca non verranno approfondite.

1.4 Progettazione partecipata

Conosciuta anche come “progettazione cooperativa”, “co-creazione” o “co-design”, la progettazione partecipata è un approccio volto al **coinvolgimento**, all’interno del processo progettuale, di tutte le parti interessate, ad esempio clienti, cittadini, consumatori o, nel senso più ampio, degli utenti finali a cui è destinato un progetto. In veste di partecipanti attivi, questi ultimi, aiutano a garantire che le soluzioni proposte rendano il **processo ottimizzato**, sia dal punto di vista economico che tempistico, sia da quello che riguarda la qualità finale del progetto. La progettazione partecipata rappresenta pertanto un mezzo per comprendere meglio, soddisfare e talvolta anticipare le esigenze degli attori coinvolti nel processo. L’obiettivo non è pertanto la delegazione di compiti e responsabilità da parte del progettista, ma quello di porre il cliente, e quindi anche le sue necessità, al centro del processo di progettazione [48].

Analizzando l’etimologia della parola, “partecipare” può essere inteso sia come azione del “prendere parte” ad un determinato atto o processo, sia come “essere parte” di un gruppo o di una comunità. Pertanto, vista la “natura umana e collettiva” che intrinsecamente la caratterizzano, gli studi che riguardano la progettazione partecipata, ricadono perlopiù all’interno degli **ambiti psicologici e sociali**. L’origine della progettazione partecipata si può infatti far risalire proprio a due differenti scuole di teorici della psicologia sociale. Il filone di Lewin (1890-1947) vede la ricerca come una forma di conoscenza ed allo stesso tempo di trasformazione della realtà in modo che l’oggetto di studio diventi piuttosto un soggetto attivo e partecipante [49]. La corrente di Iscoe (1921-2015) invece considera la comunità come un insieme che ha in sé le conoscenze, le risorse ed il potenziale organizzativo per realizzare un proprio cambiamento costruttivo [50]. La progettazione partecipata rappresenta pertanto un approccio di **impronta teorica** che può quindi facilmente essere applicato ai settori più disparati, da quello sanitario a quello urbanistico [51]. In particolare ben si presta all’ambito edilizio, dove la ricerca della migliore relazione con i singoli attori coinvolti rappresenta un anello fondamentale del processo progettuale.

16 – Principio base della progettazione partecipata: centralità dell'utente all'interno del processo progettuale



La derivazione della progettazione partecipata dai settori psicologico e delle scienze sociali permette di riscontrare un importante cambio di prospettiva nell'industria delle costruzioni. Si può infatti pensare che consista in un nuovo modo di pensare che possa modificare, almeno parzialmente, il modo di progettare. Il passaggio chiave di questa teoria consiste nel **progettare con gli utenti** anziché per loro, consentendo così di affrontare le cause dei problemi alla radice anziché i loro sintomi protratti nel tempo, a progetto compiuto. Con l'approccio tradizionale, infatti, dove l'utente non fa parte del *team* di progetto, il processo è incentrato sul singolo risultato finale, che risulterà basato sulle richieste iniziali e sulla competenza ed esperienza dei progettisti. La progettazione partecipata propone invece l'utente come componente centrale e cruciale del processo, permettendogli quindi di influenzare con le proprie necessità le modifiche del progetto. Da questo emerge che, per ottenere risultati maggiormente incentrati sulle persone, è necessario che la progettazione partecipata non sia una singola fase del processo progettuale, ma debba essere inteso come **parte integrante** dello stesso. C'è inoltre da aggiungere che le persone sono di per sé molto propense ad essere coinvolte in qualcosa che le riguarda da vicino, come possono essere la committenza di un edificio per un cliente o l'idoneità di una struttura sanitaria per un tecnico specializzato. È quindi da tenere in considerazione un reciproco interesse, sia da parte dei progettisti che da quella di terzi attori, nell'agevolazione di tale partecipazione, al fine di impiegare le idee sviluppate dagli utenti come fonte di miglioramento del progetto da una parte, e vedere risolte le proprie necessità ed intenti lavorativi dall'altra. Questo fa sì che i risultati ottenuti siano più significativi, soprattutto per le persone che trarranno beneficio dall'oggetto di studio del progetto [52].



Un'altra grande novità introdotta dall'approccio co-creativo applicata al settore edilizio riguarda la **modalità** vera e propria con cui gli utenti possono essere coinvolti all'interno del processo progettuale. Nel settore della psicologia, la progettazione partecipata comprende l'insieme degli strumenti atti a far sì che le persone esprimano idee ed emozioni in modo diverso dalla sola comunicazione verbale. Questo favorisce lo sviluppo di risultati molto differenti, e spesso più significativi ed efficienti, rispetto ai metodi tradizionali. Allo stesso modo, nell'industria delle costruzioni, è importante suscitare l'interesse ed il coinvolgimento degli utenti maggiormente attraverso l'**acquisizione visiva e materiale** dei loro bisogni e desideri, piuttosto che dialogando solamente con loro dei problemi e delle necessità che riguardano il progetto. Tale teoria è stata più volte verificata proprio nell'ambito delle sfere sociali e psicologiche, dove i risultati sono più immediati e facilmente dimostrabili. Ad esempio uno studio applicato ad una compagnia di assicurazioni ha supportato delle sessioni di progettazione partecipata in cui era chiesto alle persone dapprima di esprimere le proprie preferenze a parole riguardo ad un servizio o ad un marchio, e poi di metterle in pratica attraverso attività visive interattive, come un collage [48]. L'ordine di importanza dato a voce dalle persone è risultato differente nella seconda fase più "immersiva" dell'esperimento, dimostrando quanto le parole in realtà possano non rispecchiare effettivamente i pensieri degli utenti. È importante perciò, in qualsiasi settore a cui la progettazione partecipata venga applicata, coinvolgere le persone attraverso attività pratiche, oltre che verbali, in modo da stimolare la loro creatività per confidare nell'ottenimento dei migliori risultati possibili ai fini di un progetto. Nel prossimo capitolo verranno analizzati alcuni degli strumenti più comuni per l'approccio alla progettazione partecipata in ambito edilizio.

Il coinvolgimento degli utenti all'interno del processo progettuale nel settore edilizio può comportare un ampliamento delle potenzialità di un progetto. Assimilando gli utenti finali quasi al pari di esperti in materia, è infatti possibile ottenere risultati certamente diversi per quanto riguarda le necessità che una struttura deve affrontare

non solo in fase progettuale, ma anche nell'intero ciclo di vita che la interessa. Comprendere come qualcuno risolverebbe una sfida che lo riguarda in prima persona spesso fa infatti emergere nuove intuizioni sulle possibilità di un progetto. È quindi importante considerare la partecipazione diretta ed attiva degli utenti come un **potenziale valore aggiunto** al progetto che possa permettere di non banalizzare le necessità né i comportamenti delle persone. Grazie a questo approccio può trarre benefici non solo il complesso, che, rispetto ad uno progettato tradizionalmente, nell'arco della sua vita dovrà sopportare meno modifiche per gli adeguamenti alle esigenze dei suoi utilizzatori, ma soprattutto questi stessi, che vedranno rispecchiate fin da subito le proprie necessità.

1.4.1 Tecnologie per lo sviluppo della progettazione partecipata nel settore edilizio: realtà virtuale, aumentata e mista

La progettazione partecipata permette di coinvolgere gli utenti all'interno del processo progettuale consentendo ai progettisti di conoscere le loro esigenze e di mantenere un dialogo diretto con alcuni degli attori che prendono parte al progetto. Per ottenere risultati ottimali, come anticipato, non è tuttavia sufficiente la sola comunicazione verbale, ed è pertanto di fondamentale importanza analizzare quali strumenti rendano l'approccio cooperativo più efficiente. In particolare, nel settore edilizio, lo **sviluppo tecnologico**, specialmente riguardo ai **New Media**, sta incrementando sempre più l'impiego di dispositivi per la realtà virtuale, realtà aumentata e realtà mista che, fra le loro molteplici potenzialità ed applicazioni, permettono il progresso nell'ambito della progettazione partecipata.

Grazie all'adozione di queste tecnologie è possibile aumentare la percezione degli ambienti potendone osservare con maggiore accuratezza le **relazioni spaziali** che intercorrono tra gli oggetti ed identificare eventuali errori di progettazione o di posizionamento degli impianti. Risulta così molto più semplice e spontaneo, non solo per uno specialista del settore, ma anche per un utente "comune", comprendere in modo più immediato un progetto rispetto alla semplice presa visione di tavole cartacee bidimensionali [54]. Sommando questi vantaggi a quelli riscontrabili grazie all'adozione della **metodologia BIM** si ottiene inoltre un potente strumento in grado di restituire un modello virtuale, navigabile così in tutte le sue modalità e in qualsiasi luogo (si pensi ad esempio al cantiere), da cui poter estrapolare le informazioni relative al ciclo di vita di un edificio. Questo permette di prendere decisioni migliori e più sostenibili grazie alle quali riscontrare molteplici benefici. È infatti indiscutibile la semplicità con cui un utente possa così seguire e discutere col progettista in merito alle soluzioni adottate e ad eventuali modifiche proposte, in modo da evitare sorprese ed imprevisti, contribuendo a risparmi in termini di tempo, risorse e denaro [55].



18 – Possibilità di integrare in un unico processo BIM, progettazione partecipata e realtà virtuale, aumentata o mista. [56]

La **realtà virtuale** (in inglese: *Virtual Reality*, VR) nasce dalla volontà di riprodurre la realtà in un mondo artificiale, trasformandola in un ambiente esclusivamente digitale creato da uno o più computer. Questo mondo simulato non è tangibile e viene veicolato ai nostri sensi mediante delle console che permettono un'interazione in tempo reale con tutto ciò che viene riprodotto al suo interno. La realtà virtuale può essere di due tipologie in base al tipo di tecnologia impiegata. La VR immersiva adotta dei visori, che isolano completamente l'utente dall'ambiente esterno. Quella non immersiva, invece, impiega dei semplici schermi per simulare la realtà risultando di minor impatto emotivo sull'utilizzatore [57]. Entrambe le tipologie prevedono l'utilizzo di diverse periferiche contemporaneamente per simulare la realtà attraverso più stimolazioni sensoriali. Oltre al visore infatti, sono spesso presenti anche sistemi di *input* manuali come *joystick*, o *controller* di altro genere, ed auricolari per l'audio.



19 – Realtà virtuale [58]

La **realtà aumentata** (in inglese: *Augmented Reality*, AR) consiste nell'arricchimento e potenziamento della percezione del mondo reale mediante una serie di contenuti digitali che aggiungono informazioni in tempo reale a una porzione dell'ambiente circostante. L'AR non immerge quindi l'utente in un mondo parallelo, ma gli permette di sovrapporre un "*layer* conoscitivo" agli oggetti reali, integrandoli con video, audio o elementi tridimensionali. Non è inoltre necessario dotarsi di costosi strumenti, come i visori impiegati per la realtà virtuale, in quanto le informazioni che aumentano la realtà percepita possono essere visualizzate sullo schermo di un comune computer do-

tato di *webcam* o di un dispositivo mobile in grado di riconoscere appositi marcatori (*AR-Tag*). La realtà aumentata è infatti una tra le tecnologie più inconsapevolmente adottate dalle persone, in quanto fa ormai parte integrante di molti dispositivi di uso quotidiano, come alcuni accessori per l'automobile o applicazioni per *smartphone* [55].

20 – Realtà aumentata [59]



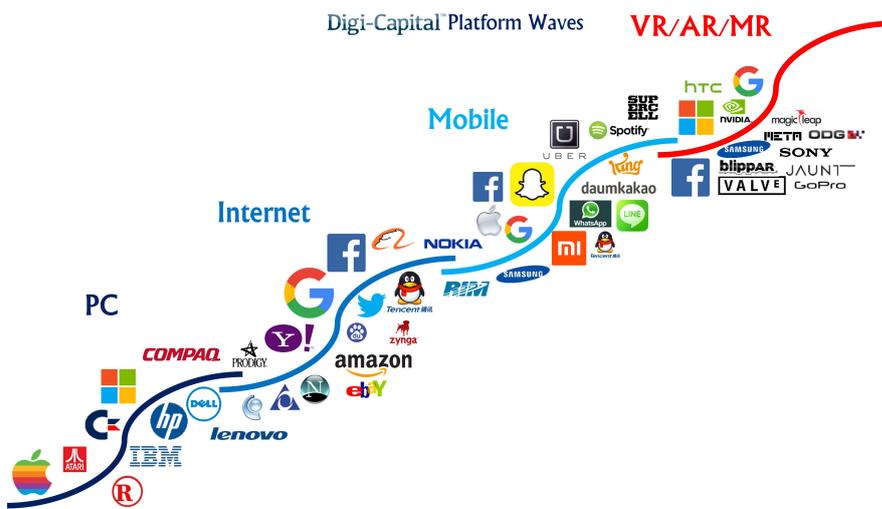
La **realtà mista** (in inglese: *Mixed Reality*, MR), come suggerisce già il termine, rappresenta una via di mezzo fra le due tecnologie appena analizzate costituendo una soluzione ibrida, che è spesso oggetto di confusione e malintesi. La MR permette contemporaneamente di osservare il mondo reale circostante aggiungendovi informazioni, prerogativa dell'AR, e di vedere ed interagire con oggetti virtuali simulati, come con la VR. Il dispositivo con cui è nata questa tecnologia è il *Microsoft HoloLens*, un visore senza cavi con cui è possibile interagire grazie a sguardi, gesti o voce, permettendo all'utente di muoversi liberamente nello spazio reale che lo circonda senza impedimenti di nessun tipo [60].

21 – Realtà mista [61]



Realtà virtuale, aumentata e mista sono attualmente tecnologie di largo impiego in molti settori, tra cui è molto sviluppato quello video-ludico. Si stanno tuttavia estendendo anche all'industria delle costruzioni, dove si prevede che possano presto diventare **strumenti**

indispensabili, non solo per il coinvolgimento degli utenti, ma anche per la progettazione vera e propria. Tutte e tre permettono infatti un nuovo tipo di approccio al progetto, attraverso cui è possibile implementare gli aspetti di previsione degli esiti e di aumento della qualità dei risultati finali, grazie ad una visualizzazione ed interazione con il modello semplice, veloce ed intuitiva. C'è infatti chi sostiene che la fusione tra la realtà virtuale, aumentata e mista ed il BIM potrebbe portare ad un'ulteriore **rivoluzione metodologica**, che trasformerebbe definitivamente il settore delle costruzioni [62]. Altri ancora prevedono che tali strumenti daranno vita alla quarta ondata di crescita tecnologica, dopo l'invenzione del *Personal Computer*, la diffusione di Internet e degli *smart device* [63].



22 – Ondate di crescita tecnologica, secondo l'indagine di Digi-Capital [63]

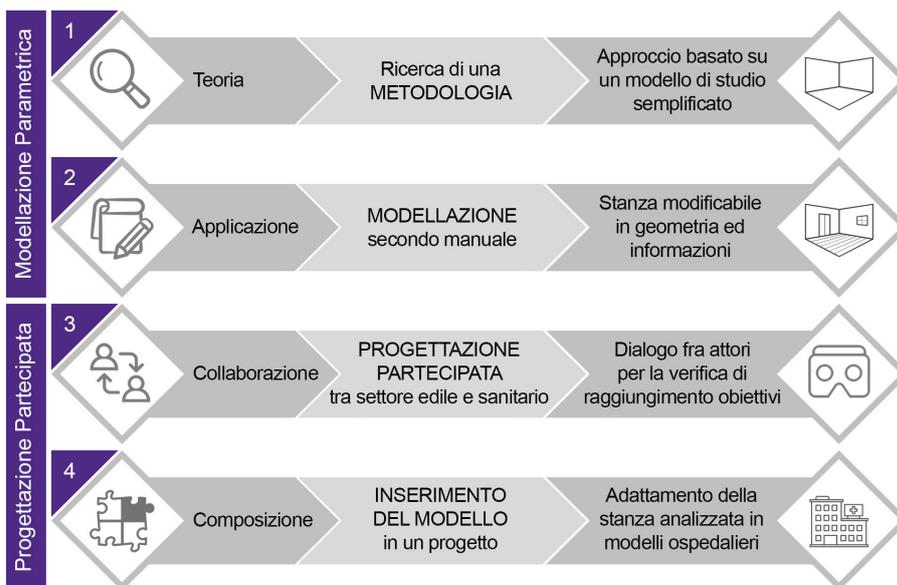


2

Metodologia

2 Metodologia

La presente ricerca è suddivisa in due tematiche principali, costituite dalla **modellazione parametrica** e dalla **progettazione partecipata**. Entrambe presentano a loro volta due fasi, distinte fra loro per tematica, obiettivi e risultati, che sono brevemente riassunti nello schema sottostante. La **prima fase** è quella più teorica, ma anche la più importante in quanto concerne lo studio del metodo e dei software da adottare per la fase successiva, centrale allo sviluppo della ricerca. Il primo *output* ottenuto è perciò un modello di studio testato e semplificato, che viene implementato nella **seconda fase** di modellazione. La terza rappresenta anche l'**ultima fase analizzata** e proietta infine il modello ottenuto nell'ottica di una progettazione partecipata mediante l'adozione di specifiche soluzioni progettuali. La **quarta fase** è stata raffigurata per contestualizzare la ricerca nel più ampio ambito della modellazione di una struttura sanitaria, che non verrà qui approfondito.

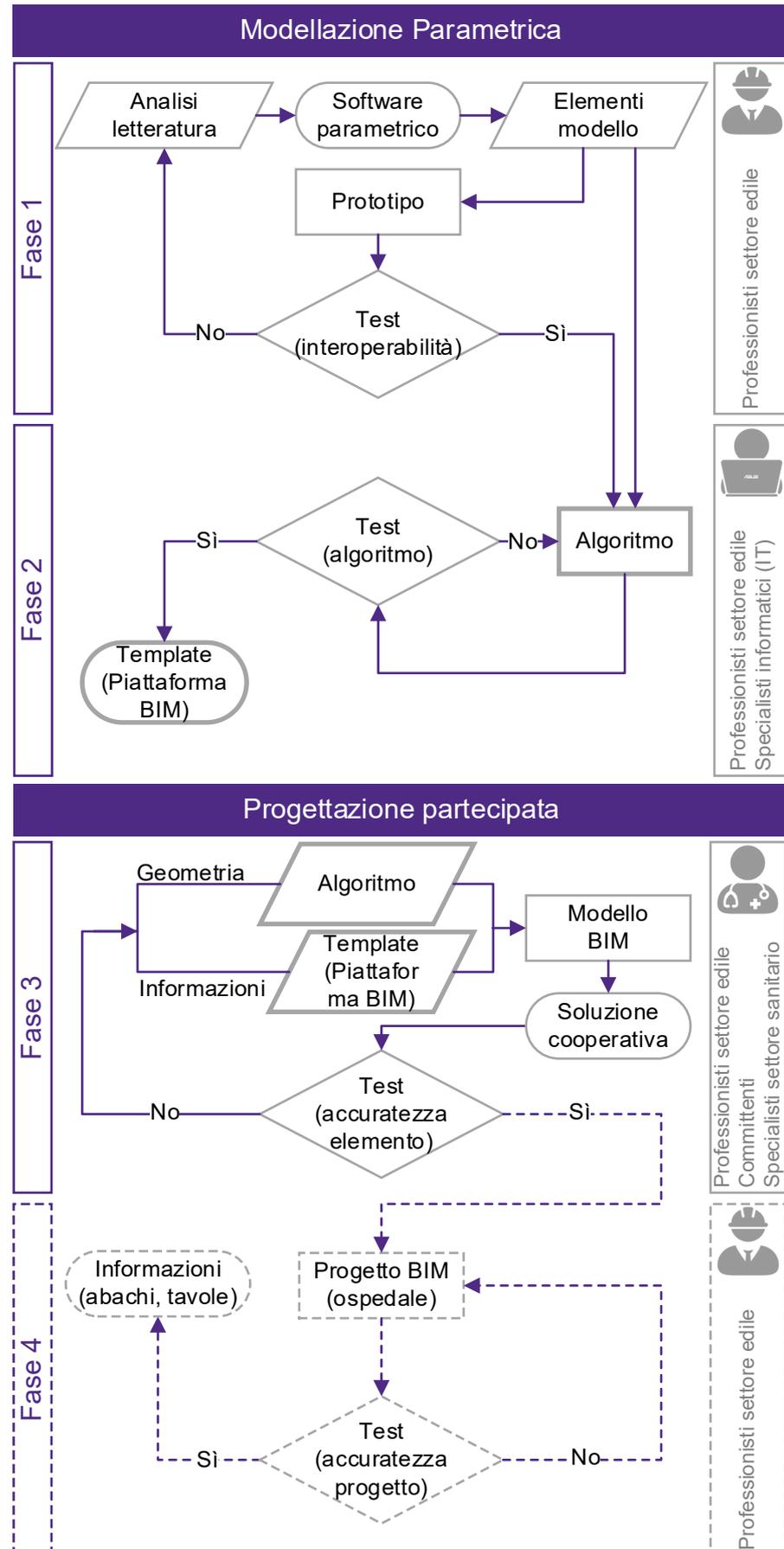


23 – Schema metodologico: panoramica di tematiche, obiettivi e risultati delle fasi che compongono il processo progettuale analizzato

Il seguente diagramma di flusso rappresenta invece più nello specifico la metodologia applicata alla ricerca, risaltando i passaggi chiave di ogni fase. Sono inoltre definiti gli **attori coinvolti**, prevedendo per le fasi di modellazione parametrica solo professionisti settoriali e ag-

giungendo, in quelle della progettazione partecipata, il coinvolgimento di utenti esterni. Nei capitoli successivi ogni fase verrà approfondita riportando la porzione di diagramma che la concerne.

24 – Diagramma di flusso: schema metodologico della tesi di ricerca



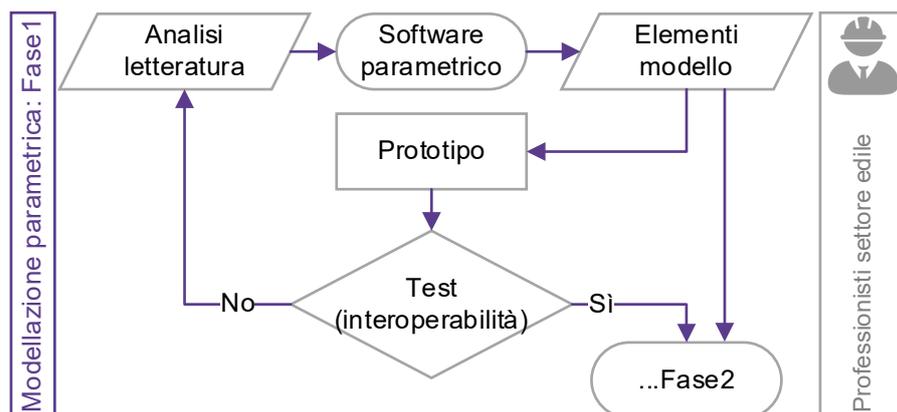
La ricerca verte sul tentativo di progettare una camera di degenza parametrica che possa essere impiegata in **diverse piattaforme BIM** per la modifica da parte di studi di professionisti che abbiano la possibilità di lavorare insieme pur possedendo licenze di programmi differenti fra loro. Si vuole quindi ottenere un “**oggetto-stanza**” che possa essere utile per velocizzare il flusso di lavoro dei progetti di opere sanitarie, che sia insieme facilmente **modificabile** nonché **gestibile** nel tempo. Come anticipato nei capitoli precedenti infatti, i progetti ospedalieri - lo dimostra anche il caso del Parco della Salute - richiedono innumerevoli attenzioni non solo in fase di progetto, ma anche durante l'intero ciclo di vita del manufatto. Per questo motivo risulta fondamentale trovare una metodologia che porti a limitare il più possibile le risorse e i costi volti alla correzione di errori, incongruenze o mancanza di informazioni, per poterli rivolgere verso aspetti più rilevanti del progetto.

Con queste stesse finalità verrà approfondita una delle possibili soluzioni per la **progettazione partecipata** volta a coinvolgere medici e professionisti del settore sanitario, ma anche committenti o utilizzatori, che potranno fornire la propria competenza grazie ad un'interfaccia intuitiva anche per coloro che non sono pratici dell'ambiente edilizio. Per cercare di raggiungere questi obiettivi saranno sviluppate ipotesi differenti tra loro, che manterranno sempre come caratteristica comune il fulcro del **Building Information Modeling**.

2.1 Fase 1: Test d'interoperabilità di un modello concettuale

La prima fase della metodologia ha come scopo quello di esaminare non solo i possibili software da impiegare, ma anche – e soprattutto – la vera e propria metodologia da adottare. Pertanto essa si fonda su un base teorica dello stato dell'arte, grazie alla quale ricercare il programma con cui generare un prototipo rudimentale. Il passo successivo riguarda la definizione degli elementi da modellare a la conseguente **semplificazione** dell'oggetto-degenza parametrizzato, modellato attraverso i software scelti, allo scopo di testarne le modalità di condivisione. L'aspetto che caratterizza maggiormente l'intera fase è infatti quello dei **test di interoperabilità**, essendo questa un requisito fondamentale per lo scambio del modello e delle informazioni di lavoro con altri studi o professionisti. Si evince quindi che, se il prototipo sviluppato non supera le aspettative e non soddisfa gli obiettivi, si avvia un processo descritto da ciclicità che prevede di ripetere le ricerche, cambiare software adottati, rimodellare il prototipo, basato su specifici elementi selezionati, e testarlo nuovamente fin quando non ci si avvicina ai risultati attesi.

L'intera fase si propone come primo passo verso la vera e propria modellazione, che caratterizza invece la fase successiva. Si può quindi impostare un margine di livello di dettaglio molto basso, che potrebbe corrispondere ad un **LOD A** della normativa UNI 11337-4 [20], che identifichi il prototipo della degenza come oggetto simbolico, atto ad essere impiegato per testarne l'interoperabilità.

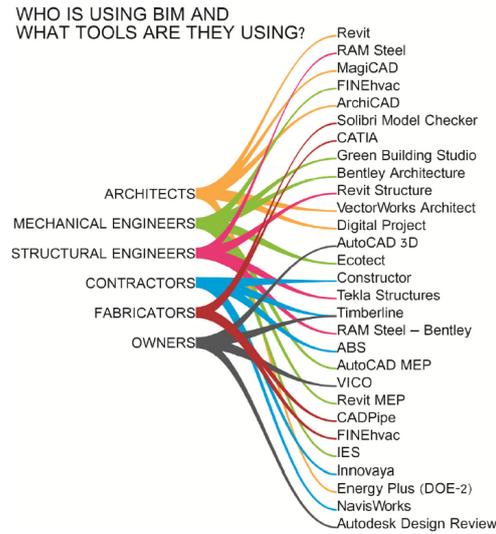


25 – Schema metodologico della fase 1. Per una visione d'insieme vd. diagramma di flusso dell'intera metodologia (fig. 24)

2.1.1 Scelta della piattaforma BIM: Revit VS ArchiCAD

La metodologia BIM, essendo di “recente” sviluppo, presenta ancora alcuni elementi da perfezionare, e così sono anche i software su di essa basati. La verità è che il BIM mira alla **condivisione** del flusso di lavoro volta a coinvolgere diversi professionisti del settore edilizio, ognuno dei quali riscontra maggiori vantaggi con un programma piuttosto che con un altro. Ciascun software ha sia le potenzialità che le limitazioni per poter diventare quello principale, ed è per questo motivo che risulta maggiormente conveniente combinare più applicazioni fra loro per ottimizzare il flusso di lavoro [36].

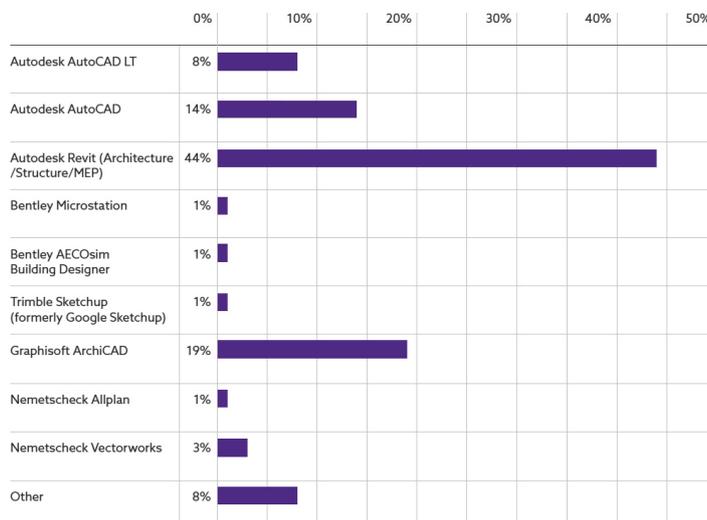
26 - Piattaforme adottate in base all'utilizzo relativo alla professione [36]



Nella presente fase della tesi si cerca di simulare un processo di **interoperabilità** mettendo a confronto due delle piattaforme BIM di più largo impiego. Per quanto limitarsi a due soli software consista in una grande restrizione, si pensa che un'analisi di questo tipo possa comunque dar luce a differenti riflessioni e criticità che potrebbero essere generalizzate a più ampio spettro. Pertanto, analizzando il *National BIM Report* del 2018, sono stati scelti *Revit* e *ArchiCAD* per questo studio, in quanto dichiarati gli strumenti maggiormente adottati per la modellazione [64].

27 - Diffusione delle principali piattaforme usate per modellare [64]

When producing drawings or models, which of the following tools do you mainly use?



Revit, introdotto da *Autodesk* nel 2002, è l'attuale leader di mercato BIM, mentre il primato di **ArchiCAD** consta nell'essere la più vecchia piattaforma BIM esistente, essendo stata sviluppata da *Graphisoft* nel 1980, la cui *software house* è poi stata acquisita da *Nemetschek* nel 2007. I punti di forza dei due programmi consistono in un'interfaccia intuitiva e menu ordinatamente organizzati per categoria, per quanto piuttosto differenti fra loro sia per forma che per funzioni. Possono importare elaborati derivanti da software o fonti esterne, come disegni *.dwg*, modelli *.skp*, immagini *.jpeg* o file *.pdf*. Inoltre entrambi hanno la possibilità di implementare il proprio *database* attraverso oggetti esterni, sia scaricabili da internet che modellati singolarmente dall'utente stesso [7]. Ciò che però apre un divario tra *Revit* e *ArchiCAD* è la modalità di condivisione dei progetti, l'organizzazione degli oggetti, le notifiche e la gestione dei parametri, caratteristica quest'ultima non poco rilevante per lo sviluppo di questa ricerca.

- **Worksharing e teamwork**

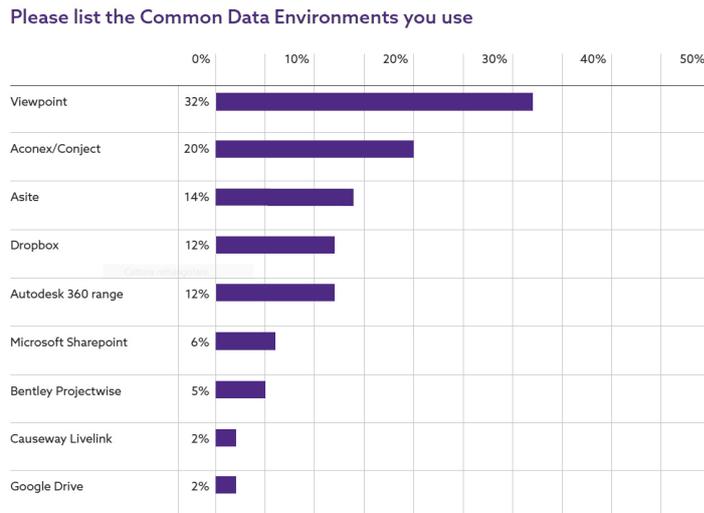
Sia *Revit* che *ArchiCAD* permettono di scegliere se utilizzare un metodo di condivisione del lavoro basato su file o su server in base alla complessità e alle dimensioni del progetto, nonché all'ubicazione geografica dei vari membri del *team*. Entrambe le opzioni richiedono una meticolosa suddivisione di ruoli ed operazioni nelle fasi preliminari di progetto, poiché sarebbe poi complicato e rischioso apportare modifiche di questo genere in corso d'opera.

La condivisione basata sui **link** è piuttosto simile nelle due piattaforme in analisi, in quanto in entrambi i casi vengono impiegati file esterni non solo per spartire il lavoro, ma anche per suddividere il modello in più parti, diminuendo così la dimensione di quello finale. Tali *link* possono facilmente essere aggiornati e ricaricati dal loro proprietario. Tuttavia non tutti i membri del *team* devono tassativamente possederli per poterli visualizzare, benché in questo modo non possano verificarne immediatamente gli eventuali conflitti all'interno del modello ospite. Piccole differenze consistono nel fatto che, contrariamente ad *ArchiCAD*, *Revit* non permette di salvare in versioni precedenti, risultando quindi non retro-compatibile e obbligando tutti i membri del *team* a operare con la medesima versione del software. Inoltre *Revit* non consente di lavorare ad un file collegato ad un modello ospite mentre quest'ultimo è aperto, se non scollegandolo da esso e poi ricollegandolo, oppure chiudendo momentaneamente l'ospite, cosa che, lavorando con *ArchiCAD*, non è invece causa di tali complicazioni.

L'alternativa del **teamwork** presenta invece più differenze tra i due software, ma è quella più sicura ed ottimizzata per la condivisione ed il lavoro in tempo reale nell'ottica di evitare conflitti nel progetto. *Revit* opera attraverso la creazione di modelli locali da parte dei singoli utenti partendo da un **modello centrale** che può essere collocato su pressoché qualunque

28 – Diffusione delle più usate piattaforme di condivisione di dati [64]

piattaforma di condivisione, sia essa *Google Drive*, *Dropbox*, o quelle proprie di *Autodesk*.



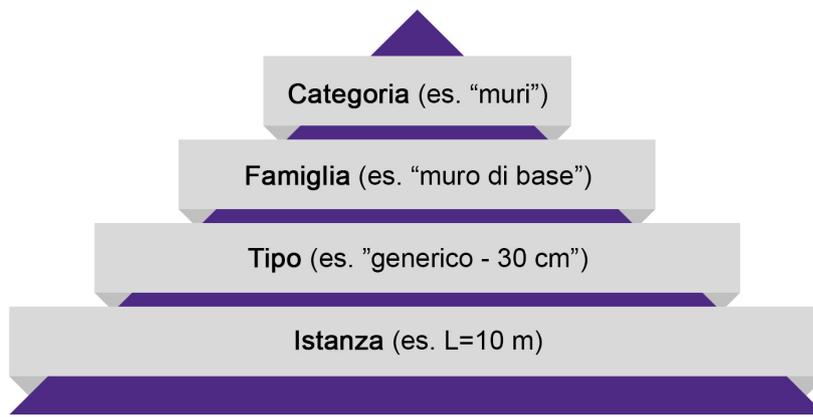
Nel file di progetto che diventerà il modello centrale è possibile gestire i **workset**, grazie ai quali vengono definiti i ruoli e le azioni eseguibili dai vari membri del *team*. Con questo metodo bisogna ricordarsi di applicare il medesimo percorso della cartella di piattaforma condivisa per poter sincronizzare correttamente il file locale con quello centrale.

Il *teamwork* di *ArchiCAD* invece non viene gestito direttamente dall'interno del software, ma dalla piattaforma *Graphisoft BIM cloud*, che deve essere installata e configurata separatamente rispetto al software, preferibilmente su un apposito server o computer. Tramite un'interfaccia web l'amministratore può quindi gestire i vari utenti, con i corrispettivi ruoli ed azioni effettuabili, e le cartelle di progetto. Terminate queste operazioni preliminari basterà infine essere loggati con il proprio account all'interno di *ArchiCAD* per poter accedere al modello condiviso e sincronizzare il proprio con il resto del *team*.

È forse superfluo dire che sia *Revit* che *ArchiCAD* sono stati pensati per avere una condivisione ottimizzata fra gli utenti che utilizzano il medesimo software, viceversa sarà questione di successiva analisi approfondirne la reciproca interoperabilità.

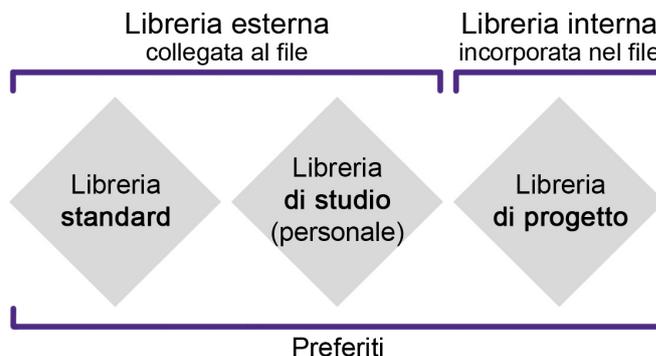
- **Librerie, famiglie e oggetti**

La gestione dei componenti di *Revit* è organizzata secondo una **struttura gerarchica** che vede al vertice le categorie, ovvero i gruppi di elementi, di seguito le famiglie, cioè le loro classi, poi i tipi, che definiscono più nel dettaglio il componente, ed infine le istanze, che rappresentano il singolo elemento inserito nel progetto. Le **famiglie** sono di tre tipologie: quelle di sistema, che sono preimpostate e non modificabili in alcun modo se non duplicandole, quelle caricabili, che sono create in file esterni (con estensione *.rfa*) e le famiglie locali, che sono create appositamente dall'utente all'interno di uno specifico progetto.



È possibile fare un discorso analogo per quanto riguarda le **librerie** di *ArchiCAD*, il quale invece non possiede una struttura piramidale per la gestione dei suoi componenti. Sono pertanto presenti tre tipi di librerie (*.lcf*), che racchiudono al loro interno i differenti oggetti (*.gsm*). Le librerie esterne, ovvero quelle collegate al file *ArchiCAD* di progetto, possono essere standard o di studio. La prima categoria deve essere sempre caricata nel file e viene generalmente installata assieme alla piattaforma BIM, infatti la sua versione dipende da quella di *ArchiCAD*. La seconda invece, detta di studio o personale, contiene oggetti scaricati dal web, realizzati direttamente dall'utente o importati nel software in altri formati, ed è l'utente stesso che deve gestire la sua organizzazione, ad esempio ordinando le cartelle per tematiche (arredo, texture, impianti, strutture ecc.) e decidere, in base al progetto, cosa caricare. L'ultima libreria è quella di progetto, che è interna al file *ArchiCAD* e comprende tutti gli oggetti sviluppati in modo specifico per il progetto in corso ed è quindi incorporata nel file. Come per le famiglie caricabili di *Revit*, la libreria di studio deve essere condivisa ed accessibile a tutto il *team* di lavoro, mentre le famiglie locali e gli oggetti di libreria di progetto, essendo inclusi nel file, vengono automaticamente condivisi assieme ad esso.

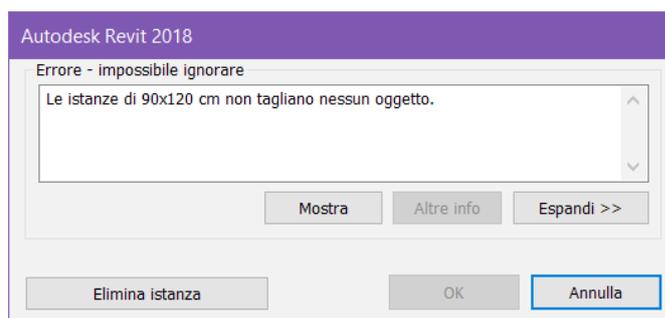
A partire dalla versione 20 di *ArchiCAD* (commercializzata dal 2016) è inoltre integrata la funzione dei "**preferiti**", che incrementa notevolmente la velocità di modellazione. Grazie ai preferiti è infatti possibile salvare un oggetto di libreria specificandone determinate informazioni, come dimensioni e proprietà, in modo da poterlo impiegare più volte nel modello senza doverle reinserire da zero né copiare un oggetto precedentemente definito. È possibile in questo senso fare una similitudine tra i preferiti di *ArchiCAD* e i tipi di *Revit*, in quanto entrambi possiedono impostazioni tali per cui, una volta inseriti nel modello, danno vita a delle istanze.



- **Notifiche**

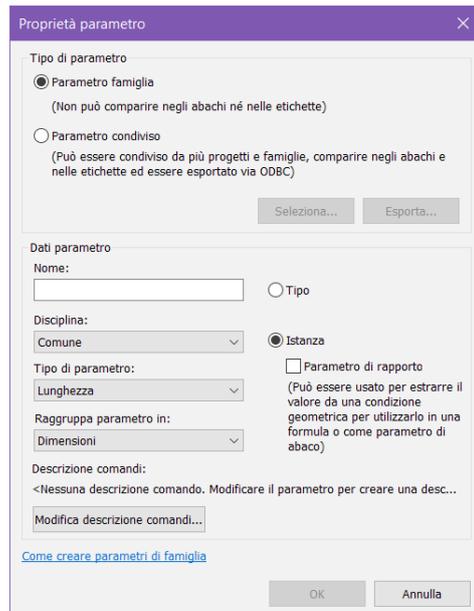
La *graphical user interface* (GUI) di *Revit* è ottimizzata per rendere sempre cosciente l'utente di ciò che sta per fare attraverso delle finestre di notifica. Grazie a queste ultime è infatti più semplice comprendere perché alcune azioni si verifichino o non possano essere portate a termine. *ArchiCAD* invece non sempre permette la stessa percezione di **consapevolezza**, consentendo quindi grossolani errori che sarebbe semplice evitare. Un banale esempio consiste nella modellazione dei serramenti, cioè di elementi che necessitano di una struttura principale che li ospiti. Dopo aver generato una finestra su *ArchiCAD*, posizionandola sul relativo muro ospite, è possibile che l'utente, inconsapevolmente, sposti il serramento all'esterno della struttura, rendendo tale elemento privo di significato. Questo non abolisce tuttavia la relazione fra le due strutture, che continuano ad essere subordinate, infatti eliminando quella principale si cancellano entrambe. Su *Revit* invece non è possibile commettere un errore simile in quanto, cercando di spostare la finestra all'esterno del suo muro, appare subito un messaggio che notifica la violazione del vincolo relativo all'*host*, senza cui prendere visione non è possibile procedere. Ciò che ne risulta è che, nonostante in entrambi i casi il serramento sia vincolato al muro, solo nell'ultimo l'utente può veramente giovare delle necessarie restrizioni native del software, grazie alle notifiche che segnalano errori di questo tipo. Più un modello diventa complicato, più è difficile da parte dell'utente gestire manualmente tutte le possibili variabili che lo compongono, ed è pertanto un considerevole aiuto avere un sistema di notifiche che assista le operazioni immerse nel software.

31 – Esempio di una notifica di *Revit* non ignorabile, relativa all'impossibilità di spostare un serramento all'esterno del muro che lo ospita, se non eliminando l'istanza stessa



- **Parametri**

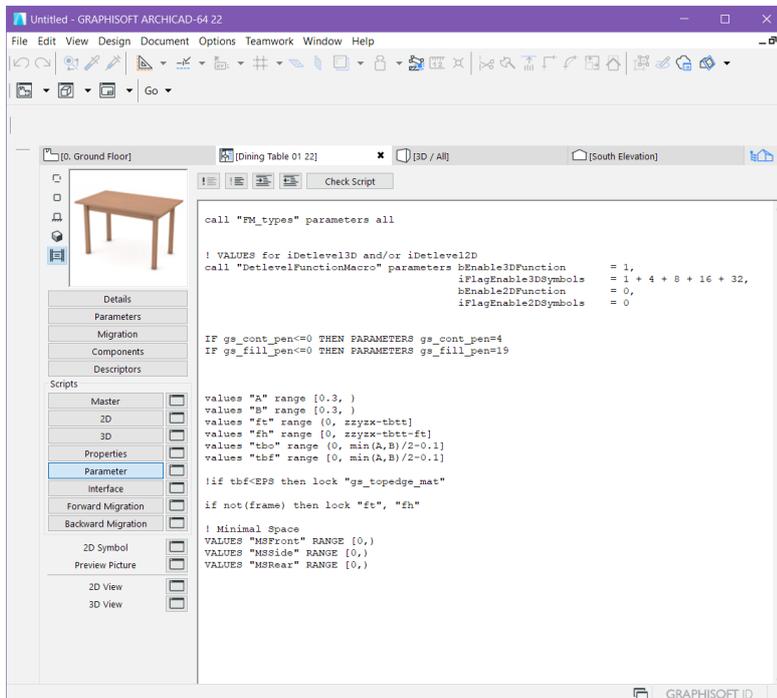
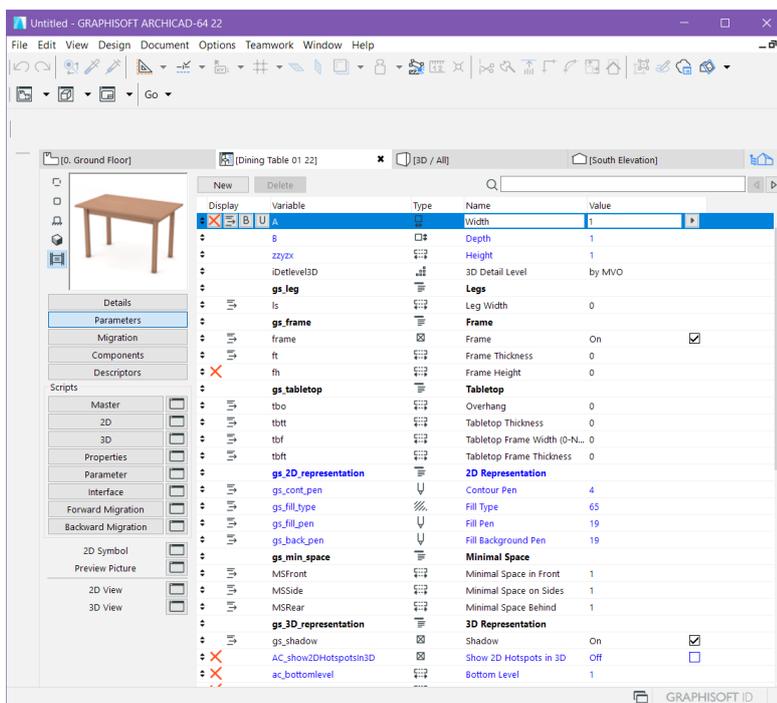
Autodesk ha lavorato molto alla **semplificazione** della creazione dei componenti parametrici di *Revit* in modo che l'utente possa crearne facilmente di nuovi in base all'occorrenza. È infatti possibile gestire i parametri attraverso comandi semplici ed intuitivi, come quelli di quotatura e di allineamento, e creare formule algoritmiche mediante menu non dissimili da quelli da cui si modificano le comuni proprietà degli elementi stessi. In seguito verrà approfondita la creazione di una famiglia parametrica nei suoi singoli passaggi chiave.



32 – Interfaccia di creazione di un parametro dal file di famiglia di Revit

ArchiCAD non è invece ugualmente ottimizzato per la gestione dei parametri, in quanto i suoi componenti sono definiti in **GDL** (*Geometric Description Language*), che è un linguaggio di programmazione simile al *Basic*. È implicito specificare quindi che per chi non ha competenze in questo campo risulta molto complesso elaborare nuovi parametri e modificare quelli esistenti senza dover prima consultare delle apposite guide applicative. Per questo motivo gli oggetti presenti nelle librerie di *ArchiCAD* sono preimpostati per avere numerosi parametri di base, cosicché l'utente possa scegliere fra le molteplici opzioni disponibili, anziché doverne creare di nuove.

33 – Esempio di interfaccia dei parametri di un oggetto di libreria su ArchiCAD. In alto l'editing, in basso lo script



Come anticipato quindi, sia *Revit* che *ArchiCAD* hanno insieme pregi e difetti, che li rendono entrambi al contempo validi ed incompleti, ed è questo il motivo per cui è necessario trovare un punto di incontro per poter lavorare con entrambi in una *team* senza dover scegliere “il software migliore”.

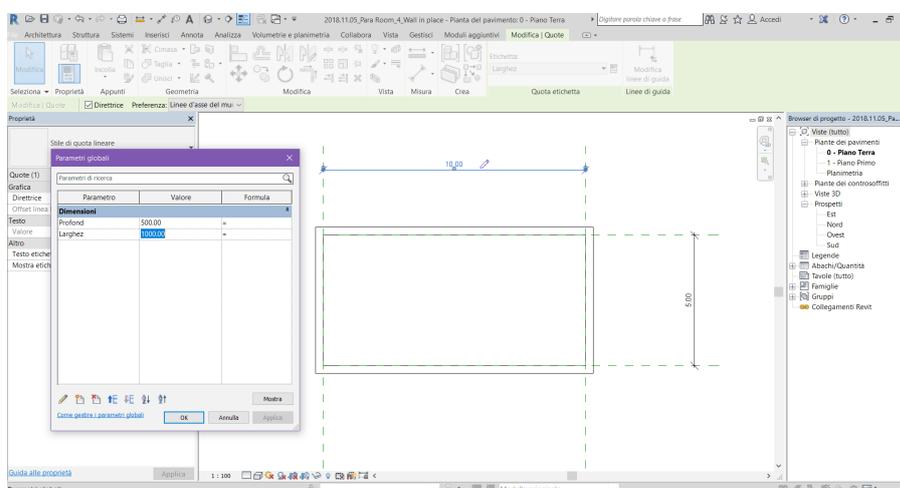
2.1.2 Parametrizzazione di un “oggetto-stanza” con Revit

Dopo aver analizzato le possibilità delle due piattaforme BIM in analisi nel campo della parametrizzazione, si è scelto di escludere momentaneamente l'opzione di creare oggetti parametrici direttamente all'interno di *ArchiCAD*, per mancanza di competenze in ambito di programmazione. L'obiettivo è perciò quello di progettare la camera di degenza parametrica partendo da *Revit*, che è apparso come il software maggiormente compatibile per un più semplice approccio di questo tipo, per poi testare l'importazione del risultato così ottenuto su *ArchiCAD*.

Innanzitutto si deve constatare quali elementi siano necessari per la creazione dell'oggetto-stanza che si vuole progettare, essendo questa fase ancora di **sperimentazione del metodo** e non di vera e propria modellazione. Si decide quindi di provare a creare un primo **prototipo semplificato** che abbia almeno la possibilità di applicare parametri dinamici ai muri, per poi evolvere eventualmente il metodo anche ad altre variabili che caratterizzeranno la degenza.

Per prima cosa si è ritenuto opportuno impiegare un approccio basato sui **link**, in quanto rappresentano un metodo di condivisione ottimizzato per progetti con moduli ripetuti, come avviene nel caso delle degenze nei modelli ospedalieri. Si presentano quindi due alternative, ovvero se decidere di operare direttamente tramite l'impiego di muri o se identificare prima una massa concettuale alla quale applicare poi le strutture.

La prima opzione vede dapprima la predisposizione di quattro piani di riferimento paralleli a due a due, ai quali vengono in seguito allineati e bloccati dei **muri** generici, che rappresentano il perimetro della degenza. Dopodiché è sufficiente associare delle quote ai piani per poter assegnare a queste ultime dei parametri, ad esempio larghezza e profondità. In questo modo, al variare del valore all'interno della quota variano anche le dimensioni dei muri.

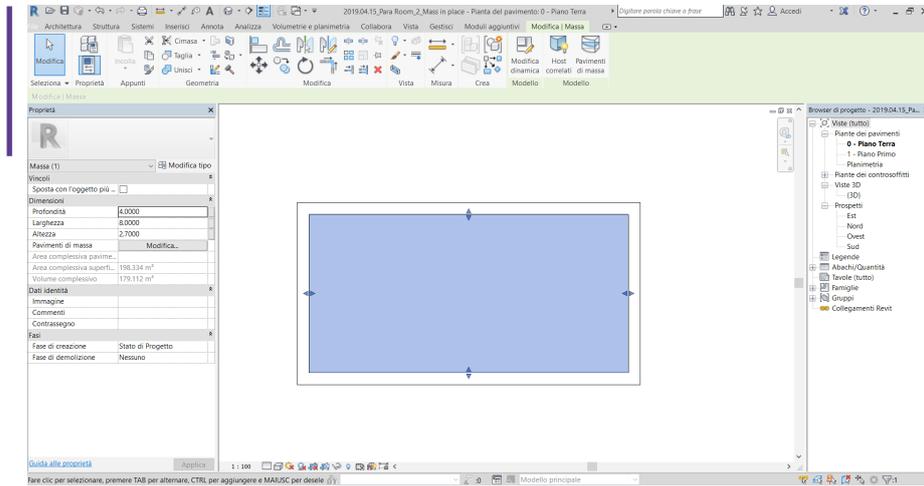


34 - *Ipotesi 1: Generazione dei parametri direttamente agli elementi “muro” nel progetto*

L'altra ipotesi verte invece sulla realizzazione di una **massa concettuale** nel file di progetto, che identifica il volume interno della degenza, alla quale associare i piani di riferimento. Il procedimento di

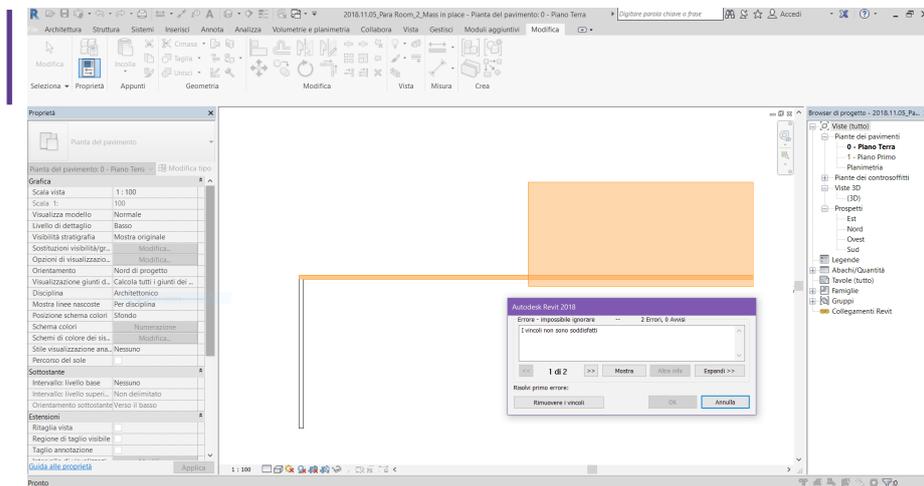
creazione dei parametri è lo stesso, ma, contrariamente al caso precedente, in cui i parametri immessi potevano essere solo di tipo globale, in questo caso possono essere creati parametri di tipo o di istanza. Ciò comporta che nel primo caso è necessario editare i parametri agendo direttamente sulla quota, nel secondo invece, selezionando la massa locale, è possibile apportare modifiche semplicemente dal menu delle proprietà, nel quale compaiono i **parametri di istanza** precedentemente associati alla massa.

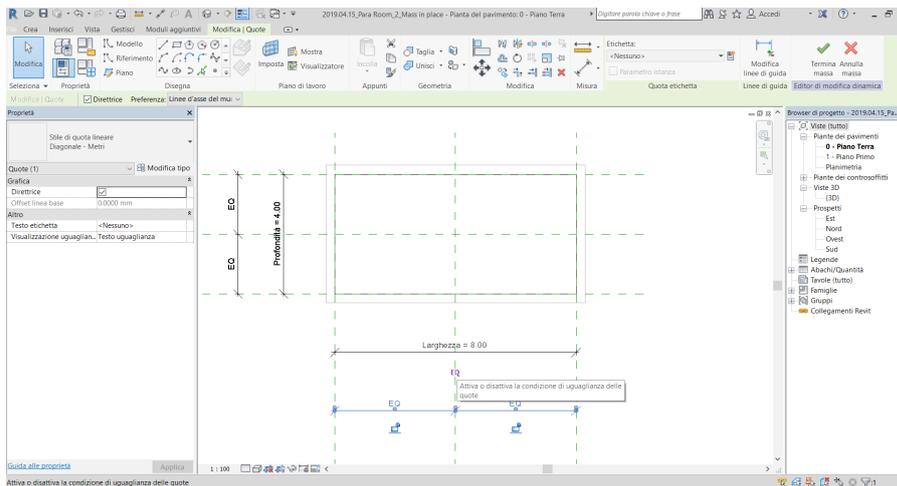
35 – *Ipotesi 2: Visualizzazione dei parametri di istanza applicati ad una massa nel file di progetto*



È indispensabile infine aggiungere due ulteriori piani di riferimento perpendicolari fra loro, ai quali attribuire la definizione di **origine** e a cui associare delle quote con la condizione di uguaglianza per mantenere gli elementi legati ad un punto centrale del file ed evitare così che i vincoli creino conflitti seguitamente alle modifiche effettuate sui parametri.

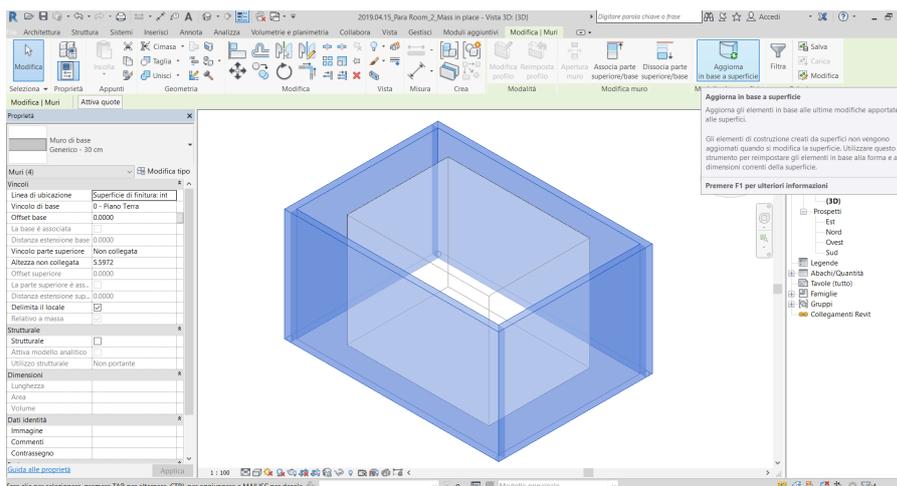
36 – *Ipotesi 2: Errore di interferenza dei vincoli*





37 – *Ipotesi 2: Correzione dell'errore tramite l'inserimento di due piani di riferimento adibiti ad origine e delle quote con condizione di uguaglianza*

A questo punto è sufficiente applicare dei “muri da superficie” all'esterno della massa creata per poter perimetrare fisicamente la degenza. Con questo metodo i muri, pur essendo associati ad una superficie, non vengono automaticamente aggiornati nella loro geometria al variare delle modifiche apportate alla massa. È pertanto necessario ricordarsi di **modificarli manualmente**, tramite l'apposito strumento, ogni volta che si modificano i parametri creati.

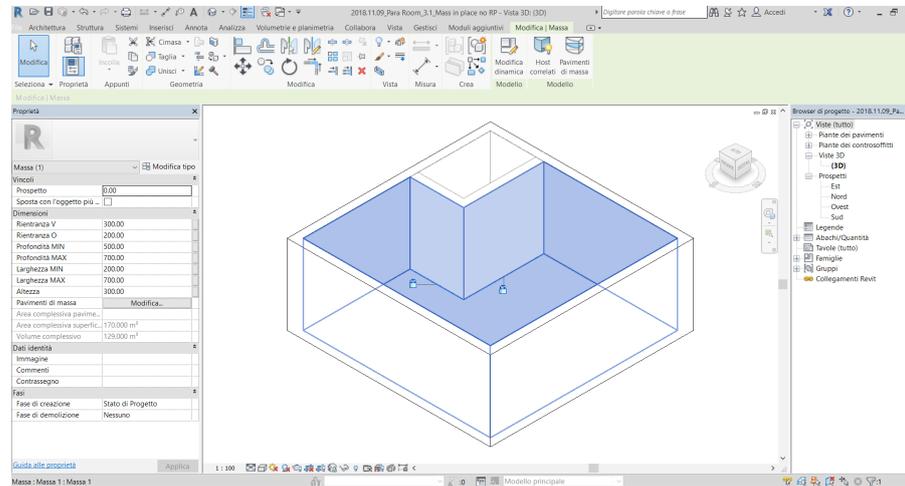


38 – *Ipotesi 2: Modifica dei muri da superficie attraverso il comando di aggiornamento manuale*

Raggiunto questo obiettivo, con i limiti sopra elencati che esso concerne, si può adesso provare ad implementare ed evolvere l'oggetto parametrico così creato con ulteriori caratteristiche. Basandosi sugli archetipi del manuale dell'architetto si desume che alcune camere di degenza debbano avere un **servizio annesso** direttamente. L'inserimento dell'elemento bagno, anch'esso parametrizzato, consiste perciò nel secondo gradino di difficoltà imposto da superare. Partendo dalla seconda ipotesi, che è risultata più flessibile, si cerca ora di creare un file di progetto contenente due masse distinte, una a forma regolare, per identificare il locale bagno, e l'altra con una rientranza che si incastrerà “a puzzle” per la modellazione della degenza. Per semplificazione grafica si procede con questa prima ipotesi applicando i parametri direttamente alle superfici della massa anziché a dei piani di riferimento, come sarebbe invece più corretto fare da un punto di vista logico. I muri da superficie in questo caso devono essere applicati talvolta dal filo interno, talvolta da quello esterno delle due masse, in modo da

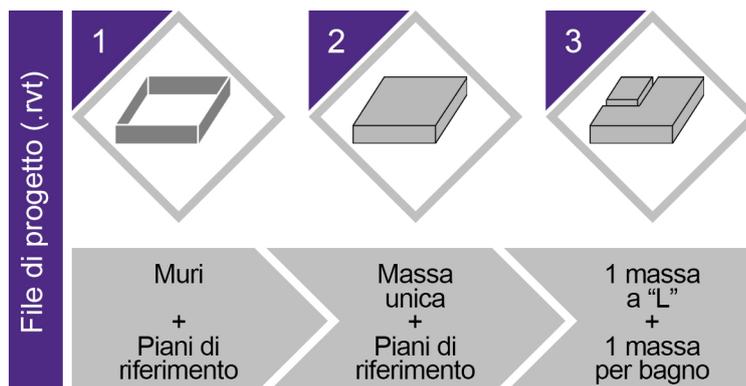
ottenere “pulizia volumetrica” per quanto riguarda una più sistematica predisposizione al calcolo della superficie della degenza. Una volta applicati i parametri e i muri ad entrambe le masse si può quindi provare a variare i parametri per valutare la correttezza delle impostazioni applicate. Il risultato, rispetto all'unica massa di partenza del prototipo precedente, perde molta flessibilità, rendendo **poco intuitivo** e maneggevole la modifica dei parametri stessi.

39 – *Ipotesi 3: Modellazione di due masse parametriche all'interno del progetto, una adibita a degenza e una a servizio igienico*



Dopo aver testato questi due prototipi si prova infine a valutare l'effettiva efficienza di questo approccio collegandoli ad un modello che interpreti simbolicamente il progetto di un complesso sanitario. Ne risulta che per modificare i muri da superficie ed i parametri dimensionali del file *linkato* bisogna necessariamente aprire il file della degenza. A questo si aggiunge il limite già riscontrato in precedenza del non poter lavorare su un file collegato mentre quello ospite è ancora aperto. Si evince quindi che tale approccio comporta **tempistiche** non poco rilevanti ad ogni modifica da attuare sul file di degenza a causa della stessa apertura e chiusura dei vari modelli. Inoltre bisogna considerare la **mole di file** contenenti l'oggetto-stanza che devono essere creati: ad ogni minima differenza fra le degenze corrisponde un diverso modello da collegare, aumentando la possibilità di errori e la confusione generale nell'organizzazione delle cartelle di lavoro. Tale limite, non essendo proprio del programma utilizzato, ma della metodologia adottata, è quindi motivo di abbandono di questi prototipi iniziali, che non risultano ottimizzati per l'obiettivo preposto e che non verranno quindi ulteriormente approfonditi.

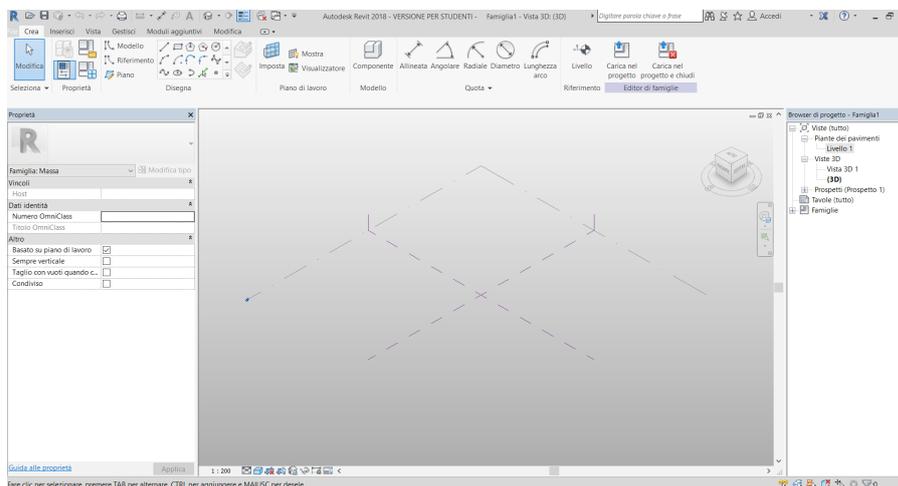
40 – *Specchietto riassuntivo delle ipotesi relative alla parametrizzazione con base di partenza un file di progetto (.rvt), con l'obiettivo di collegare tale modello tramite link*



Approfittando ancora dell'intuitività di *Revit* nella gestione dei parametri si può pertanto intraprendere una nuova strada che eluda le restrizioni riscontrate nell'approccio per *link*. Questa via prevede la generazione di una **famiglia parametrica** grazie alla quale ottenere un unico elemento che possa però essere replicato più volte, anche con tipi ed istanze differenti, all'interno dello stesso modello.

Si cerca pertanto di non rendere vani i risultati ottenuti con le due soluzioni precedenti e di trasporli quindi in questo nuovo approccio. Il primo prototipo è fallito in quanto, benché sia possibile creare famiglie nidificate l'una nell'altra, i muri appartengono alla tipologia di famiglia di sistema, che, tra le già menzionate limitazioni, non possono essere inserite all'interno di altre. È quindi impossibile creare una famiglia contenente tali strutture, il che porta obbligatoriamente ad aggirare l'ostacolo ragionando con le masse concettuali.

La creazione di una famiglia apre subito il quesito di quale **template (.rft)** impiegare per la modellazione della stessa. In questo caso si sceglie di basarsi sul *template* preimpostato "massa metrica", al cui interno si trovano già dei piani di riferimento perpendicolari fra loro.



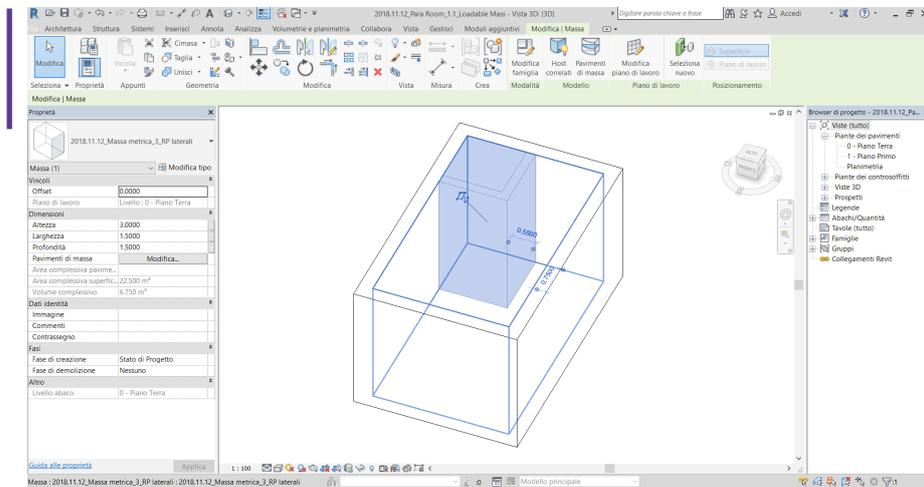
41 – Template “massa metrica”, da cui parte la modellazione di una famiglia composta da masse

L'associazione dei parametri alla massa realizzata all'interno della famiglia non è differente dal procedimento messo in atto per quella nel file di progetto. Ciò che cambia è il fatto che ora i muri da superficie verranno associati alla famiglia, anziché direttamente alla massa, ma le operazioni e i limiti riguardo al loro aggiornamento manuale sono ancora gli stessi.

Per la creazione del locale adibito a bagno si può provare a semplificare ed affinare l'opzione già sviluppata per il prototipo *link*. Ad esempio si può usare la medesima famiglia di massa concettuale dalla geometria regolare inserendola due volte nel progetto. Sovrapponendo le **due masse** è quindi possibile adibirne una al servizio igienico ed una al locale degenza, dando ovviamente dei parametri dimensionali differenti. Per far sì che modificando i parametri i due volumi non si disallineino fra loro è necessario che la famiglia della massa creata abbia dei piani di riferimento con l'opzione di definizione dell'origine nel vertice che si desidera sia costantemente allineato. Ciò non solo comporta la creazione di quattro masse geometricamente identiche, ma con origini identificate ognuna nei quattro differenti vertici, ma

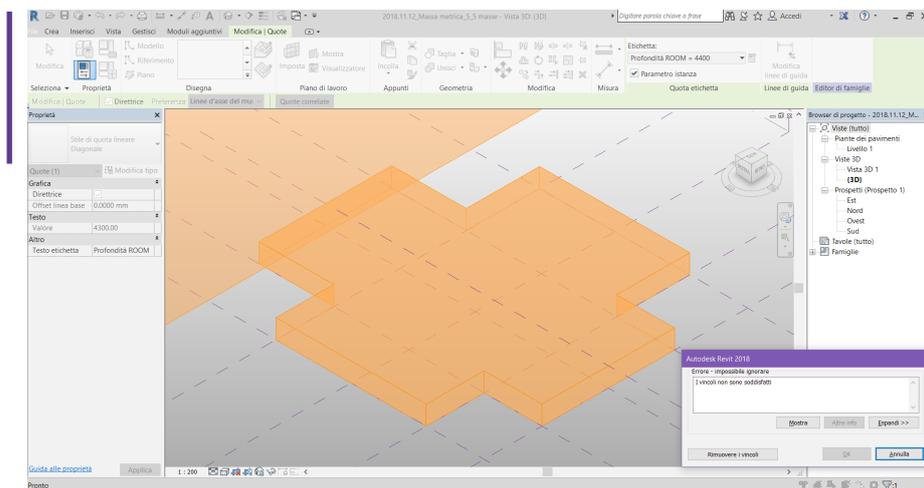
anche che l'ipotesi dell'inserimento del bagno sia semplificata. Infatti, in questo modo il locale adibito a servizio può essere sempre e solo compreso nel filo determinato dalla volumetria della massa-degenza, il che corrisponde solo ad una delle possibili composizioni volumetriche degli schemi distributivi illustrati nel manuale dell'architetto [40].

42 – *Ipotesi 4: Sovrapposizione di due masse per la generazione di due ambienti*

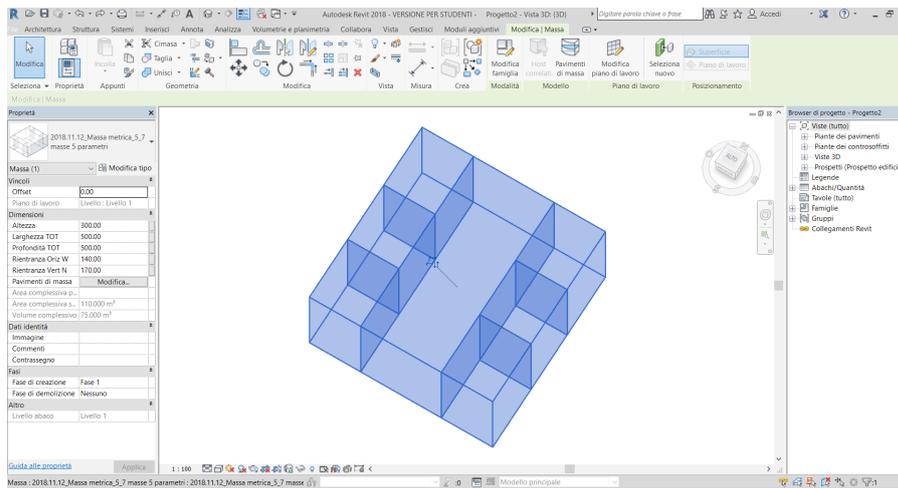


Volendo perfezionare ulteriormente quest'ultima opzione, con l'obiettivo di ottenere un'unica famiglia anziché quattro differenti, si può letteralmente **scomporre il problema** in più parti. Si procede quindi immaginando un'unica famiglia composta da più masse. Il primo tentativo è quello di creare una massa centrale crociforme, nei cui cavetti inserire quattro volumi parallelepipedi che faranno da bagni. Modellando la massa principale però si nota subito che modificarne i parametri comporta lo svincolamento dai piani di riferimento, ed è quindi chiaro che risulta una soluzione troppo rigida, che non merita di essere portata a termine.

43 – *Ipotesi 5: Interferenze nella creazione di una massa centrale, con spazi per i volumi dei servizi*

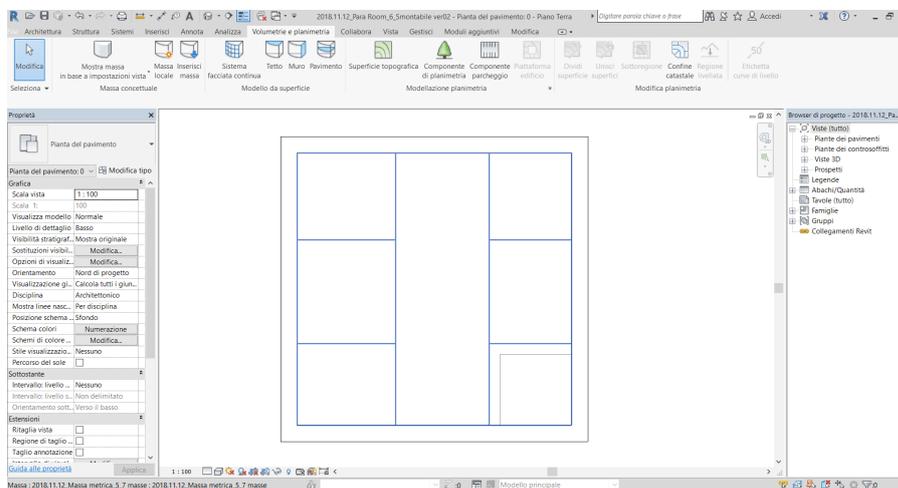


Si effettua perciò un'ulteriore scomposizione delle volumetrie in modo da ottenere un risultato più **flexibile**. Mantenendo lo stesso ragionamento di base, è infatti sufficiente suddividere la massa centrale crociforme in tre differenti forme adiacenti, cosicché in totale, aggiungendo i volumi dei bagni perimetrali, si ottenga una famiglia composta da sette masse e governata da **cinque parametri** dimensionali.



44 – *Ipotesi 6: Famiglia composta da sette masse, con cinque parametri, inserita in un file di progetto*

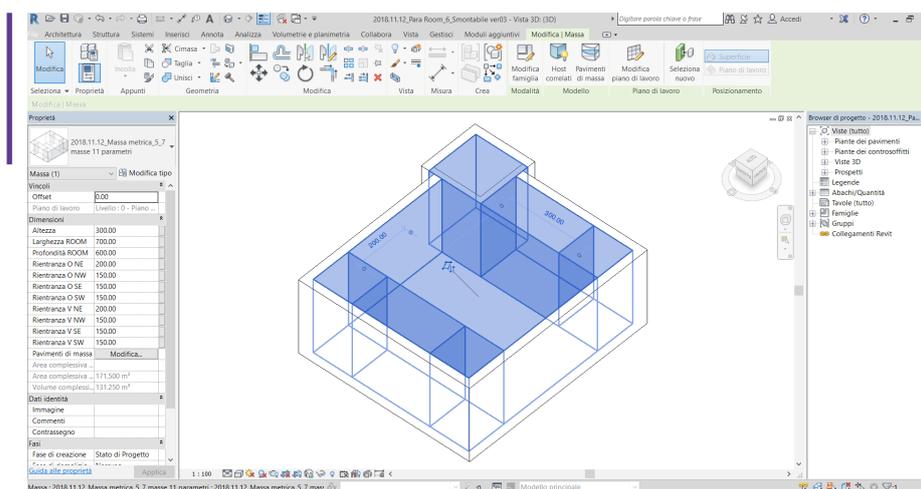
Caricando in seguito la famiglia all'interno di un file di progetto l'utente potrà quindi decidere a quali superfici applicare i muri, in base alla posizione del bagno desiderata.



45 – *Ipotesi 6: Generazione dei soli muri necessari, partendo dalla famiglia di masse*

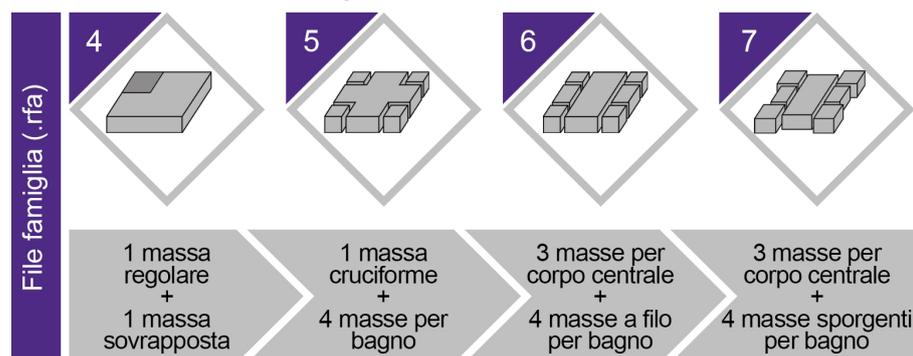
In questo modo permane comunque il vincolo che mantiene il filo del locale bagno allineato a quello della degenza. Per poter sviluppare il servizio all'esterno del filo della degenza è necessario infatti implementare ulteriormente i parametri, mantenendo le masse con la stessa composizione, in modo da raggiungere una maggiore flessibilità progettuale. Si ottiene perciò una seconda famiglia composta di nuovo da sette masse, ma stavolta con **undici parametri**. Il numero di variabili aumenta perché, rispetto al caso precedente, ogni massa-bagno deve avere la possibilità di essere svincolata e dimensionata separatamente rispetto alle altre. Vengono perciò creati due parametri per ogni bagno potenziale, per dargli la possibilità di sporgere dalla geometria del corpo centrale. Prima questa non era considerata una priorità in quanto modificando le dimensioni di uno dei volumi ad angolo non si modificava anche il perimetro complessivo della degenza. Le masse nel primo caso possono quindi essere associate allo stesso piano di riferimento, limitando così il numero di variabili, e tenendo sempre presente che la successiva estrusione dei muri riguarda solo le superfici di interesse, e non tutte quelle presenti nella famiglia.

46 – *Ipotesi 7: Famiglia composta da sette masse, con undici parametri, inserita in un file di progetto*



La seconda famiglia così ottenuta presenta però le stesse difficoltà riscontrate nel caso della prima ipotesi in cui è stato modellato il bagno. La **scarsa immediatezza** di quest'ultima lascia quindi trasparire la possibilità che debbano coesistere due tipi di famiglie, una più semplice da gestire, ma con le masse vincolate al filo della degenza, e una meno intuitiva che tuttavia lascia una maggiore flessibilità nella modellazione dei servizi igienici.

47 – *Specchietto riassuntivo delle ipotesi relative alla parametrizzazione con base di partenza un file famiglia (.rfa)*



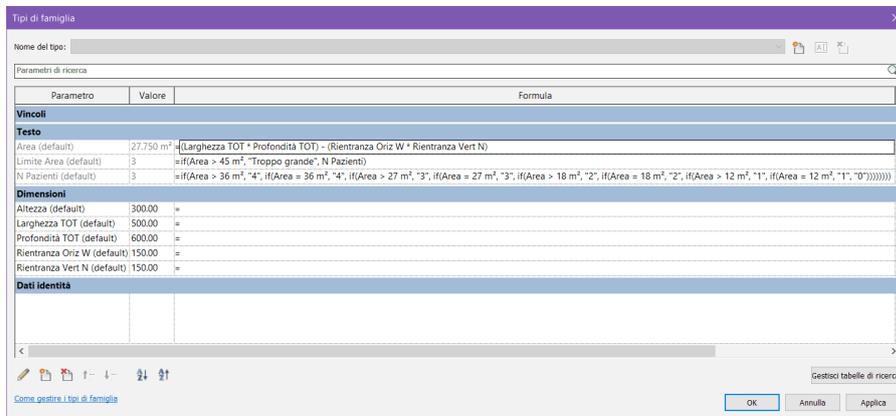
Si può ora incrementare il livello di complessità provando ad inserire delle nuove variabili per continuare a rendere più completo il prototipo della degenza parametrica. Si passa quindi ora alla creazione di **parametri di area e testo**, che possano essere utili al modellatore nella definizione della stanza. La superficie è infatti una variabile molto importante dalla quale dipende il numero di pazienti che la degenza può ospitare. È quindi fondamentale, basandosi sui valori del manuale dell'architetto [40], correlare questi due valori in modo che l'utente conosca automaticamente i vincoli che devono essere editati senza dover consultare la normativa. Per fare ciò è sufficiente creare un nuovo parametro dal menu dei tipi di famiglia, a cui immettere come valori i parametri dimensionali precedentemente creati. In particolare per il calcolo dell'area della famiglia con i cinque parametri dimensionali (con bagni a filo della degenza) bisogna sottrarre alla superficie totale uno dei volumi dei bagni. La medesima strategia riscontra però degli ostacoli nella famiglia con undici parametri, in quanto i volumi dei bagni non sono uguali fra loro e non è possibile conoscere con anticipo quale dei quattro verrà utilizzato. Pertanto la superficie da sottrarre all'area totale andrebbe editata manualmente in base a quale massa

vedrà i propri muri estrusi.

Ottenuto il valore dell'area, che si modifica così automaticamente ad ogni cambiamento delle variabili dimensionali a cui è correlato, si può generare il **numero di pazienti** come parametro di testo ad esso correlato. Per raggiungere questo obiettivo si usa la formula “if” che presenta l'espressione:

if (condizione, “vero”, “falso”).

Una gestione poco efficiente dei simboli nell'interfaccia delle proprietà di *Revit* ha portato a scomporre ogni condizione di “maggiore o uguale” in due parti, rendendo l'espressione più lunga del dovuto. Infine si pone un limite all'area massima oltre cui non dovrebbe più essere possibile progettare, in quanto si darebbe vita ad un inefficiente spreco di spazio.



48 – Definizione dei parametri di area e di testo

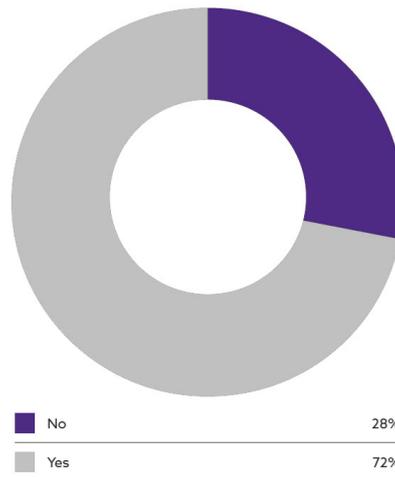
A questo punto, prima di procedere con il perfezionamento del prototipo, si ritiene di avere una base con sufficienti tipologie di parametri e dati per poter testare l'interoperabilità con *ArchiCAD*, che in questa prima fase è stata messa momentaneamente da parte per lasciare spazio alla parametrizzazione.

2.1.3 Interoperabilità orizzontale diretta

I test di interoperabilità rappresentano il punto cardine per comprendere se il metodo intrapreso finora possa effettivamente rappresentare una valida strategia ai fini del raggiungimento degli obiettivi della ricerca. Il sistema adottato, trattandosi di un passaggio tra due piattaforme BIM differenti, è quello di sfruttare un formato standard di interscambio. In particolare, come anticipato nel capitolo introduttivo, l'**IFC** è il formato promosso come tale, ed è pertanto quello che verrà impiegato in questa fase di interoperabilità orizzontale. Alla base di questo ci sono i dati riportati dal *National BIM Report* del 2018, in cui si afferma che più del 70% dei professionisti del settore usi questo formato per l'interscambio dei modelli BIM [64].

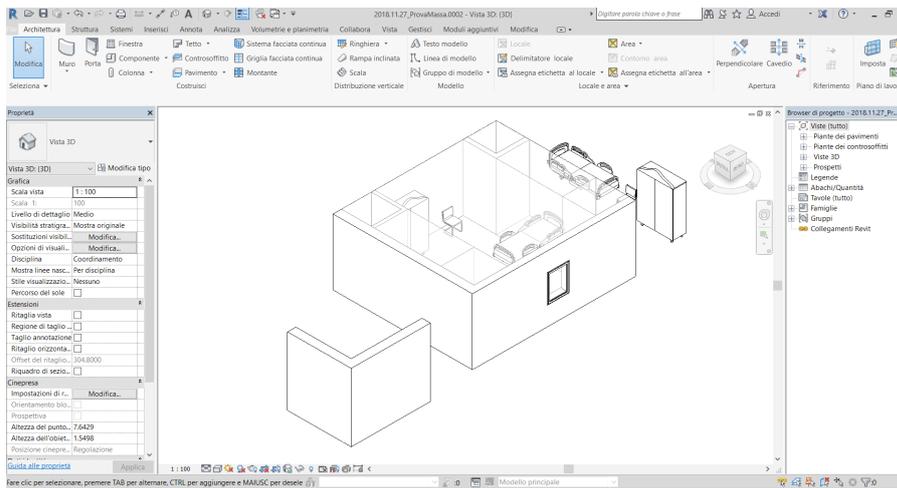
49 – Percentuale di utilizzo del formato IFC [64]

Do you use IFC on projects you've been involved with?



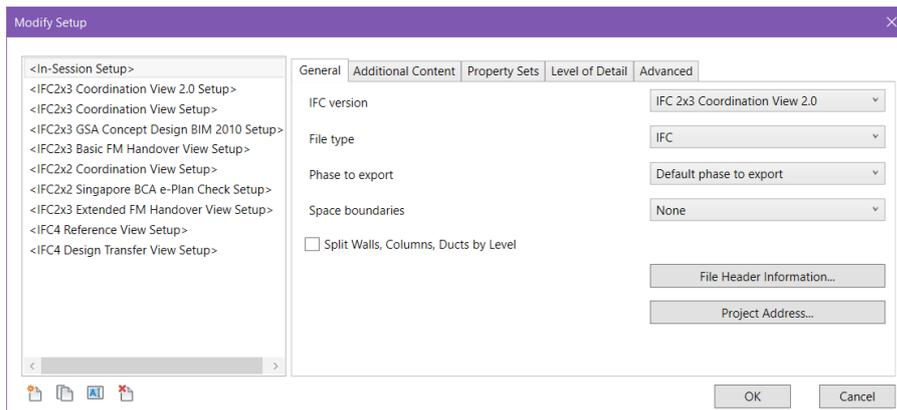
Per convalidare l'ipotesi e procedere con il metodo che prevede prima la parametrizzazione su *Revit* e poi l'importazione della degenza su *ArchiCAD*, si comincia con la **preparazione del modello** da esportare. La protagonista di questo test è sicuramente la famiglia di masse ottenuta dalla precedente elaborazione dei prototipi, ma si parla di modello perché si constata fin da subito che non è possibile né importare né esportare un IFC direttamente dal file di famiglia. Dovendo quindi creare un file di progetto, ci si può avvalere dell'occasione per inserirvi anche altri elementi, in modo da ottenere dei riscontri più completi per le eventuali future fasi di implementazione dell'oggetto-stanza. In particolare, oltre alla famiglia con i suoi parametri dinamici e di testo, si testano anche:

- Dei **muri** generici senza informazioni sui materiali che li compongono, alcuni modellati con riferimento dall'esterno, altri dall'interno
- Due tipologie di muri con una **stratigrafia** scelta fra le predefinite del software, sia vincolati alla superficie della massa che autonomi, e sia modellati con riferimento dall'esterno che dall'interno
- Una **finestra**, elemento "hosted", cioè che necessita di un altro (un muro in questo caso) per essere immesso
- Tre famiglie di **arredi** differenti, sia all'esterno, che all'interno delle masse: una già presente nel file, una da caricare fra quelle installate assieme al software, e una scaricata da internet come *.rfa* tramite la piattaforma online di *BIM object* [65].



50 – Modello Revit con gli elementi da testare per l'esportazione

Procedendo con l'esportazione ci si trova di fronte a differenti impostazioni da definire, tra cui la versione di IFC che si vuole creare. Il primo tentativo è subito rivolto alla tipologia maggiormente impiegata e diffusa, che è l'"**IFC 2x3 Coordination View 2.0**", aggiungendo anche la spunta all'esportazione delle proprietà di *Revit*, volendo mantenere tutti i parametri creati all'interno del file.



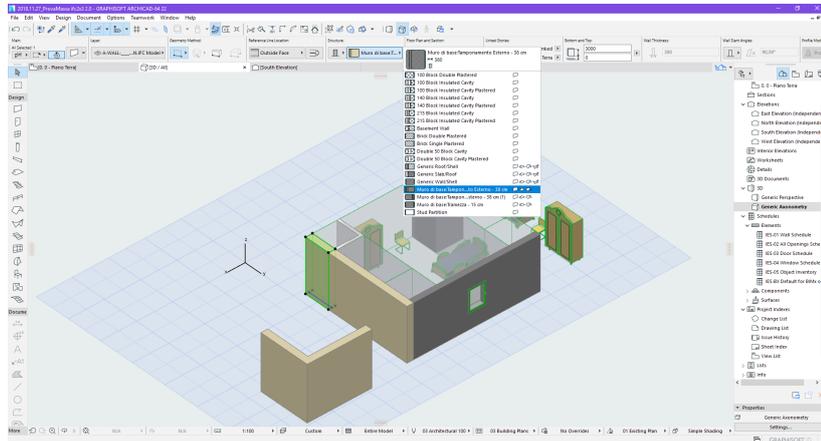
51 – Impostazioni disponibili per l'esportazione del formato IFC da Revit

Ora si procede importando il file così creato in *ArchiCAD*, dove a primo impatto si nota che ogni geometria nativa del file di origine è stata importata. Tutti i *layer* importati sono per impostazione di *default* bloccati, ed è quindi necessario agire sulle loro proprietà per poter editare i componenti. Approfondendo l'analisi, un elemento per volta, si riscontra che:

- I **muri** vengono riconosciuti come tali e la loro importazione fa sì che le stratigrafie impiegate su *Revit* vengano automaticamente aggiunte all'elenco di quelle già presenti su *ArchiCAD*, permettendo quindi di usufruire delle medesime proprietà. Vengono inoltre aggiunti ulteriori parametri nativi di *ArchiCAD*, anche se nel programma di origine non erano presenti, come il costo per volume, uguale però per tutte le categorie di muro impiegate. Non viene tuttavia mantenuta la differenziazione nella modellazione dei profili, in quanto anche quelli modellati dal filo interno vengono identificati con riferimento esterno. Questo comporta quindi che, in seguito a delle modifiche relative allo spessore degli stessi, il punto di "ancoraggio" del muro sarebbe diverso da quello che si era prefissato su *Re-*

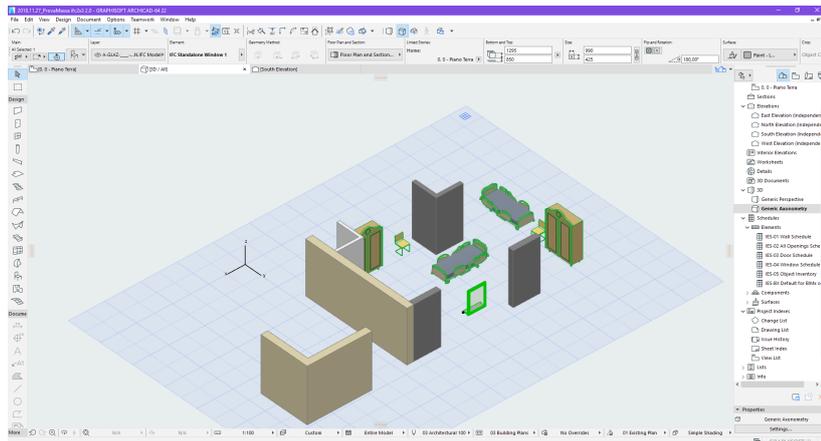
vit, e bisogna quindi correggere manualmente questa discrepanza. Inoltre i muri realizzati come perimetro dei servizi non vengono visualizzati in una vista bidimensionale di pianta se non spegnendo temporaneamente il *layer* della massa che li “ingloba”. Nonostante questo le strutture sono modificabili, sia nella loro stratigrafia che nella dimensione e posizionamento, con la stessa libertà di un muro realizzato direttamente su *ArchiCAD*.

52 – Modello importato su *ArchiCAD*: pannello delle stratigrafie dei muri



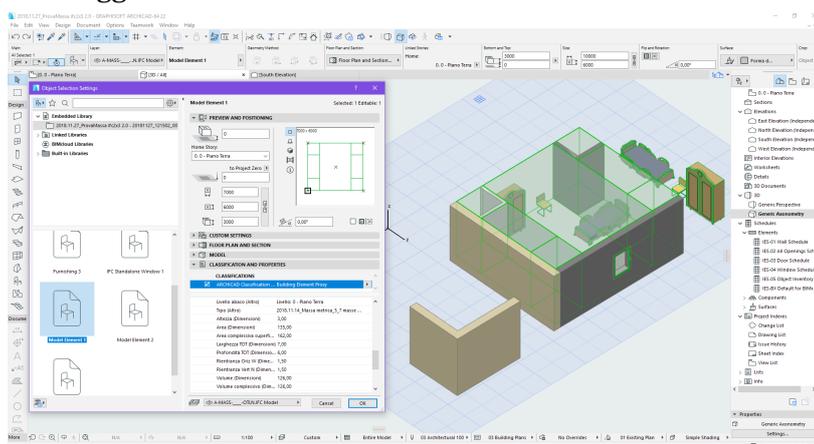
- La **finestra** non viene riconosciuta come serramento, bensì importata come oggetto e la sua funzione risulta pertanto “indefinita”. Inoltre essa perde la sua associazione alla struttura del muro che la ospita. Questo è facilmente verificabile eliminando il muro che la dovrebbe ospitare: normalmente tale operazione fa sì che venga cancellata anche la finestra stessa, mentre in questo caso invece essa risulta esistere indipendentemente da altre strutture.

53 - Modello importato su *ArchiCAD*: perdita dell'associazione al muro della finestra (visibile grazie all'eliminazione del muro a cui dovrebbe essere vincolata)



- Le **famiglie** di arredi e masse sono identificate come oggetti che, nel loro relativo pannello di proprietà, mostrano alcuni dei parametri dimensionali presenti su *Revit*. Tuttavia editando tali valori non viene effettivamente applicata nessuna modifica e gli elementi, che risultano quindi come dei “blocchi”. Non appaiono grandi differenze, né a livello di proprietà né di editabilità, fra le diverse tipologie di famiglie caricate nel file *Revit*. Ne conviene quindi che i parametri dimensionali si siano trasformati in semplice testo e che l'unica modifica applicabile riguar-

di il punto di posizionamento dell'oggetto, che può quindi essere spostato o copiato partendo da un definito punto di ancoraggio.



54 - Modello importato su ArchiCAD: perdita di editabilità dei parametri dinamici

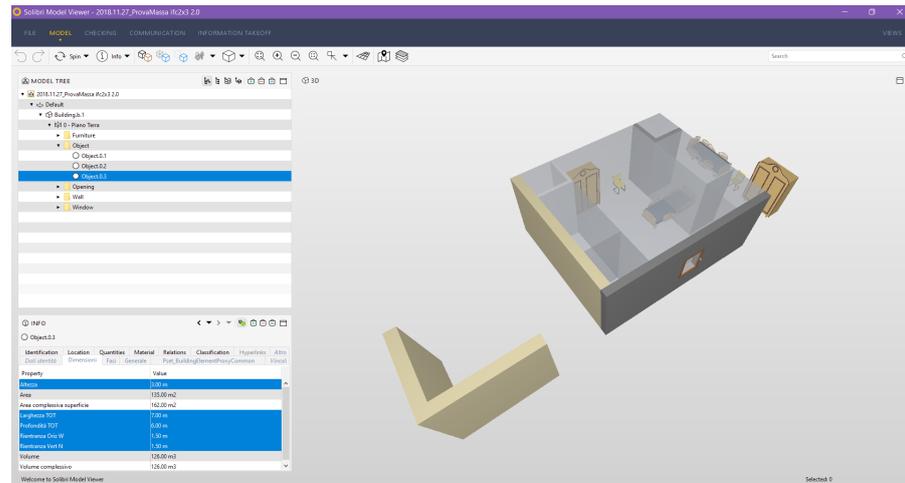
Provando ad esportare da *Revit* in altri formati IFC, come l'.*ifcxml* e l'.*ifczip*, si ottengono i medesimi risultati. Perfino impiegando lo schema IFC4, di più recente sviluppo, gli esiti non cambiano, anzi, in quest'ultimo caso, oltre a non importare i parametri dinamici, non è presente nemmeno l'intera geometria iniziale.

Per verificare se la perdita di dati avvenga effettivamente in fase di esportazione o durante quella di importazione è possibile utilizzare un visualizzatore di modelli che interagisca direttamente con il file IFC, prima che esso venga importato in un secondo software. A questo scopo è stato impiegato **Solibri Model Viewer**, grazie al quale si ispezionano gli elementi che presentano informazioni errate o mancanti e dal quale si evince che entrambe le operazioni contribuiscono alla suddetta perdita.

Da un lato infatti si può notare che la finestra, con l'esportazione in IFC è ancora riconosciuta come tale, scomponendola inoltre della geometria di apertura nel muro che viene generata automaticamente nel software di modellazione con il suo posizionamento all'interno della struttura. Inoltre alcune delle famiglie caricate vengono riconosciute qui come oggetti, altre come arredo. Questo è dovuto tuttavia al fatto che su *Revit* le famiglie presenti con l'installazione del software possiedono già la corretta categoria di arredo impostata sotto cui classificarsi, mentre a quella scaricata da internet bisogna editarla manualmente, operazione possibile, ma che in questo caso non è stata effettuata. La famiglia di masse parametrizzata, per lo stesso motivo di quest'ultima citata, risulta come un oggetto alla pari dell'arredo scaricato da internet. È però possibile vedere i parametri impostati su *Revit* con la nomenclatura corrispondente a quella creata ed i loro valori dimensionali o il testo presente al momento della generazione dell'IFC. Dall'altro lato però si notano già alcune discrepanze con i dati generati automaticamente, come quello dell'area, che riporta un valore errato. Inoltre grazie a *Solibri*, in quanto software visualizzatore, è possibile vedere se un parametro è stato esportato nel file IFC, ma non si è in grado di capire se esso sia mantenuto come variabile dinamica o se sia diventato di semplice testo. Per quanto riguarda infine la mo-

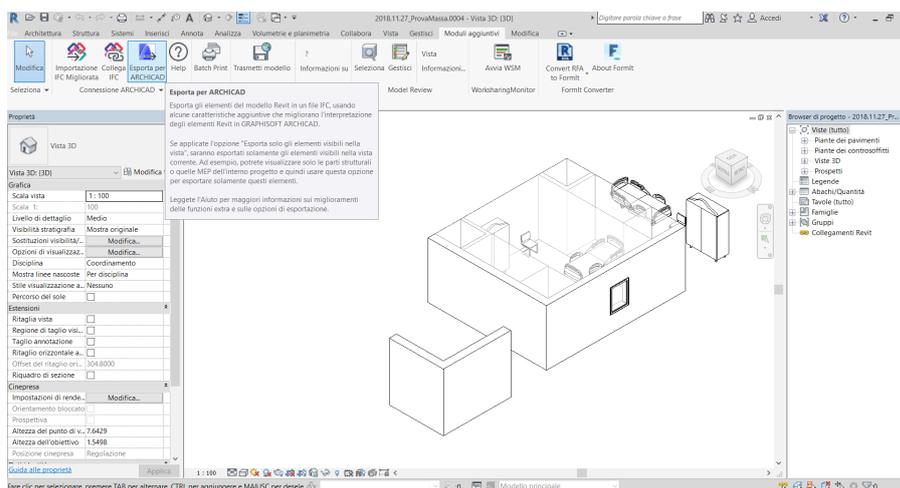
dalità di modellazione dei muri, non compare nessun dato da cui si desuma se essi siano stati modellati con riferimento interno o esterno, quindi questo tipo di informazione si perde già con l'esportazione.

55 - Modello visualizzato su Solibri Model Viewer: mantenimento dei parametri creati su Revit



Per testare la veridicità delle informazioni ricavate dal software di visualizzazione dell'IFC si può provare ad importare ora il medesimo file, aperto prima con *ArchiCAD* e poi con *Solibri*, in *Revit* stesso. Questa verifica comporta tuttavia delle nuove variabili dovute all'operazione di importazione che, sebbene avvenga nello stesso programma in cui il file originario è stato creato, comporta comunque delle criticità. È infatti possibile notare tutte le discrepanze già individuate mediante il software di visualizzazione intermedio, ed ulteriori differenti sia da queste ultime che da quelle identificate su *ArchiCAD*. Tra queste è importante sottolineare che tutte le tipologie di famiglie caricate, pur mantenendo la categoria iniziale di appartenenza, perdono sia i parametri dinamici che la possibilità di essere modificate. Cercando infatti di accedere all'editor della famiglia, al suo posto si apre l'interfaccia di modifica dinamica come se fossero delle masse. Non fa eccezione la finestra che, oltre a perdere editabilità, risulta priva dell'*host* che la vincolava alla struttura muraria.

A questo punto è possibile fare ancora delle prove di esportazione da *Revit* ad *ArchiCAD* grazie all'utilizzo di *plug-in* gratuiti. Il primo, sviluppato da *Graphisoft*, comporta l'aggiunta del modulo di "**Connessione ArchiCAD**" alla piattaforma di *Autodesk*. Una delle funzioni di tale *add-in* dichiara proprio di poter esportare gli elementi del modello di *Revit* in file IFC specificamente migliorati per l'utilizzo in *ArchiCAD* [66].



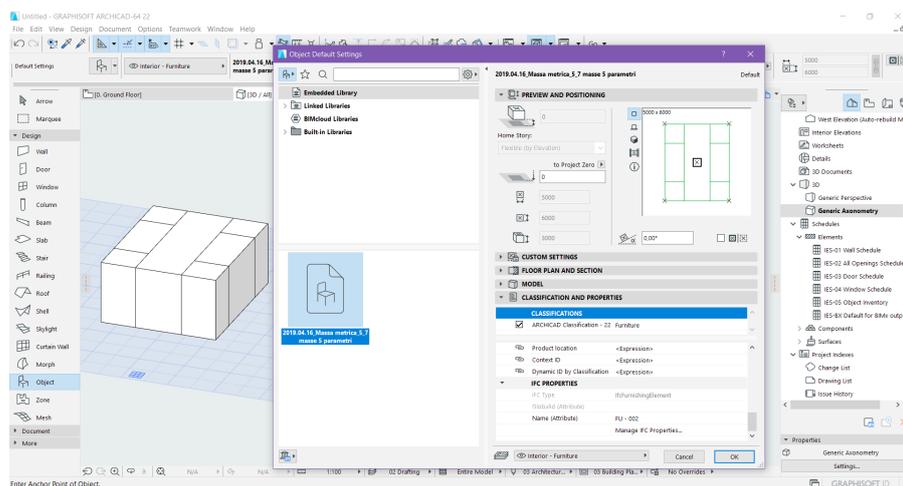
56 – Add-in “Connessione ArchiCAD” fra i moduli aggiuntivi di Revit

Con tale premessa si procede quindi al test di interoperabilità, seguendo le indicazioni fornite dalla breve guida presente sul sito di *Graphisoft*, e consultabile anche mediante lo strumento di aiuto del *plug-in* stesso. Si nota subito una voluta limitazione delle opzioni, in quanto si escludono sia il formato *.ifcxml*, che lo schema IFC4, in quanto si considera ancora l'*IFC2x3 Coordination View 2.0* quello standard, certificato e raccomandato a livello mondiale [67].

Una volta importato il file ottenuto mediante l'*add-in* su *ArchiCAD* non si notano in realtà particolari differenze rispetto al normale processo applicato senza l'impiego del suddetto *plug-in*. Si può quindi pensare che tale modulo aggiuntivo possa forse fare la differenza su grandi progetti con tanti elementi da esportare, ma non su piccoli dettagli come nell'analisi in questione.

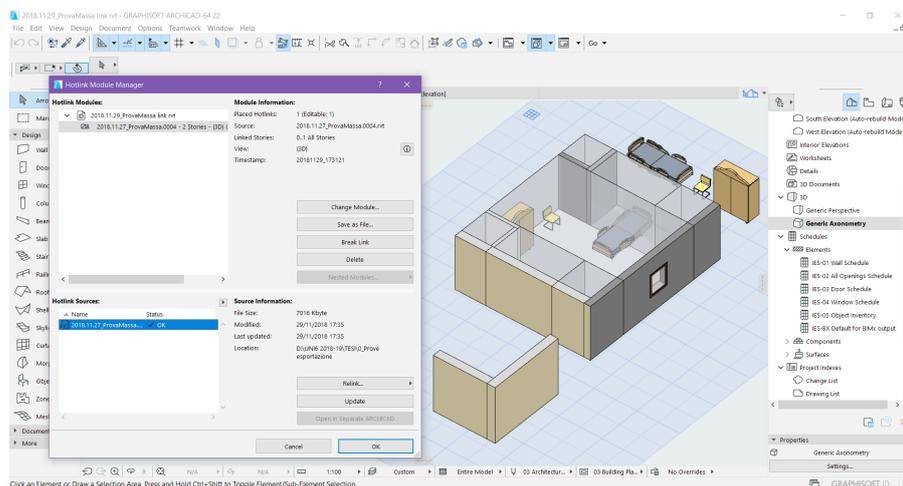
Il secondo *plug-in* che si intende usare è “**RFA & RVT Geometry Exchange**”, sviluppato da BIM6x. Questo *add-in* ha due funzionalità che gli permettono di far interagire un file creato su *Revit* con *ArchiCAD*. La prima consente di trasformare una famiglia di *Revit* in formato *.rfa* in un oggetto GDL di libreria di progetto in un modello *ArchiCAD*. Importando la famiglia di masse parametrizzata creata in precedenza su *Revit*, l'oggetto risulta privo di parametri dinamici e di possibilità di modifica. Questo ovviamente conferma quanto già appurato dai produttori stessi del *plug-in*, che assicurano che i parametri creati su *Revit* possano essere importati, e di fatto compaiono nell'elenco, ma non editati [68]. Questo può essere utile, secondo gli sviluppatori, a ricavare informazioni da enumerare negli abachi, ma questo consiste solo in uno degli obiettivi che si vogliono raggiungere con questa ricerca.

57 – Importazione della famiglia di masse parametrica creata su Revit in ArchiCAD, mediante il plug-in “RFA & RVT Geometry Exchange”



La seconda funzionalità di questo stesso *plug-in* permette invece di collegare un file *.rvt* di *Revit* attraverso un *link* direttamente ad *ArchiCAD*. Questo prevede che la famiglia di masse venga dapprima caricata e completata con muri e tutte le strutture secondarie e arredi che devono comporre la degenza su *Revit*, dopodiché collegare tale modello ad uno di *ArchiCAD*. Mediante questa metodologia il file *.rvt linkato* è un blocco unico di sola visualizzazione, in quanto non è possibile modificarlo direttamente su *ArchiCAD*, ma solo dal programma originario in cui è stato creato. Nonostante questo, è possibile tenere costantemente aggiornato il modello o addirittura lavorare con entrambi i programmi aperti per vederne le modifiche in tempo reale. Questo rappresenta un buon compromesso per poter lavorare con i due software, tuttavia non è ancora abbastanza per l'esito atteso da questa ricerca.

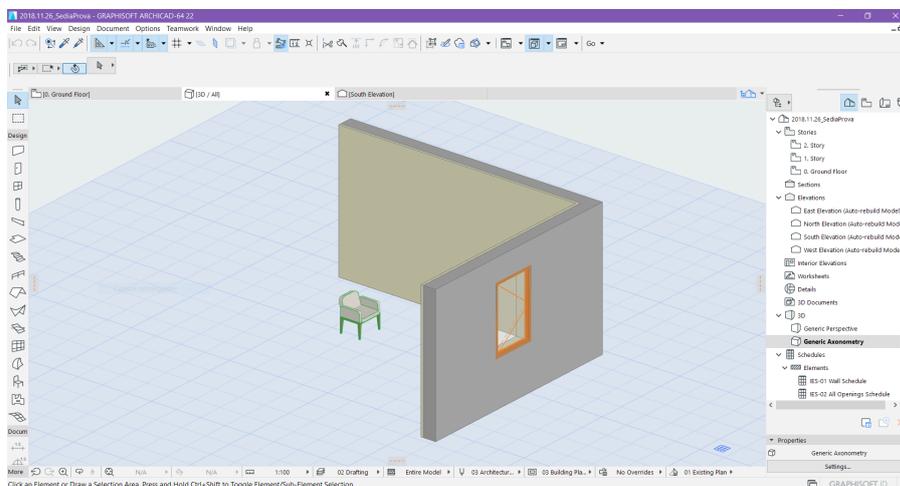
58 – Collegamento di un file *.rvt* all'interno di *ArchiCAD*



I risultati ottenuti con i test effettuati finora non possono essere considerati soddisfacenti per poter proseguire la ricerca basandosi sugli stessi presupposti iniziali. Si può quindi provare a verificare se esista un **flusso di interoperabilità preferenziale** inverso a quello sperimentato. Se questa nuova ipotesi portasse a risultati migliori, si potrebbe prendere in considerazione l'idea di impiegare ulteriori risorse per parametrizzare l'oggetto-stanza prima su *ArchiCAD* ed importarlo successivamente su *Revit*.

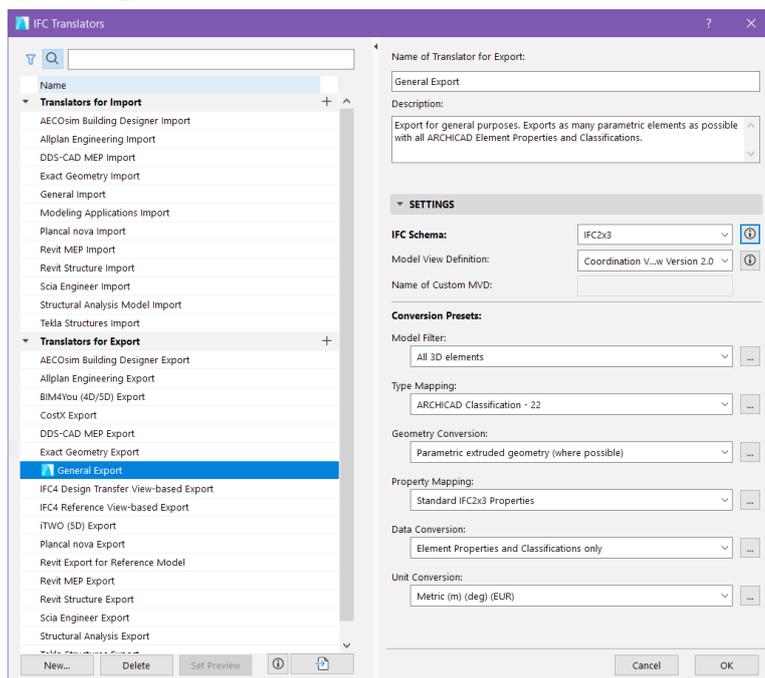
Si procede seguendo lo stesso *iter* usato in precedenza, partendo

quindi dalla preparazione di un file su *ArchiCAD* che abbia pochi elementi con dei parametri che si desiderano importare su *Revit*. In questo caso, data la poca abilità nel maneggiare i parametri *ex novo*, il modello contiene semplicemente un oggetto da libreria, due muri ed una finestra.



59 - Modello ArchiCAD con gli elementi da testare per l'esportazione

Anche in questo caso l'operazione di esportazione porta a scegliere fra diverse impostazioni disponibili. Come *Revit*, *ArchiCAD* permette di scegliere tra i tre possibili formati di IFC (.ifc, .ifczip e .ifcxml), nonché lo schema, ma, visti i risultati del precedente test, in questo caso si decide di usare unicamente l'IFC2x3, con definizione "coordination view version 2.0" in quanto standard associato. Sono inoltre presenti ulteriori impostazioni volte ad ottimizzare il tipo di esportazione in base al software a cui essa è destinata, tra i quali è presente anche *Revit*. Tra le impostazioni dell'"**IFC Translator**" è possibile prendere visione, creare, importare (in formato .tpl, .pln, .pla, .xml) o modificare le variabili di conversione dei dati, tra le quali poter scegliere cosa esportare e anche la relativa unità di misura. Nei test illustrati di seguito verranno impiegate le impostazioni predefinite per tre diverse tipologie di esportazione.

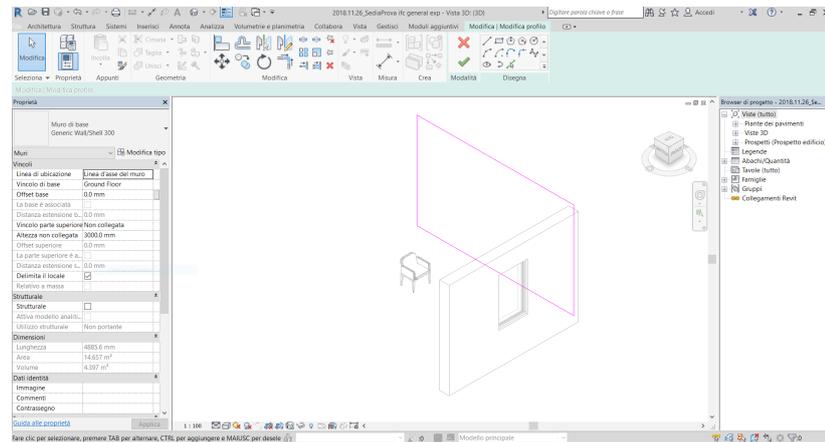


60 - Impostazioni disponibili per l'esportazione/importazione del formato IFC in ArchiCAD

1. Esportazione con caratteristiche generali: **“general export”**.

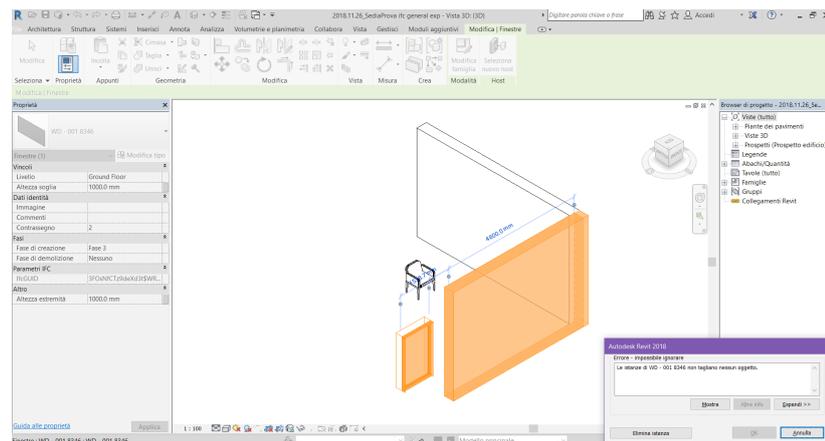
- Importando il file IFC con questa impostazione la stratigrafia dei **muri** viene aggiunta a quelle già presenti in archivio, come accade con l'importazione dell'IFC generato da *Revit* in *ArchiCAD*. Nonostante possiedano ancora i loro parametri di testo, ma nessuno parametrico, i muri sono completamente editabili, sia in dimensioni che in profilo e stratigrafia. Essi appaiono inizialmente disgiunti fra loro, ma alla prima modifica vengono nuovamente raccordati.

61 - Modello importato su Revit: modalità di modifica dei profili dei muri

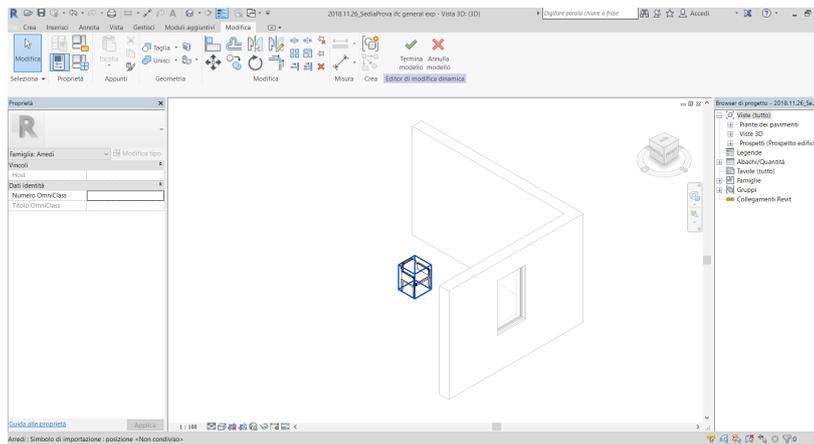


- La **finestra** viene riconosciuta come elemento di serramento, con il suo vincolo di ospite nel muro. Viene anche riconosciuta come una famiglia, nonostante *ArchiCAD* non possieda questa nomenclatura, ed è possibile accedere all'editor di famiglie come se fosse un file *.rfa*. Grazie a questo è quindi possibile apportarvi delle modifiche, dopo averla salvata come file di famiglia esterno, e ricaricarla nel modello. Come per i muri, vengono importati alcuni parametri di testo ed un parametro editabile per modificarne il posizionamento lungo l'asse Z.

62 - Modello importato su Revit: verifica del mantenimento dell'host del serramento



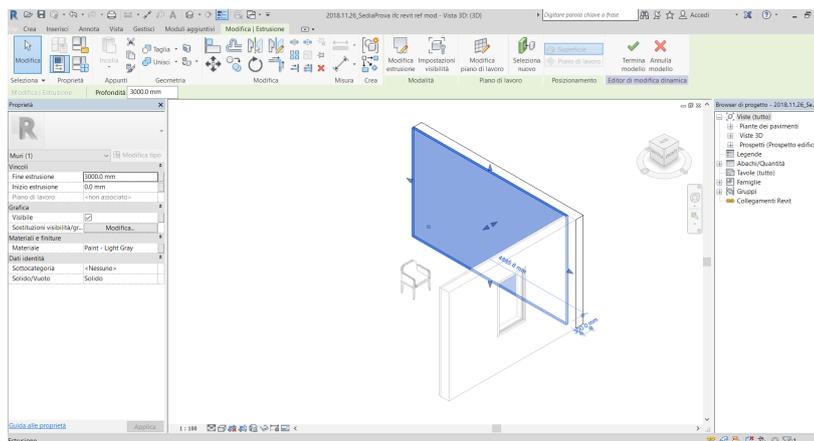
- L'elemento d'**arredo** viene anch'esso identificato come tale, ma è ora assimilato ad una famiglia locale. Anche accedendo alla modifica dinamica risulta comunque come un blocco sulla cui geometria è impossibile intervenire, nemmeno provando ad esploderne i componenti. Per quanto riguarda i parametri, ne vengono riportati pochi fra quelli di testo e nessuno fra quelli di tipo dinamico.



63 - Modello importato su Revit: impossibilità di modifiche sull'elemento d'arredo

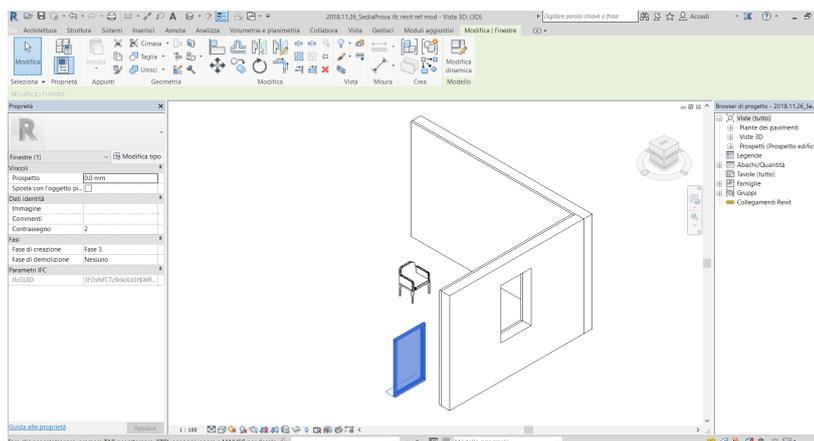
2. Esportazione ottimizzata per un generico modello di Revit: “*Revit export for reference model*”.

- I **muri** vengono apparentemente riconosciuti come tali con questa impostazione, ma, anche se la loro geometria rispecchia il modello esportato, le informazioni non sono altrettanto corrette. Ogni stratigrafia che compone i muri viene infatti visualizzata come un elemento a se stante e, invece di accedere alla modifica del profilo del muro, con il doppio *click* si apre l'interfaccia di modifica di famiglia locale in cui poter editare la morfologia di ogni stratigrafia.



64 - Modello importato su Revit: stratigrafie dei muri come masse separate

- La **finestra** rientra nella categoria corretta, ma non mantiene le stesse caratteristiche del test precedente, risultando quindi priva di ogni parametricità.

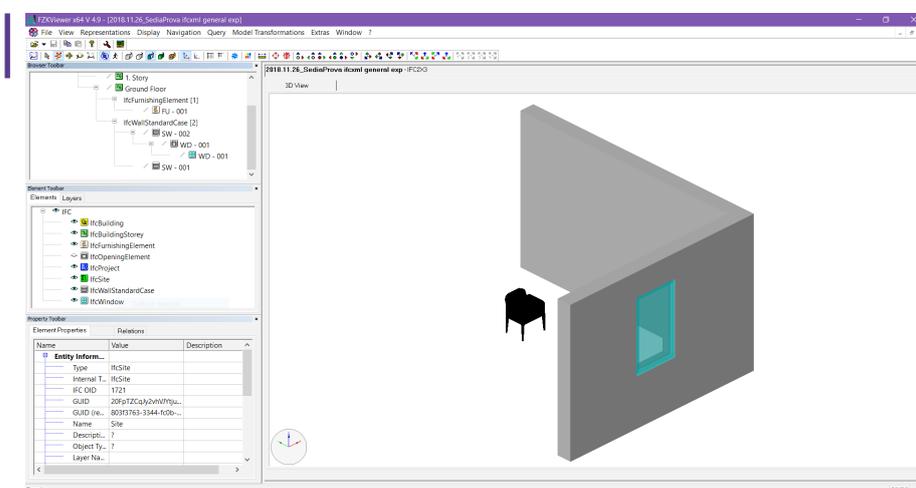


65 - Modello importato su Revit: perdita del parametro “host” del serramento

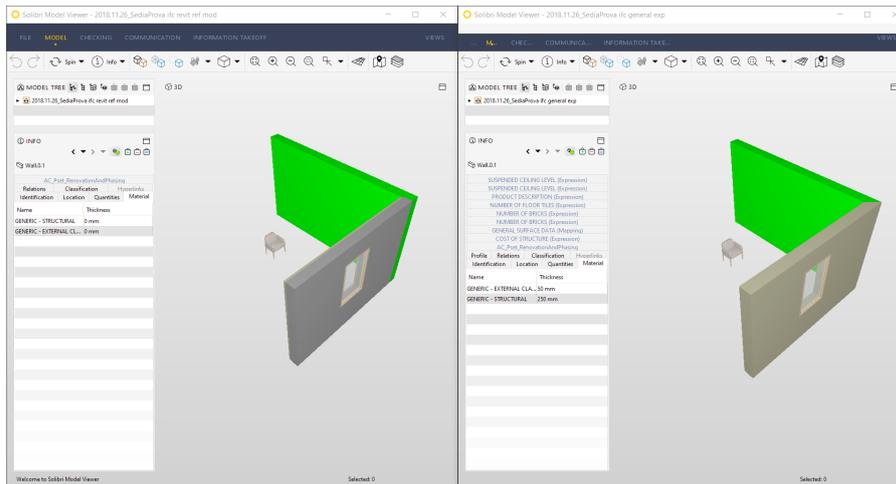
- L'**arredo**, come nel caso precedente, viene inserito nella categoria corretta, ma risulta privo di ogni parametro e, quindi, di possibilità di essere modificato.
3. Esportazione specifica per un modello strutturale di *Revit*: “**Revit structure export**”.
- I **muri** e la **finestra** mantengono le stesse caratteristiche del test di esportazione generale, mentre l'**arredo** non viene importato per niente.

L'ultimo test voleva solo essere dimostrativo del fatto che è possibile fare un'esportazione per un settore specifico, per quanto il modello impiegato sia molto semplificato e generico. In questo caso è stato testato quello strutturale, ma è disponibile anche quello MEP (*Mechanical Electrical Plumbing*), per ognuno dei quali ci sono delle varianti. Ad ogni modo tutti e tre questi test hanno mostrato, per ogni tipologia di esportazione, uguali risultati per il formato *.ifc* e *.ifczip*, che rappresenta di fatto solo una compressione del primo, ma ogni file *.ifcxml* si è mostrato sempre vuoto una volta importato su *Revit*. Dopo alcune ricerche questo è risultato essere un problema comune fra gli utenti, che hanno segnalato infatti sul sito di *Autodesk* l'impossibilità di importare o collegare questo tipo di formato [69]. È di fatto dimostrabile che si tratti di un errore di importazione in quanto usando un software di visualizzazione come *FZKViewer*, che contrariamente a *Solibri* apre anche i file *.ifcxml*, tutti gli elementi vengono correttamente visualizzati dopo l'esportazione.

66 – File *.ifcxml* visualizzato su *FZKViewer*



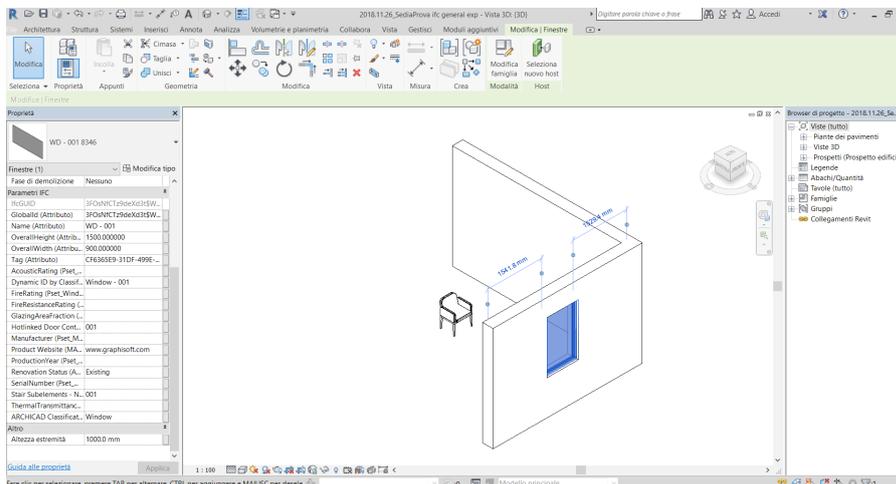
Impiegando nuovamente **Solibri Model Viewer** per visualizzare i file IFC prima di essere importati su *Revit* non si notano particolari perdite di dati dal passaggio di esportazione a quello di importazione. Da questo si evince quindi che le maggiori perdite avvengano già in fase di esportazione. Si può inoltre confermare, come è risultato dai test appena elencati, che l'impostazione volta ad ottimizzare l'esportazione per *Revit* perda effettivamente più informazioni rispetto a quella generale. Ad esempio la stratigrafia del muro nel primo caso, sebbene sia visibile graficamente, viene elencata, ma con gli spessori nulli, mentre nel secondo non si distingue nella visualizzazione, ma mantiene questi dati.



67 – File IFC a confronto in Solibri Model Viewer: a sinistra il file esportato come “Revit export for reference model”, che non mantiene gli spessori della stratigrafia; a destra il file esportato come “general export”, che mantiene i dati della stratigrafia.

Come per *Revit*, si può testare se *ArchiCAD* sia in grado di perdere meno informazioni possibili aprendo un IFC creato dal software stesso. In questo caso si ottengono risultati migliori, seppur rispecchiando quelli già discussi nell’esportazione su *Revit*. I muri sono infatti editabili come se fossero stati creati direttamente sul file, e la finestra mantiene il suo *host*, proprio come accade con l’esportazione nella seconda piattaforma BIM. Tuttavia, entrambi gli oggetti caricati, la finestra stessa e la sedia, vengono ora considerati come oggetti di libreria di progetto, quindi interna, e per apportarvi modifiche è necessario prima salvare gli oggetti in una cartella. Non è infatti possibile, una volta eseguita tale operazione, procedere con la modifica degli elementi, in quanto l’*editing* risulta comunque bloccato e le modifiche apportabili non riguardano la geometria degli oggetti, ma solo la loro posizione.

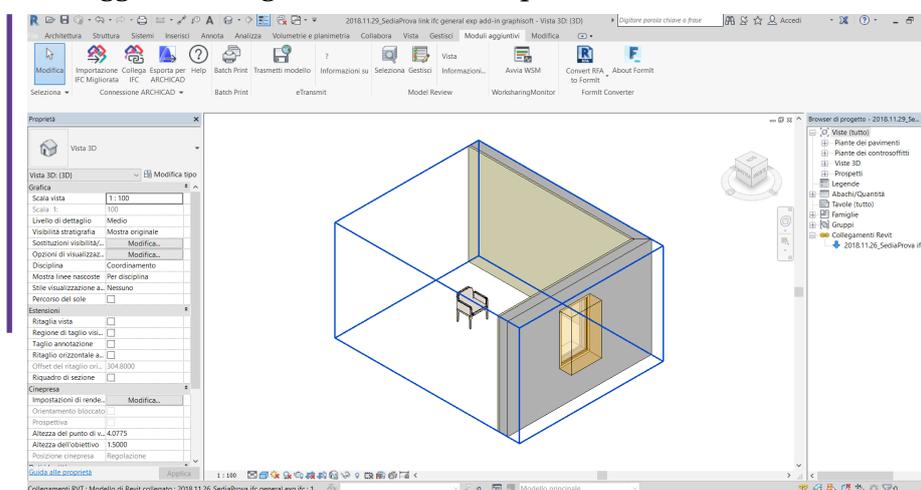
Il *plug-in* “Connessione *ArchiCAD*”, utilizzato prima per la sola funzione di **ottimizzazione dell’IFC** da esportare da *Revit*, ha due ulteriori funzioni che permettono di operare col flusso di interoperabilità inverso che si sta ora analizzando. In particolare la prima permette di importare i modelli IFC in *Revit* migliorandone le prestazioni. Ciò consente di importare un maggior numero di parametri di testo in *Revit*, sotto la voce di “Parametri IFC”, che però non mantengono nessuna relazione fra loro e spesso raddoppiano informazioni già presenti.



68 – File IFC importato in Revit mediante la specifica funzione dell’add-in “Connessione *ArchiCAD*”.

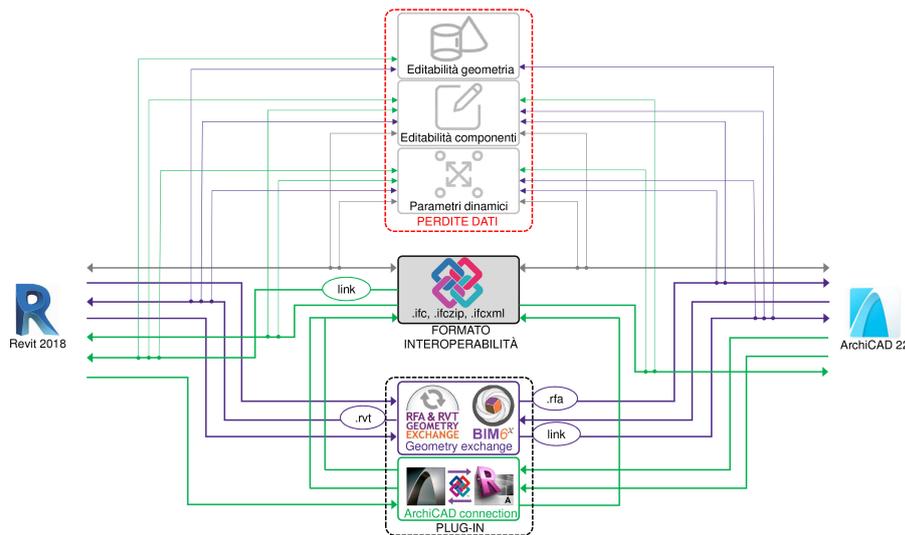
La seconda ulteriore funzione dell'*add-in* permette invece di unire un file IFC, attraverso un collegamento per **link**, al modello *Revit*. Come spiegato anche nella guida, questo procedimento non permette di editare il modello collegato e quindi, per quanto possa essere utile per una visualizzazione del file IFC, le sue funzioni risultano limitate. Pertanto, questa operazione è più vicina ad un'importazione che ad una condivisione per *link*, in quanto il modello collegato appare come un blocco non modificabile né scomponibile nelle sue parti. È infatti impossibile agire perfino sulla visibilità degli elementi che lo compongono, quindi volumi come quello di apertura creato per sottrazione nell'inserimento della finestra nel muro vengono rappresentati nonostante non abbiano significato nel modello. Inoltre un modello collegato in formato IFC non può essere direttamente modificato, e di conseguenza aggiornato nel software in cui è collegato, perdendo così i vantaggi di una regolare condivisione per *link*.

69 – File IFC collegato via link al modello *Revit*: importazione di sola lettura in cui tutti i volumi sono visibili anche se creati dal software come “oggetti di costruzione” di altri elementi (es. foro della finestra).



L'ultimo test di interoperabilità consiste nell'impiegare la terza funzionalità del plug-in “*RFA & RVT Geometry Exchange*”, che consente di salvare un modello generato su *ArchiCAD* direttamente nel formato *Revit .rvt*. Con questo espediente la geometria viene completamente importata ed ogni elemento mantiene la sua corretta categoria, ma si perde ogni editabilità. Ogni elemento è di fatto selezionabile singolarmente, ma non possiede parametri di alcun tipo né ha la possibilità di accedere alla modifica dinamica o all'editor di famiglia. La finestra perde il suo *host*, l'arredo continua ad essere un blocco ed i muri, per quanto categorizzati come tali, non possono ospitare nuovi serramenti.

Lo schema sottostante è volto a riassumere il processo d'interoperabilità che intercorre tra le due piattaforme BIM, integrando sia il metodo “tradizionale” sia l'impiego dei *plug-in* analizzati. Come si può notare dalle consistenti perdite di parametri ed editabilità, non sembra esistere un flusso d'interoperabilità preferenziale tra *Revit* ed *ArchiCAD*.



70 – Processo di interoperabilità messo in atto tra Revit ed ArchiCAD, con le rispettive perdite di dati

Dai test svolti si desume quindi che, con gli strumenti disponibili, un'interoperabilità orizzontale diretta fra due piattaforme BIM non è possibile senza una perdita di dati. Si deve quindi trovare un metodo che comunichi con entrambi i programmi scelti, con la minor perdita di dati possibile, per riuscire ad ottenere una degenza parametrica che rispetti gli obiettivi della tesi.

2.1.4 Software parametrico “intermediario”: Grasshopper

Il primo approccio metodologico sperimentato non ha avuto i risultati attesi ed è stato perciò necessario tornare al primo punto della fase invece che procedere a quello successivo. Pertanto, siccome l'interoperabilità fra le due piattaforme BIM prese in analisi non ha ancora raggiunto un livello sufficientemente elevato per soddisfare gli obiettivi posti, si possono cercare **programmi “intermediari”** per evitare il più possibile la perdita di dati che avviene invece con lo scambio diretto. Si può quindi approfondire l'ipotesi di non operare subito con delle piattaforme *BIM-based*, ma di fare metaforicamente un passo indietro per ragionare prima di tutto sull'obiettivo e secondariamente sul metodo. Se lo scopo è quello di realizzare un oggetto-degenza parametrizzato che abbia da un lato la possibilità di essere modificato facilmente e velocemente, limitando possibili incongruenze, e dall'altro contenga tutte le informazioni necessarie per la sua gestione, non è detto che la metodologia in fase di modellazione possa limitarsi al solo uso del BIM. Per quanto sia già appurato infatti che quest'ultimo ben si presti ai requisiti che si basano sui dati e sulla loro gestione, è risaputo anche che spesso questa qualità sia ostacolata da una difficoltà nella modellazione e nella creazione e condivisione dei parametri, come è stato anche parzialmente dimostrato precedentemente. I software che viceversa riescono al meglio a realizzare modelli parametrici sono quelli di **programmazione visuale**, dedicati al “*generative modeling*”, che non sono invece solitamente in grado di gestire il genere di informazioni che una degenza deve possedere per la sua manutenzio-

ne e gestione nel tempo. Si ipotizza pertanto che una combinazione di queste due metodologie possa costituire una carta vincente per il raggiungimento degli obiettivi desiderati.

Cruciale ai fini di questa fase della ricerca sono state ricerche reperite dalla letteratura internazionale, grazie alle quali è stato possibile acquisire familiarità con un metodo che non sfruttasse unicamente le potenzialità del BIM, ma anche quelle di un **processo più ampio ed integrato**. Infatti, grazie alla collaborazione tra *McNeel & Associates* e *Graphisoft*, è nato un *plug-in* mediante il quale è possibile far connettere *Grasshopper* con *ArchiCAD*, integrando la logica della progettazione algoritmica e fornendo un più elevato livello di parametrizzazione alla piattaforma BIM. Questo nuovo approccio, che si appoggia anche sul **visual editing**, è di entità differente rispetto al precedente di sola concezione BIM. Se con un tradizionale programma di modellazione infatti si ha un *effort* basso in relazione ai risultati ottenuti graficamente, bisogna ora tenere conto che con la modellazione algoritmica la situazione è opposta. Nonostante la semplificazione del metodo grafico fornito da *Grasshopper*, è necessario in realtà impiegare molte risorse per la concezione dei passaggi logico-matematici di ogni elemento, ed è di fatto per questo motivo che è considerato come approccio secondario rispetto al primo testato.

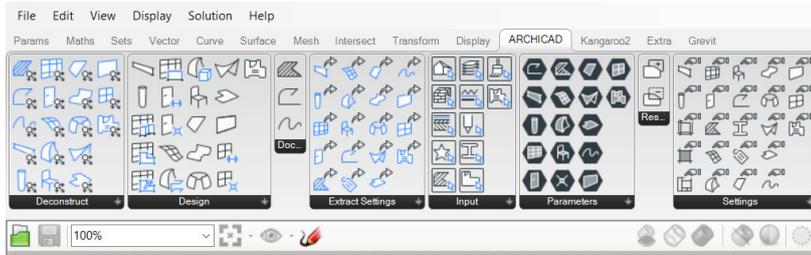
Il programma consente di elaborare **algoritmi**, ovvero di restituire degli *output* partendo da una sequenza logica e finita di istruzioni elementari fornite dall'utente sotto forma di *input*. La piattaforma di *Grasshopper* è a sua volta pensata come *plug-in* di *Rhinoceros* che, in quanto tale, opera in parallelo al software di modellazione 3D, che però in questa ricerca non verrà preso in considerazione. L'interfaccia utente è molto elementare ed è composta da un pannello, che raggruppa **parametri e componenti**, ed il "**canvas**", ovvero l'area di lavoro su cui operare. I primi raccolgono i dati, che possono essere numeri, colori, geometrie o altro ancora, e sono facilmente riconoscibili per il bordo esagonale intorno all'icona, mentre i componenti eseguono le azioni che restituiscono i parametri [70].

Approfondendo le ricerche è stata inoltre riscontrata la possibilità di usufruire di ulteriori *plug-in* gratuiti, disponibili sulla piattaforma di applicativi per *Rhinoceros* e *Grasshopper* "**food4Rhino**" [71], che permettono il dialogo anche con *Revit*, seconda piattaforma BIM in analisi nella presente ricerca.

Si è quindi scelto di analizzare alcuni *plug-in*, scelti in base alle funzionalità ed alla disponibilità online rispetto alla versione dei programmi BIM impiegati. I requisiti che si ritiene essi debbano rispettare sono, come nei test precedenti, la bidirezionalità del flusso di lavoro e l'**esportazione di geometrie, informazioni e parametri**. Alcuni, come *Lyrebird* (di Lmnts) [72], *Chameleon* (di Hiroshi Jacobs) [73] o *Mantis Shrimp* (di Konrad K Sobon) [74], per quanto potenzialmente interessanti, sono stati scartati perché non più aggiornati negli ultimi anni o non disponibili per il *download*. I *plug-in* esaminati per questo secondo tentativo di modellazione di un prototipo semplificato al fine di testarne la condivisione dei dati fra *Revit* ed *ArchiCAD* sono

tre:

- **Grasshopper - ArchiCAD Live Connection** (ver 2.2) [75]
Plug-in sviluppato da *Graphisoft*, in collaborazione con *McNeel & Associates*, che permette una connessione diretta ed in tempo reale tra *Grasshopper* ed *ArchiCAD*. Alla base del suo funzionamento c'è la possibilità di operare direttamente dal software di *Visual Programming Language*, ma impiegando anche strumenti ed impostazioni caratteristiche della piattaforma BIM.



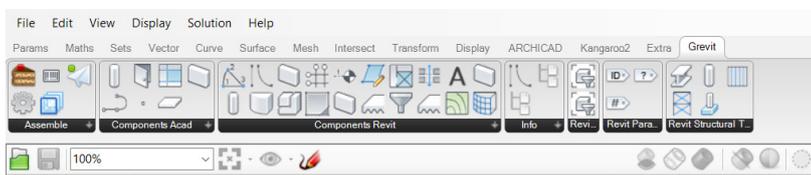
71 - Barra dei componenti del plug-in *Grasshopper - ArchiCAD Live Connection* su *Grasshopper*

- **Hummingbird** (ver. *Rhino 6* e *Revit 2018-2020*) [76]
Plug-in sviluppato da *WhiteFeet Software* grazie al quale *Grasshopper* e *Revit* si scambiano dati indirettamente attraverso l'esportazione e l'importazione di *Comma-Separated Values* (CSV), un file di testo per la rappresentazione di informazioni a struttura tabellare.



72 - Barra dei componenti del plug-in *Hummingbird*

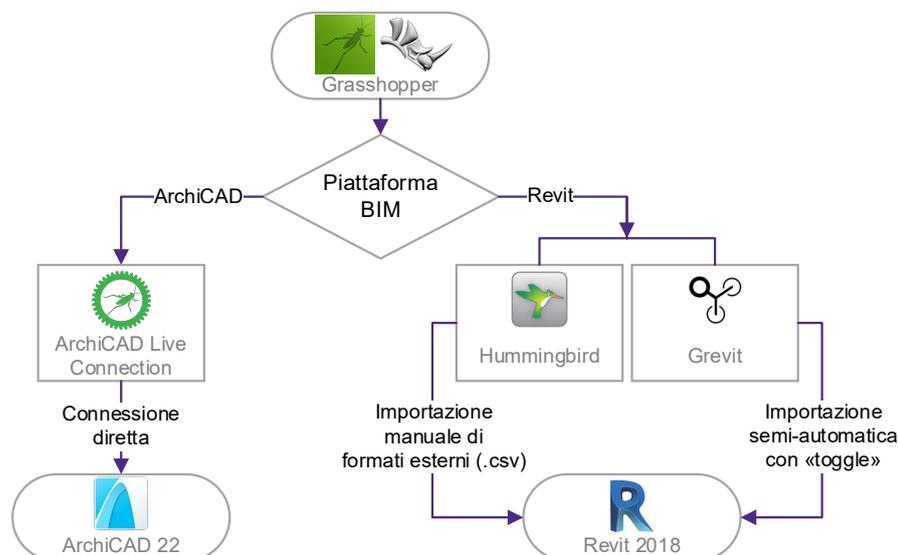
- **Grevit - Grasshopper Native BIM** (ver. *Revit 2018*) [77]
Plug-in sviluppato da *Moethu*, che permette di far dialogare *Grasshopper* e *Revit* mediante lo stesso principio di quello sviluppato da *Graphisoft*, ma senza la possibilità di avere una connessione in tempo reale. Per l'invio dei dati da una piattaforma all'altra è infatti necessaria un'operazione manuale da parte dell'utente.



73 - Barra dei componenti del plug-in *Grevit - Grasshopper Native BIM*

L'approccio con cui ora si intende proseguire prevede quindi la modellazione parametrica del prototipo di degenza partendo da *Grasshopper*, in quanto programma di *visual editing*, grazie anche ai *plug-in* sopra elencati che lo implementano dei componenti di *ArchiCAD* e *Revit*, ed i relativi test di interoperabilità con le piattaforme BIM. Nel diagramma di flusso sottostante è schematizzato il *work-flow* con cui si intende procedere, in base ai *plug-in* selezionati.

74 – Diagramma di flusso della modellazione parametrica in base ai software e plug-in selezionati



2.1.5 Parametrizzazione ed interoperabilità orizzontale indiretta

Lo scopo della modellazione del prototipo di degenza basato sul *visual editing* è quello di ottenere un **unico file** su *Grasshopper* (.gh), grazie al quale poter generare l'oggetto-degenza sia su *Revit* che su *ArchiCAD*. In questo caso, però, non si tratta più di pura esportazione, ma è forse più corretto parlare di **trasferimento di dati**, in quanto l'algoritmo che origina le geometrie sulle piattaforme BIM continua a rimanere sul programma di modellazione parametrica, dove si può continuare a visualizzarlo ed editarlo. Inoltre un'esportazione prevede un tipo di staticità degli elementi che non può essere tipica di un processo che deve invece basarsi su **parametri dinamici** ed oggetti che devono poter essere costantemente modificabili ed aggiornabili. Malgrado questa sfumatura, che però è bene sottolineare sia per fare una distinzione con la metodologia precedente che per meglio comprendere i futuri sviluppi della ricerca, si intende comunque analizzare l'**interoperabilità** fra i programmi utilizzati. In particolare si tratta di un'interoperabilità nuovamente **orizzontale**, in quanto la comunicazione avviene sullo stesso piano di competenze, ma questa volta **indiretta** poiché le due piattaforme BIM si avvalgono di un intermediario per scambiare i dati, che in questo caso sono di definizione parametrica-geometrica.

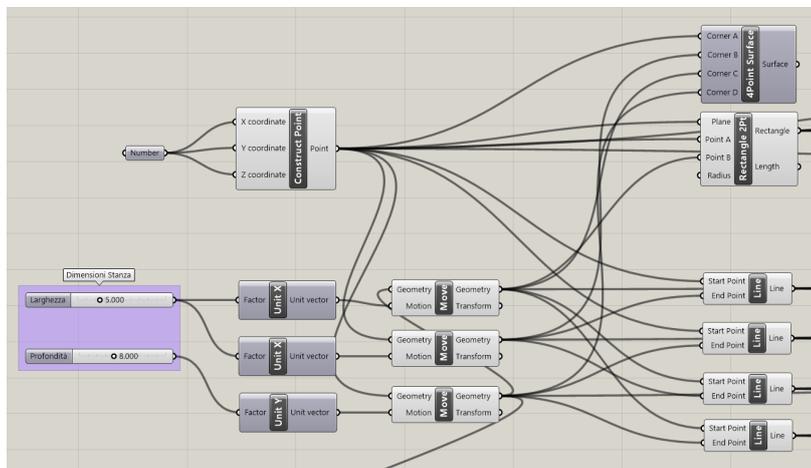
Essendo un metodo differente dal primo messo in atto, è diverso anche il modo con cui viene affrontato. Se prima infatti si poteva procedere "per *step*" decidendo come modellare ed in seguito testare i risultati ottenuti, ora è necessario invece avere un **riscontro immediato dell'algoritmo** che viene generato. Questo deriva ovviamente dalla differente natura del metodo, che si basa su una serie di componenti che costituiscono uno *script* e che, solo una volta concatenati fra loro, generano un *output* grafico ispezionabile dall'utente. È pertanto necessario che i test di interoperabilità avvengano in contemporanea

con la concezione degli *script* in modo da avere più controllo sui singoli risultati dei componenti.

Nonostante il cambio di metodologia, l'obiettivo di questa fase rimane invariato rispetto a quello affrontato con la precedente tecnica. Si vuole infatti arrivare ad ottenere un **prototipo di degenza semplificato**, costituito da pochi elementi di tipologia differente, al fine di verificare se l'interoperabilità con le piattaforme BIM ha un riscontro positivo o se sarà necessario cercare ancora una diversa strategia. Si procede quindi componente per componente confrontando parallelamente ed in modo diretto i tre *plug-in* selezionati per l'analisi.

- **Definizione del confine della degenza: muri**

Per prima cosa bisogna stabilire il perimetro dell'oggetto-degenza a cui si assoceranno i componenti del muro. Per fare ciò si suppone, come semplificazione, che la stanza non abbia il bagno e che presenti una forma regolare con un volume a parallelepipedo. Come ulteriore agevolazione si sceglie di posizionare uno dei vertici perimetrali sulle coordinate di origine, generando gli altri tre come sue copie traslate rispetto agli assi X e Y. Dopodiché è sufficiente aggiungere dei componenti "linea" per connettere i punti fra loro. Per definire le lunghezze delle linee si associano alle traslazioni dei punti dei cursori numerici (**slider**), che possono essere personalizzate in valori (minimo e massimo), arrotondamento e tipo di numero (intero, razionale ecc.).



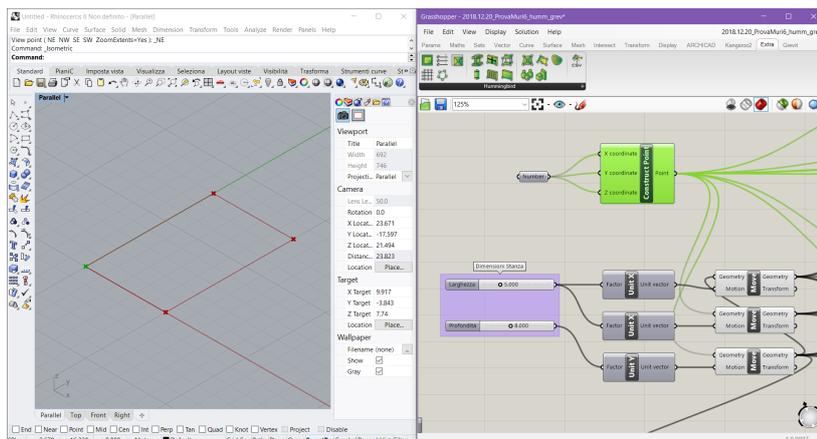
75 – Definizione del perimetro della degenza su Grasshopper

Questi *slider* sono i parametri su cui è possibile agire per modificare larghezza e profondità della stanza. Data la loro importanza si ritiene appropriato raggruppare e mettere in evidenza in modo chiaro su *Grasshopper* la loro funzione, nell'ottica di immaginare che questo *script* dovrà poi essere impiegato da un **utente esterno** che non lo ha mai visto e che sarà fornito al massimo di qualche linea guida per il suo utilizzo. Si adotterà quindi questo principio ogniqualvolta si presenti un'analogha situazione.

Come si può notare, i componenti usati finora non appartengono a nessuno dei *plug-in* citati. Le geometrie così generate non vengono pertanto visualizzate sulle piattaforme BIM in

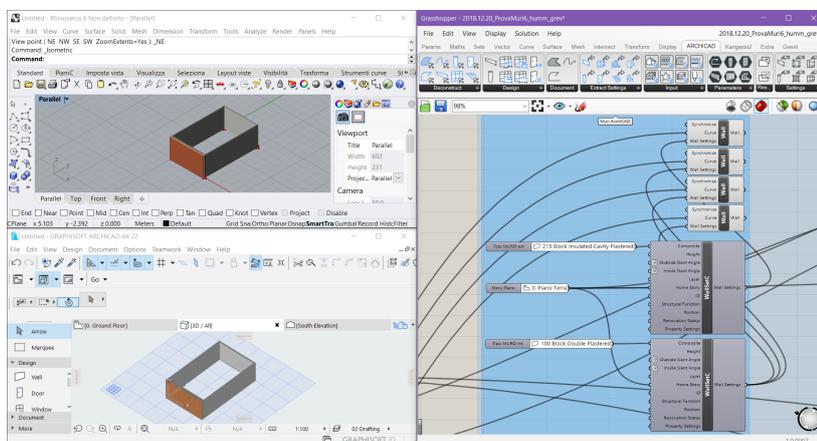
76 – Visualizzazione del perimetro della degenza, definito da punti e linee, su Rhinoceros

quanto identificano degli elementi bidimensionali di costruzione (punti e linee) visibili graficamente solo su *Rhinoceros*, considerata come software CAD a cui si appoggia *Grasshopper*.

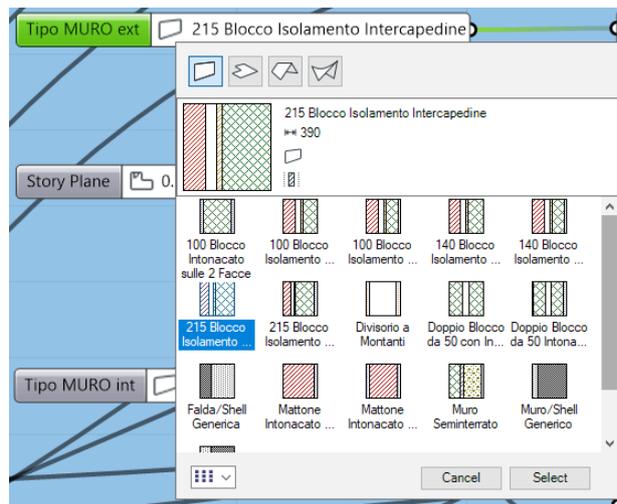


Per generare dei muri bisogna ora impiegare i componenti derivanti dai tre singoli *plug-in*, che necessitano tutti come *input* una **curva** per poter restituire l'*output* di un muro. Proprio per questo si ragiona con il perimetro bidimensionale della degenza. In particolare si sceglie di lavorare con delle linee, anziché con un unico rettangolo, in modo da poter più facilmente separare i muri, operazione che permette di associare impostazioni differenti ai singoli componenti. Per il prototipo semplificato si ipotizza che tre murature appartengano ad una partizione interna e che una facciata sia invece a contatto con l'esterno.

77 – Componenti che generano i muri mediante il *plug-in* “Grasshopper - ArchiCAD Live Connection”. Schermata con i tre programmi di lavoro: in alto a sinistra Rhinoceros, in basso a sinistra ArchiCAD, a destra Grasshopper.



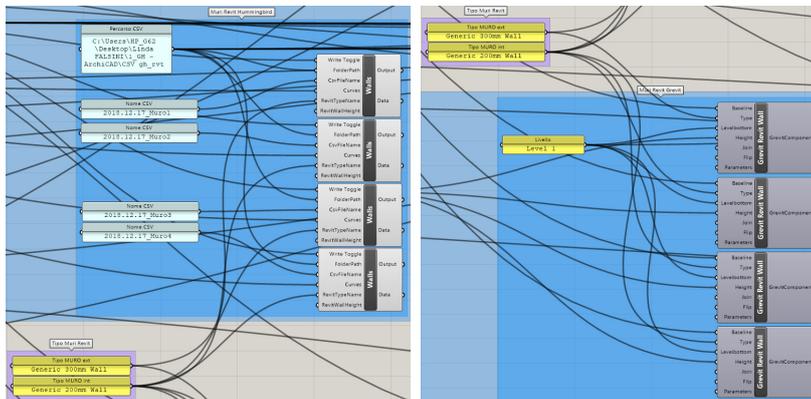
Il *plug-in* “Grasshopper - ArchiCAD Live Connection” consente di immettere numerose impostazioni aggiuntive, che permettono di personalizzare ogni singolo componente. Le principali, che sono state testate nel prototipo, sono la **stratigrafia** e i **livelli piano**. La peculiarità del *plug-in* è che, essendo una connessione diretta ed in tempo reale, non solo le modifiche sono visualizzabili immediatamente su *ArchiCAD*, ma si ha anche accesso ai pannelli, e quindi alle medesime opzioni, presenti sulla piattaforma BIM. Si evince quindi che per aggiungere e modificare stratigrafie o livelli è sufficiente inserirli da *ArchiCAD* in modo che tali impostazioni compaiano anche su *Grasshopper*.



78 – Pannelli di ArchiCAD visualizzabili da Grasshopper mediante il plug-in “Grasshopper - ArchiCAD Live Connection”

Hummingbird e *Grevit* al contrario non possiedono specifici componenti per la modifica delle impostazioni ed è pertanto necessario aggiungere una **nota di testo (panel)** per scrivere manualmente i dati che si vogliono immettere. Nel caso della stratigrafia bisogna quindi prendere visione, da *Revit*, del nome del tipo di muro che si vuole impiegare e riportarlo nel *panel* da collegare come *input* al relativo componente muro. Come elemento aggiuntivo *Hummingbird* necessita di ulteriori note di testo per specificare i nomi dei file CSV da esportare ed il percorso della cartella in cui salvarli.

Con nessuno dei tre *plug-in* testati è possibile determinare il filo di inserimento del muro, che, per impostazione standard viene definito con linea di ubicazione centrale.

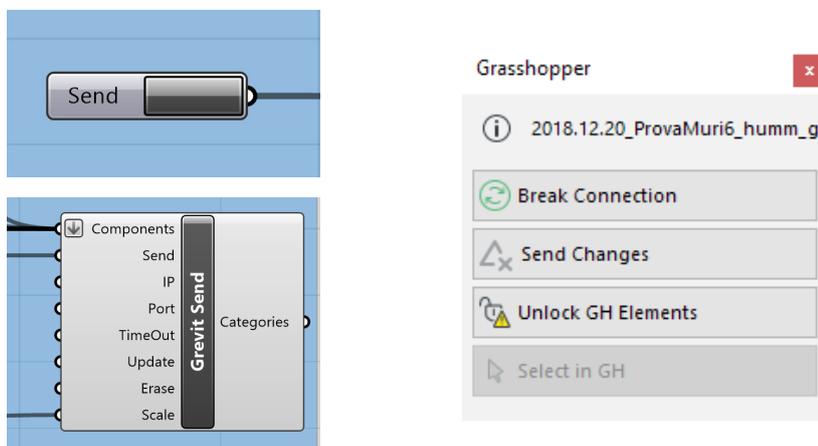


79 – Canvas Grasshopper: a sinistra i componenti per la generazione dei muri su Revit mediante il plug-in *Hummingbird*, a destra quelli di *Grevit*

I *plug-in* per il trasferimento dati in *Revit*, contrariamente a quello per *ArchiCAD*, non sono diretti e richiedono quindi degli elementi che diano l'**input per l'esportazione**. Per *Hummingbird* questo si ottiene collegando un pulsante (*button*) a tutti i componenti di cui si vuole estrarre il CSV. Per *Grevit* un componente apposito si occupa dell'invio delle informazioni, quindi ogni altro componente deve avere questo come *output*, che a sua volta deve essere collegato ad un *button*, come per *Hummingbird*. Con *ArchiCAD Live Connection* questa operazione non è necessaria, ma è tuttavia presente la possibilità di inserire un pulsante booleano per disattivare la sincronizzazio-

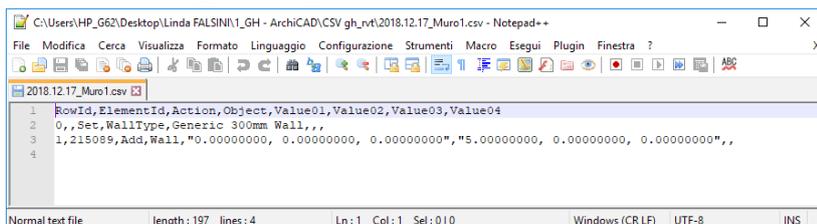
ne automatica dei dati che, di *default* è invece attiva. Affinché essa funzioni correttamente è sufficiente assicurarsi che sia su *Grasshopper* che su *ArchiCAD* sia attiva la **connessione**, oltre ad essere collegati ad una **rete internet stabile**. Se la prima condizione non si verificasse, su *Grasshopper* il componente notifica l'errore, mentre se è assente la seconda *Rhinoceros*, e quindi anche *Grasshopper*, si arresta improvvisamente.

80 – Elementi per il trasferimento dei dati da *Grasshopper* alle piattaforme BIM. Da sinistra a destra: button, componente base di *Grasshopper*, utile per il trasferimento su *Revit* sia mediante *Hummingbird* che *Grevit*; componente di invio dati di *Grevit*; pannello di connessione in *ArchiCAD*



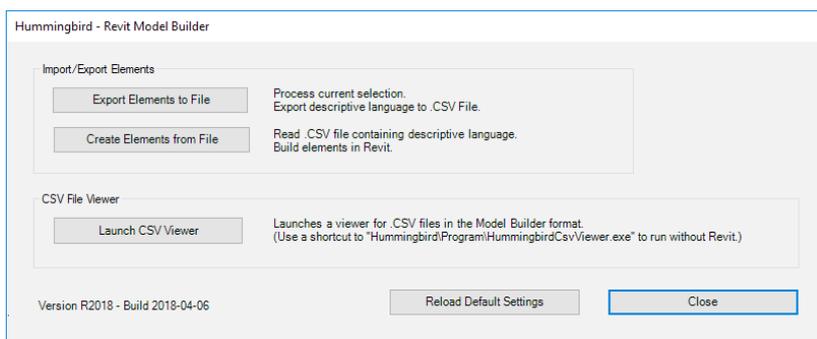
Come già precisato, su *ArchiCAD* si può visualizzare in parallelo ciò che l'algoritmo genera senza nessun passaggio di esportazione intermedio. Per avere invece un riscontro grafico su *Revit* bisogna prima di tutto cliccare il componente *button* inserito su *Grasshopper*, e poi operare dalla piattaforma BIM. Nel caso di *Hummingbird* si può verificare che si siano generati i file CSV con il nome specificato e nella cartella di destinazione scelta. È inoltre possibile visualizzare – e modificare – il file **.csv** aprendolo con un programma come *Microsoft Excel*, blocco note, incluso nel sistema di Windows, o *Notepad++*, che è un editor di testo gratuito per Windows.

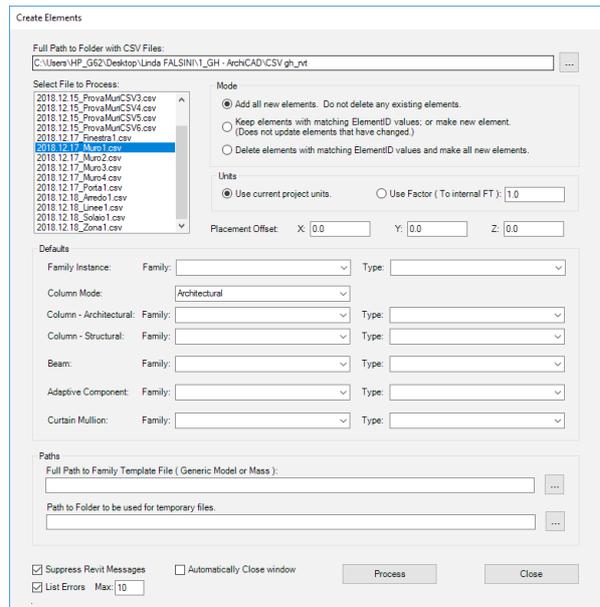
81 – Struttura di un file .csv di un componente muro, estratto da *Grasshopper* tramite il plug-in *Hummingbird* e visualizzato con *Notepad++*



Dopodiché si selezionano i CSV che si vuole trasferire su *Revit*, caricandoli **uno alla volta**, e si procede con l'esportazione appurando che il *plug-in* non segnali nessun errore al termine di ogni operazione.

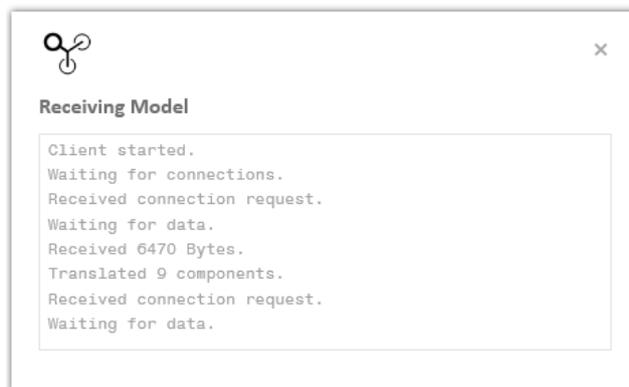
82a – Prima finestra di importazione dei CSV, esportati da *Grasshopper* grazie al plug-in *Hummingbird*, su *Revit*





82b – Seconda finestra di importazione dei CSV, esportati da Grasshopper grazie al plug-in Hummingbird, su Revit

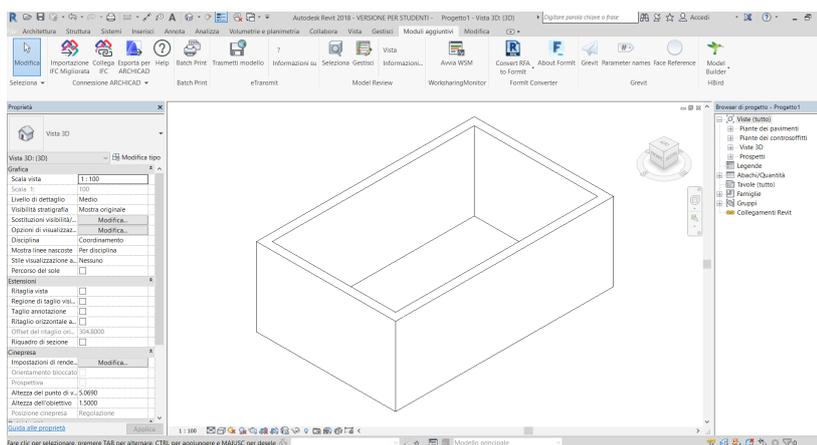
Per trasferire i medesimi dati, ma usando *Grevit*, bisogna prima predisporre *Revit* alla ricezione dei dati mediante l'apposito pannello installato fra i moduli aggiuntivi assieme al *plug-in*, e poi premere il componente di invio da *Grasshopper*.



83 – Finestra di ricezione dei dati su Revit, mediante il plug-in Grevit

In entrambi i casi ottengo lo stesso risultato su *Revit*, ovvero la **traduzione dell'algoritmo** che costituisce la rappresentazione dei quattro muri perimetrali della degenza. A differenza di come accade su *ArchiCAD* però, su *Revit* il modello non si aggiorna automaticamente al variare dei parametri su *Grasshopper*, ed è quindi necessario ripetere manualmente le operazioni di esportazione e trasferimento dei dati ad ogni modifica effettuata.

84 – Trasposizione grafica su Revit dell'algoritmo di generazione dei muri realizzato su Grasshopper



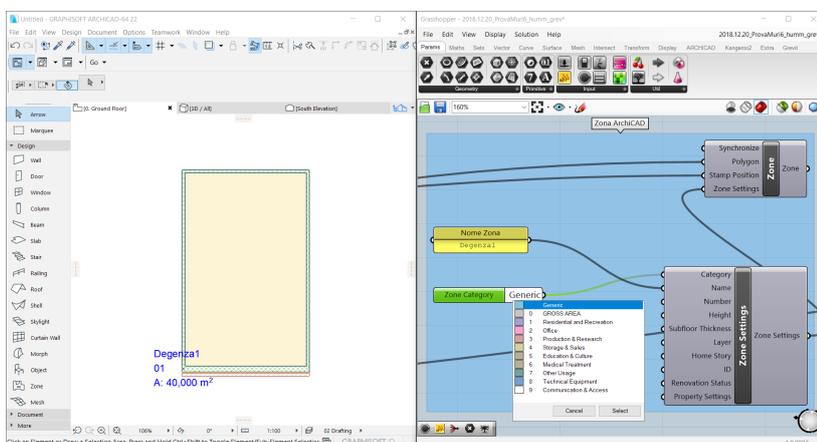
- **Definizione del pavimento: solaio**

Per lo *script* di generazione del solaio si seguono gli stessi procedimenti sopra descritti e non si riscontrano particolari differenze rispetto al processo seguito per i muri. L'unica accortezza riguarda l'*input* richiesto dal componente solaio, che deve essere costituito da una superficie chiusa ed è pertanto opportuno impiegare il **componente rettangolo** anziché la sommatoria delle linee sopra utilizzate.

- **Definizione della destinazione d'uso: zona**

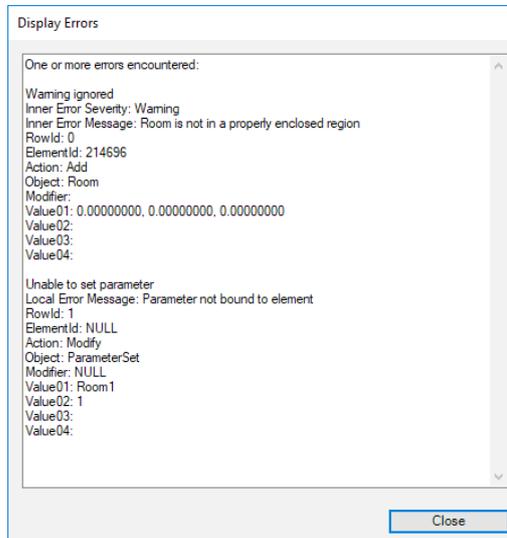
Con *Grasshopper - ArchiCAD Live Connection*, seguendo lo stesso processo valido per gli altri due componenti, si riesce a produrre, su *ArchiCAD*, una zona che identifica la funzione d'uso che si desidera individuare.

85 – Generazione della zona: a sinistra ArchiCAD, a destra porzione del canvas di Grasshopper

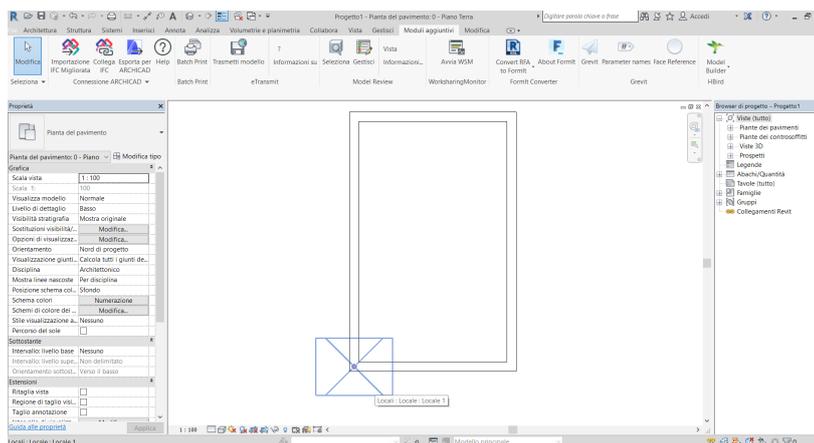


La questione è differente per quanto riguarda *Revit*, i cui *plug-in* possiedono *input* diversi per la generazione del locale. Per *ArchiCAD* infatti esso è generato grazie al componente poligonale che identifica il perimetro della degenza, mentre quello richiesto da *Hummingbird* è invece un **input di punti**. Tuttavia, collegando i vertici che delimitano l'area della degenza al componente di generazione del locale, l'importazione del CSV notifica l'impossibilità di assegnare la zona per assenza di una regione chiusa. Analizzando le informazioni presenti nel file di testo è possibile dedurre che in realtà è stato correttamente associato al componente solamente uno dei quattro punti desi-

gnati come *input*. Nell'abaco dei locali su *Revit* è infatti possibile vedere che il locale è presente ma "non racchiuso", e puntando con il mouse sulle coordinate indicate sul CSV esso compare puntualmente, ma con un'area di dimensione standard non definita.

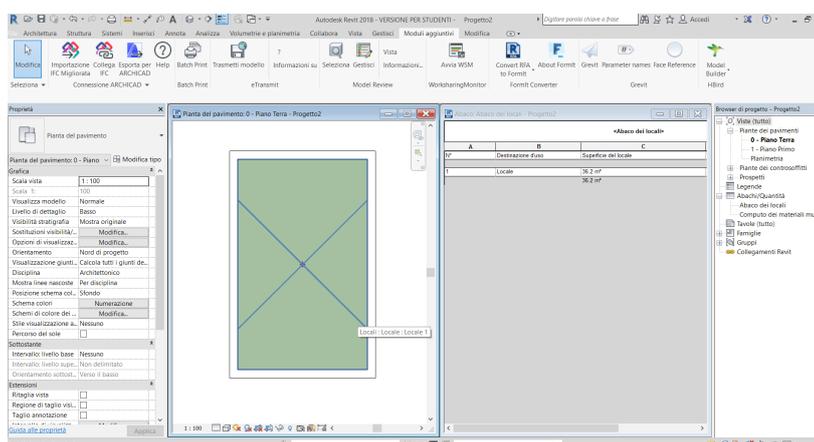


86a – Finestra di errore nel trasferimento del dato del locale, da *Grasshopper* a *Revit*, mediante il plug-in *Hummingbird*, usando come *input* i quattro vertici che delimitano la degenza



86b – Rappresentazione dell'errore su *Revit* del trasferimento del dato del locale, da *Grasshopper* a *Revit*, mediante il plug-in *Hummingbird*

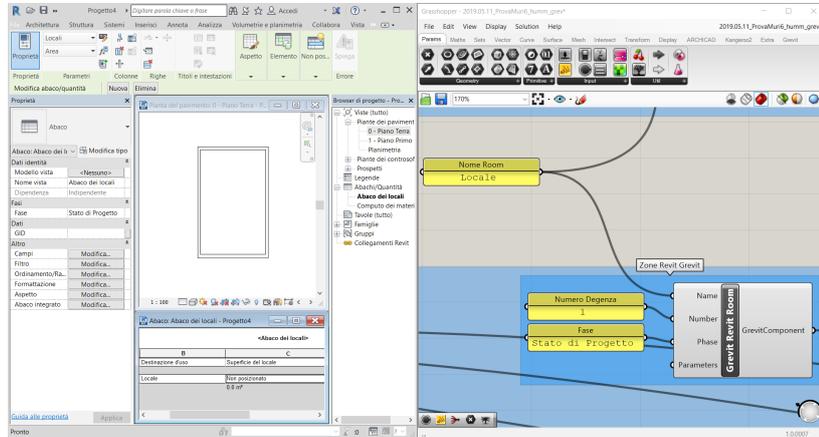
Per correggere questo errore è sufficiente pensare in analogia a come su *Revit* si assegnino le destinazioni d'uso ad una stanza delimitata da quattro mura. Tale operazione preclude che l'utente clicchi al centro della stanza, e che quindi l'*input* da impiegare su *Grasshopper* debba anch'esso essere un **unico punto**, purché interno all'area del locale che si desidera creare.



87 – Corretta generazione del locale su *Revit*, mediante l'*input* al componente di *Hummingbird* di un singolo punto centrale alla degenza

Con *Grevit*, invece, il componente per la creazione dei locali non possiede uno specifico *input* per la sua definizione ed è possibile solamente definirne i parametri di testo. Per questo motivo su *Revit*, benché il dato venga correttamente trasferito, esso non è visualizzabile se non nell'abaco dei locali, dove compare come “non posizionato”.

88 – Generazione del locale fallita tramite il plug-in *Grevit*: trasferimento del dato avvenuto, ma senza il suo corretto posizionamento e delimitazione

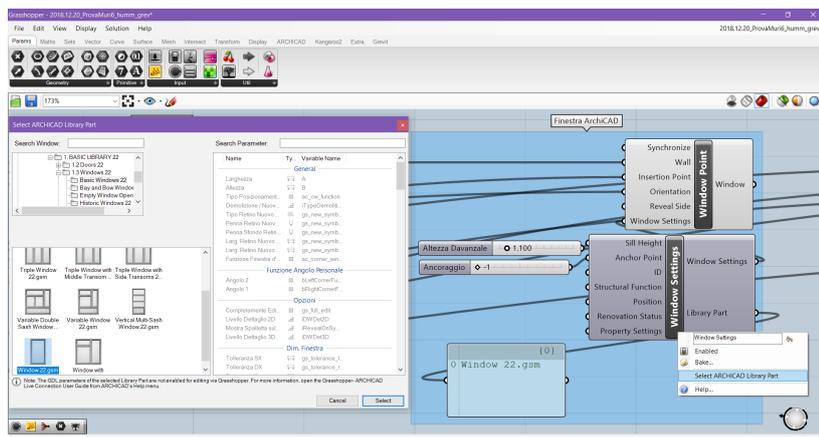


- **Inserimento di serramenti: porta e finestra**

Per inserire degli infissi su *ArchiCAD* si usano i componenti di porte e finestre come **output dei muri**, con il vincolo di poter essere posizionati solo in corrispondenza degli stessi, in quanto essi non possono “fluttuare” nel modello senza essere associati ad un componente che li ospiti. Anche in questo caso risulta vantaggioso aver generato i muri usando come *input* delle linee singole anziché un rettangolo. Con la seconda soluzione si otterrebbe infatti un serramento su ognuno dei quattro muri, senza poter decidere di inserirne uno solamente su quello desiderato.

Per selezionare il tipo di serramento, cliccando con il tasto destro direttamente sul componente, si può accedere all'intera **libreria** presente sul file *ArchiCAD* connesso. In questo caso infatti non c'è un apposito *input* che apra un pannello che elenchi le opzioni, come per le stratigrafie. Pertanto, per avere un riscontro visivo immediato su *Grasshopper* per capire quale oggetto di libreria è selezionato, bisogna aggiungere una nota di testo come *output* che ne restituisca il nome.

89 – Accesso alla libreria di *ArchiCAD* dal canvas di *Grasshopper*



Revit invece considera porte e finestre come famiglie caricabili, quindi, operando con i *plug-in Hummingbird* e *Grevit*, si ritrova un unico componente da impiegare per ogni oggetto di questo tipo. L'*input*, come in precedenza, è costituito da un *panel* con il nome specifico della famiglia, precedentemente caricata nel file *.rvt*, unitamente anche al **punto di inserimento** in cui si vuole localizzare l'elemento. Per quanto riguarda il trasferimento di questi dati mediante *Grevit* non si constatano problemi, mostrando su *Revit* la corretta localizzazione dei serramenti.

Con l'importazione del CSV esportato tramite *Hummingbird* si riscontra invece una grave mancanza del *plug-in* che non consente di importare nessun tipo di famiglia con *host*, malgrado il file *.csv* sia correttamente generato e contenga le relative informazioni. Dopo una ricerca attinente al problema rilevato si può intendere da una risposta ad un *post* di un utente sul sito ufficiale di *Grasshopper* [78] che sia un difetto proprio del *plug-in* al quale gli sviluppatori promettono di lavorare per le successive versioni. Per il momento consigliano pertanto di immettere manualmente direttamente su *Revit* porte e finestre, anche se non si esclude l'idea di impiegare un secondo *plug-in* per continuare ad automatizzare il processo.

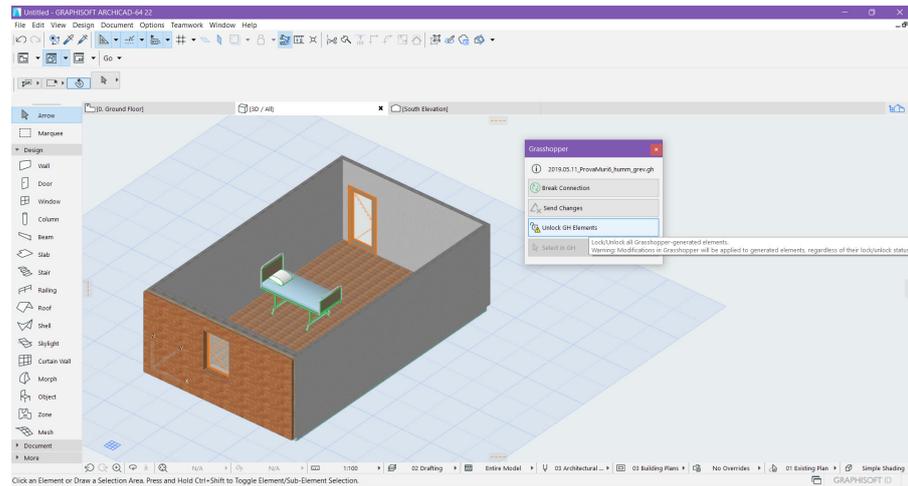
- **Inserimento di un oggetto da libreria/famiglia: arredo**

L'ultimo elemento da testare per il prototipo è un arredo generico a rappresentazione dell'intera gamma di oggetti disponibili. Per *ArchiCAD*, come per i casi precedenti, è presente un apposito componente che propone comunque le stesse modalità di funzionamento di quelli per la generazione dei serramenti. In aggiunta alle impostazioni precedenti vi è inoltre la possibilità di scegliere se impiegare un oggetto di libreria esistente o se crearne uno nuovo.

Con *Hummingbird* e *Grevit* il procedimento da seguire è lo stesso applicato per i serramenti, con la differenza che questo tipo di famiglie viene ora correttamente trasferito da entrambi i *plug-in*.

Il prototipo finito su *ArchiCAD* presenta tutti gli oggetti che sono stati generati su *Grasshopper*, con anche il requisito di rimanere aggiornato rispetto alle modifiche effettuate sul programma di *visual editing*. Tutti gli elementi trasferiti risultano di *default* bloccati su *ArchiCAD*, quindi per poterli modificare è necessario prima renderli editabili.

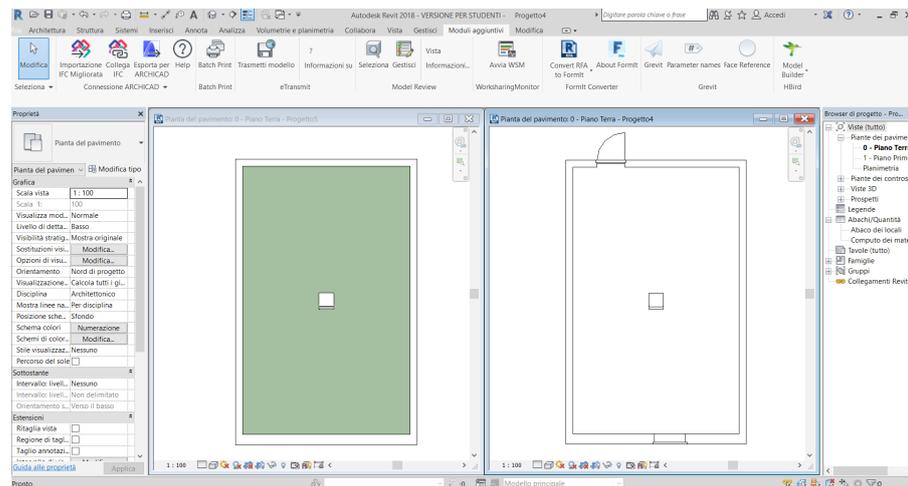
90 – Prototipo ottenuto su ArchiCAD, in cui ogni elemento generato mediante il plug-in Grasshopper - ArchiCAD Live Connection risulta bloccato



Su *Revit* si ottengono risultati diversi del prototipo in base al *plug-in* impiegato per il trasferimento dei dati. Questo non può portare all'esclusione di uno dei due, come si poteva ipotizzare prima di verificarne il funzionamento, bensì all'utilizzo di entrambi in modo **complementare**. Infatti i difetti di uno possono essere colmati dall'altro, in quanto *Hummingbird* non consente di generare famiglie con *host* come i serramenti, mentre *Grevit* non trasferisce completamente le informazioni dei locali.

A differenza dell'altra piattaforma BIM, con *Revit* non è possibile avere l'aggiornamento automatico delle modifiche ed è quindi l'utente a doversi occupare di mantenere le informazioni sincronizzate e di capire quando non lo sono. Questo sminuisce parzialmente i risultati ottenuti, per quanto siano comunque di livello superiore a quelli ottenuti mediante la precedente metodologia di trasferimento di dati parametrici.

91 – Confronto dei due *plug-in* utilizzati per il trasferimento dei dati su *Revit*. A sinistra il prototipo ottenuto mediante *Hummingbird*, carente dei serramenti, a destra quello da *Grevit*, a cui manca il locale



In entrambi i casi bisogna tener presente che è necessario configurare le corrette **unità di misura** non solo sulla piattaforma BIM, ma anche su *Rhinceros*. Questo perché, pur non lavorandoci, le informazioni derivanti da *Grasshopper* attuano comunque un passaggio intermedio di traduzione passante per il software di modellazione, che lo ospita come *plug-in*, a quello a cui vengono trasferiti i dati finali.

È inoltre da considerare che nella realizzazione di questo prototipo sono state impiegate soluzioni che potrebbero essere applicate di-

versamente. Con programmi di *visual scripting*, basati su operazioni logico-matematiche, è infatti facile che esistano numerose **strade differenti** per creare algoritmi che generino, alla fine, il medesimo risultato.

Per quanto riguarda la **bidirezionalità** dei tre *plug-in* si può affermare che il flusso preferenziale delle informazioni sia quello testato finora, ovvero dallo *script* verso le piattaforme BIM. Ciononostante la palette di *ArchiCAD* possiede uno strumento per l'invio delle modifiche a *Grasshopper*, ma modificando gli elementi essi tornano ad essere come nell'algoritmo invece che modificare lo stesso. *Hummingbird* offre la possibilità di creare dei CSV direttamente da *Revit* di elementi selezionati o di quelli relativi all'intero modello, che possono a loro volta essere impiegati per l'importazione. *Grevit* invece permette solo di ricevere da *Revit* e non di estrapolare nessun dato. Si evince quindi che, per mantenere l'oggetto-stanza aggiornato su entrambi i programmi testati, bisogna obbligatoriamente agire direttamente sul software di *visual scripting*, che funge perciò da fonte di generazione dei dati.

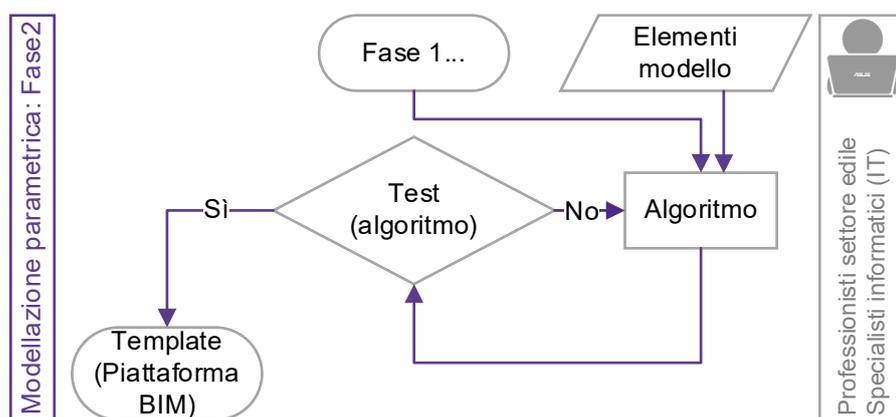
Nel capitolo “5 Allegati digitali” è possibile scansionare un QR code (n. 1) per prendere visione dell'intero algoritmo appena analizzato.

2.2 Fase 2: Generazione del modello parametrico di una degenza

La seconda fase della ricerca si basa sullo studio metodologico ed i risultati dei test attuati durante l'analisi precedente. Grazie alla prima fase si può infatti ora procedere con lo **sviluppo dell'algoritmo** per la generazione del modello parametrico della degenza. Come anticipato nel capitolo precedente, la validità dello *script* visuale deve essere costantemente testata tramite la piattaforma BIM per verificarne l'effettiva efficienza. Pertanto, nonostante il collaudo sia nuovamente il perno attorno a cui ruota l'intera fase, esso non ha la stessa valenza dei test d'interoperabilità della precedente, bensì prevede una **ciclicità del processo** che è parallelamente in continua evoluzione e raffinamento.

L'algoritmo così elaborato viene quindi trasferito all'interno della piattaforma BIM, la quale ne eredita i parametri e si prende carico di garantirne la corretta comunicazione. Il modello BIM permette infatti di gestire da un lato la resa grafica degli elementi parametrizzati, dall'altro la restituzione informativa di dati che vanno al di là del semplice aspetto geometrico. Questo è possibile grazie all'utilizzo di un **template**, ovvero di un file modello predefinito, che contenga, come un archivio di settaggi, tutte le impostazioni standard volte a facilitare ed ottimizzare tale flusso di lavoro. La realizzazione di un *template* è uno *step* fondamentale di questa fase, in quanto rende possibile un'efficiente gestione del progetto che veda alla sua base la **generazione automatica di informazioni** quantitative e qualitative. Il livello di dettaglio di tali informazioni, in questa fase ed in quella successiva, può essere superiore rispetto alla fase di ricerca precedente, oscillando tra il **LOD C e D**, in riferimento alla normativa UNI 11337-4 [20]. Alcuni elementi, come si vedrà più nello specifico in seguito, potranno arrivare ad essere oggetti dettagliati, altri invece solamente definiti, in base alle scelte progettuali attuate.

92 - Schema metodologico della fase 2. Per una visione d'insieme vd. diagramma di flusso dell'intera metodologia (fig. 24)



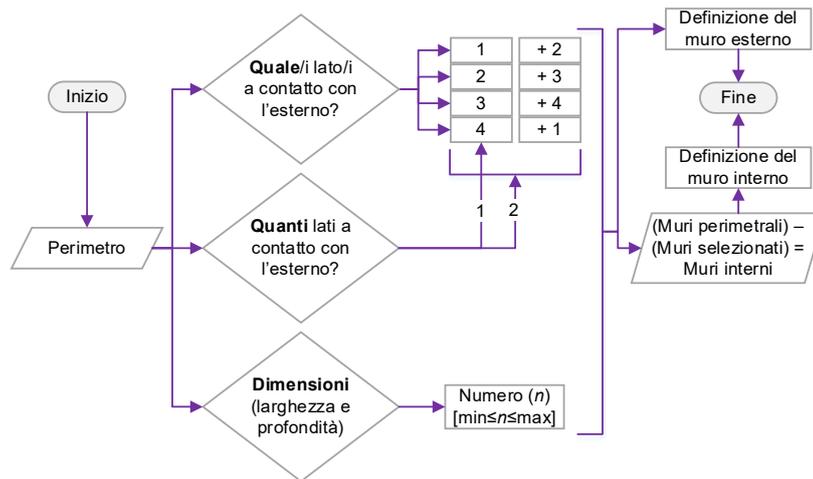
2.2.1 Definizione dell'algoritmo e dei vincoli progettuali

La precedente fase di ricerca ha dimostrato come la metodologia presa in esame possa mettere in comunicazione le due piattaforme BIM più diffuse per utilizzo. Ciononostante, la scelta del software parametrico impiegato ha fatto sì che nascessero delle criticità tali da rendere uno dei due flussi di lavoro più all'avanguardia rispetto all'altro. È di fatto evidente che il processo che vede come fulcro la piattaforma BIM di *ArchiCAD* trovi meno ostacoli sul suo cammino rispetto a quelli riscontrati nell'utilizzo di *Revit*. Per questo motivo, unitamente alla già citata poca affidabilità ed alla difficoltà nel reperimento degli specifici *plug-in* che connettano la piattaforma *Autodesk* al software per l'editor algoritmico visuale di *McNeel & Associates*, si decide di portare avanti la ricerca focalizzandosi sulle potenzialità legate all'**affinità tra Grasshopper ed ArchiCAD**. Questo non intende vanificare i risultati ottenuti né presupporre che la strada scelta sia l'unica percorribile, anzi, l'obiettivo è ora quello di approfondire il più possibile una sola delle soluzioni precedentemente analizzate in modo da far convergere l'attenzione sugli specifici problemi anziché sulle generiche criticità legate ai software.

Si illustra qui di seguito lo *script* visuale, porzione per porzione, suddividendolo in base alle funzioni svolte all'interno dell'algoritmo. Per ognuna di esse un diagramma di flusso illustra nello specifico da quali **input** lo *script* in analisi dipenda direttamente (parallelepipedo), quali **condizioni** sono poste (rombo) e fra quali **opzioni** può muoversi l'utente utilizzatore (rettangolo). Tutto ciò che non è specificato è da ritenersi metodologicamente identico al processo di generazione impiegato per il prototipo del capitolo precedente.

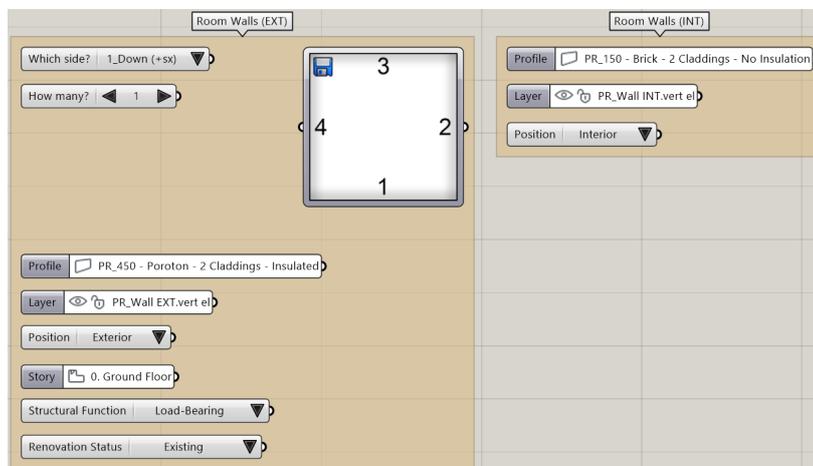
Per questioni pratiche, non è qui riportata la totalità delle schermate che costituiscono lo *script*, ma piccole porzioni a scopo illustrativo. Per prendere visione delle parti mancanti dell'algoritmo e di un video dimostrativo consultare il capitolo "5 Allegati digitali" e scansionare il relativo *QR code* (n. 2 e 3).

- **Muri esterni ed interni**



93 – Diagramma di flusso: elaborazione del componente “muro”

La generazione dei muri perimetrali si basa sulle quattro linee di confine che racchiudono l’area della degenza e le cui lunghezze vengono definite tramite due *slider*. Dovendo dare la possibilità di definire le stratigrafie di tali murature, è fondamentale porre due tipologie di condizioni. Una riguarda la **quantità di muri rivolti verso l’esterno**, immaginando lo scenario più comune in cui solo uno dei quattro appartiene alla facciata dell’edificio, ma anche quello in cui la degenza si trovi all’estremità della struttura sanitaria che la ospita, esponendo così due muri verso l’esterno. La seconda condizione serve invece a selezionare più nel dettaglio la **posizione del muro**, o dei muri, che sono a contatto con l’esterno. Dopo aver specificato queste due opzioni, l’algoritmo calcola in automatico quali murature non sono state prese in considerazione inputandole come pareti interne. Entrambe le tipologie possono ora essere definite non solo nelle stratigrafie, ma in tutti gli ambiti informativi che il *plug-in Grasshopper - ArchiCAD Live Connection* consente di specificare.

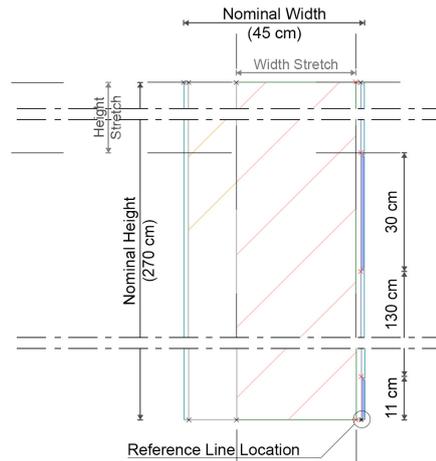


94 – Opzioni rese effettuabili su Grasshopper: muri esterni e muri interni

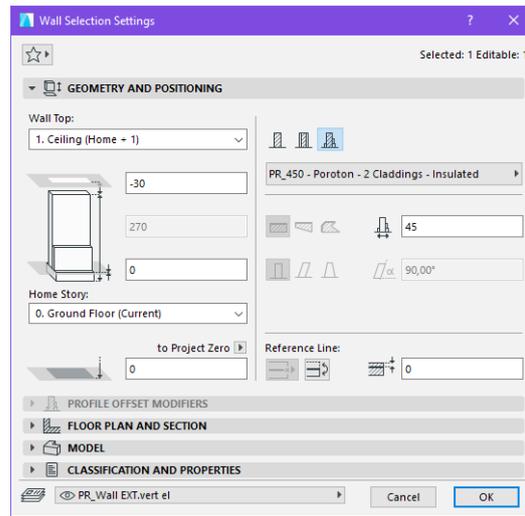
Il Manuale dell’architetto [40] precisa che i muri debbano essere provvisti di zoccolatura e fasce protettive, il che richiede la modellazione di un apposito **profilo complesso** su *ArchiCAD*. Tale accorgimento permette inoltre di aggirare l’ostacolo

del filo di inserimento del muro, che, come anticipato per il prototipo, viene definito di *default* con linea di ubicazione centrale, sia per le strutture semplici (monostrato) sia per quelle composte (multistrato). Editando il profilo si può infatti definire puntualmente dove la linea di riferimento attraversi il muro. In questo caso tutti i profili perimetrali della degenza sono modellati in modo da mantenere il **filo interno**. Questa operazione, come si vedrà successivamente, faciliterà il calcolo sia dell'area netta, soggetta a restrittivi limiti normativi, sia dei vincoli dimensionali.

95 – Esempio di profilo complesso modellato su ArchiCAD

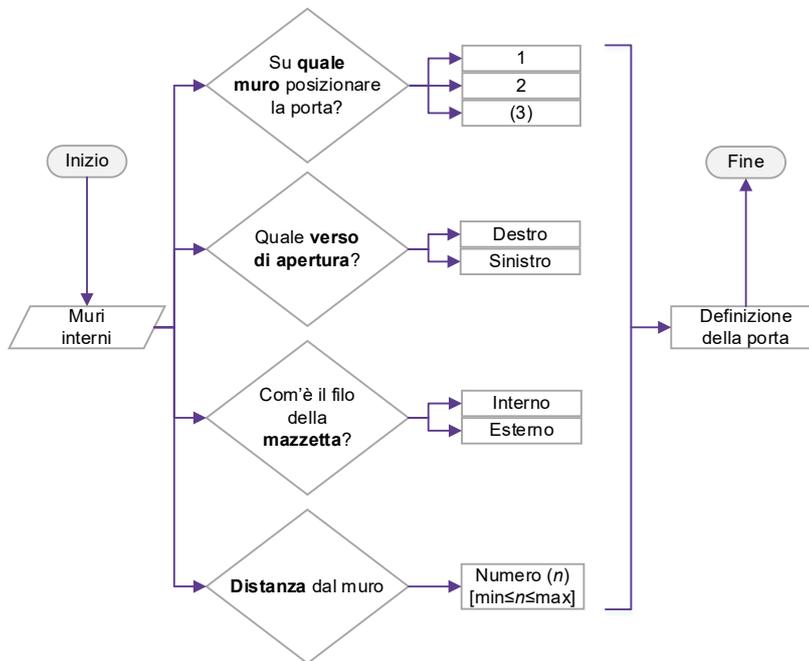


Di contro, la gestione di un profilo prevede una rigorosa definizione geometrica che ne determina un'**altezza**, la quale all'esterno dell'editor può solo essere incrementata. È di fatto consigliabile modellare il muro con l'altezza minore consentita dalla normativa, che in questo caso risulta di 2,70 metri. Utilizzando solamente *ArchiCAD* per la modellazione basterebbe collegare il muro, con il profilo selezionato, ai due **livelli** tra cui è compreso, impostando un *offset* pari a 0 da entrambi. Così facendo, dovendo modificare una quota, è sufficiente editare il valore del livello corrispondente per correggere allo stesso tempo tutti i muri che vi sono collegati, nonostante i profili possiedano intrinsecamente una misura differente. Con l'approccio in analisi questo passaggio non risulta però efficace, in quanto il valore dell'altezza con cui viene modellato il profilo prevale su quello assegnato ai livelli. Ciò significa che, pur collegando il muro ai due livelli tramite le impostazioni dello specifico componente su *Grasshopper*, esso mantiene la sua altezza originaria creando automaticamente un *offset* dal livello superiore, pari alla differenza tra la sua quota e quella del livello. Per ovviare a questa incongruenza è quindi necessario **regolare l'altezza dei muri manualmente** tramite uno *slider* che è da aggiornare manualmente in caso di modifiche alle quote.



96 – Muro generato da Grasshopper su ArchiCAD, senza la correzione della quota: pur essendo collegato al livello superiore (di quota 3,00 m) mantiene la sua altezza originaria (2,70 m), presentando un offset di 30 cm.

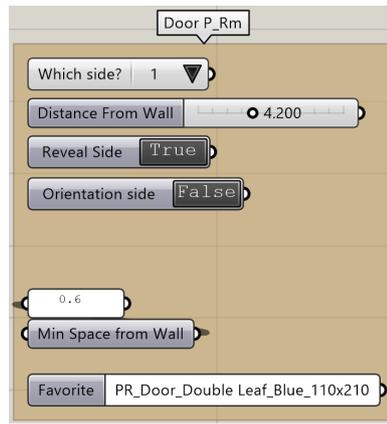
- **Porta d'ingresso della degenza**



97 – Diagramma di flusso: elaborazione del componente "porta"

Il posizionamento della porta d'ingresso della degenza dipende da quali **muri interni** sono risultati selezionabili dalla precedente definizione delle pareti. La prima scelta effettuabile, infatti, riguarda proprio quale muro ospiti l'accesso. In secondo luogo possono essere definiti altri aspetti pratici, quali la direzione di apertura del serramento, ovvero il lato di inserimento dei cardini, ed il filo della mazzetta.

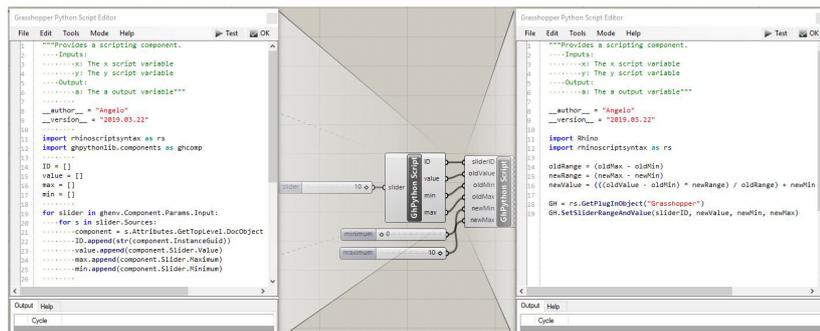
98 - Opzioni rese effettuali su Grasshopper: porta



Il primo vincolo posto all'interno dell'algoritmo, invisibile agli occhi di un utente esterno, riguarda il **senso di apertura** della porta. Esso è infatti inevitabilmente rivolto verso l'interno della degenza, come specificato dalla normativa di riferimento, e non rappresenta pertanto un'opzione modificabile. Allo stesso modo, anche la scelta della quota, che distanzia la porta dal muro ad essa perpendicolare, è vincolata. In tal caso l'utente può però modificarne i valori, ma inconsapevole del fatto che l'intervallo fra cui può muoversi è strettamente vincolato in base alla dimensione del muro su cui il serramento giace.

Quest'ultima condizione è inserita per evitare che il punto di ancoraggio dell'elemento superi una prefissata **"distanza di sicurezza"** che farebbe altrimenti fuoriuscire la porta dai muri perimetrali. Per porre questo vincolo, adottato anche in altre successive porzioni di *script*, è impiegato il componente di *scripting editor*, messo a punto, in linguaggio *Python*, dal collega Angelo Lombardo. Questo *script* permette di associare ad uno *slider* un valore minimo ed uno massimo oltre i quali non può sconfinare. Nell'algoritmo in analisi a tali *input* sono state connesse le condizioni da cui dipendono le variabili dimensionali a cui è legato l'oggetto. Questo significa che esse non corrispondono a numeri fissi, ma si modificano in modo **parametrico** all'unisono con le modifiche apportate alle porzioni di *script* a cui il componente è vincolato. Nel caso specifico della definizione della porta le condizioni da cui dipendono minimo e massimo sono rappresentate dal muro su cui giace la porta e dalla sua lunghezza.

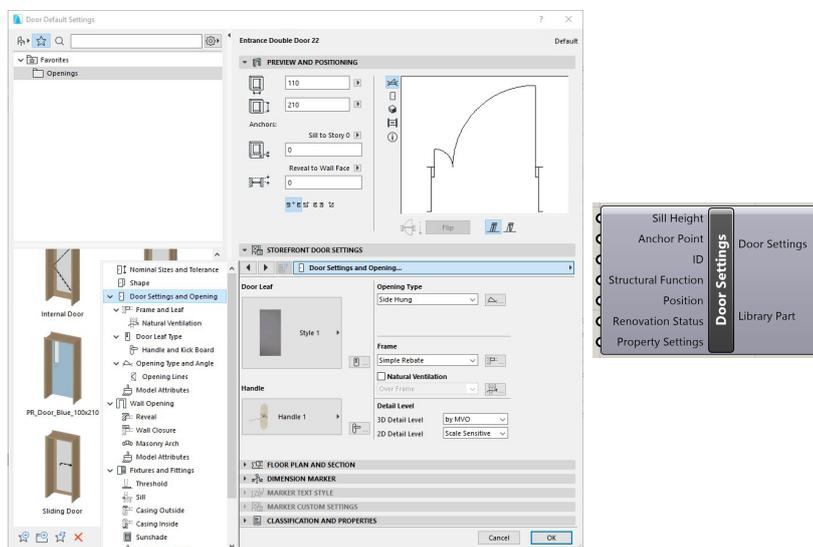
99 – Componente "GH Python Script" da cui dipendono i valori dello slider a cui è collegato



Una valida alternativa, in assenza di un apposito elemento editato, sarebbe l'impiego di componenti per la creazione del

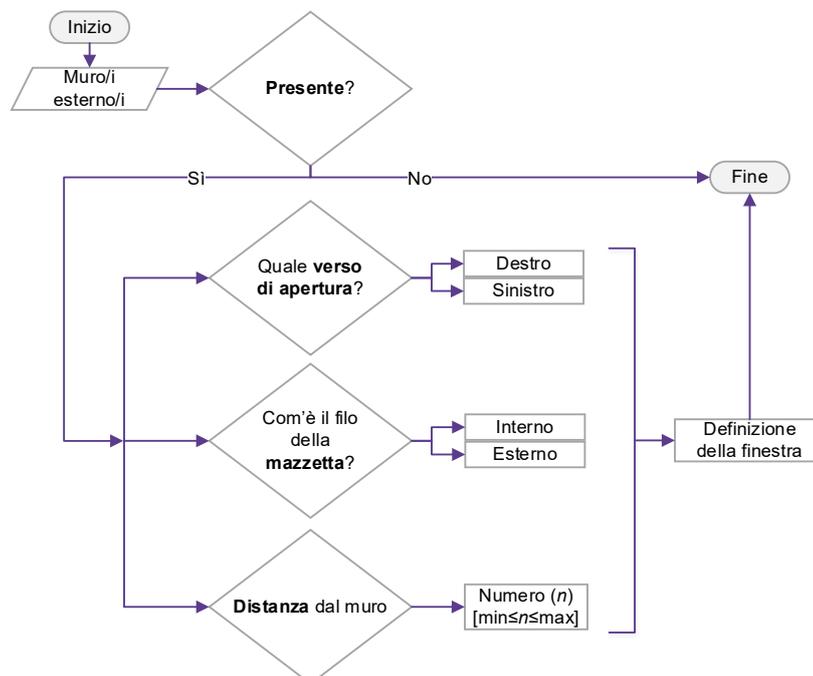
dominio. Questo consentirebbe ugualmente di muovere gli oggetti ancorati solo all'interno di un dato intervallo parametrico, ma lo *slider* continuerebbe a mostrare anche i valori numerici in cui il suo dominio non viene verificato. In tal modo, pur essendo concettualmente corretto da un punto di vista geometrico, l'utente avrebbe meno consapevolezza delle dimensioni con cui ha a che fare.

Nonostante la flessibilità intrinseca del *visual scripting*, la modellazione del serramento permessa da *Grasshopper* prevede la scelta di poche impostazioni per quanto riguarda la geometria dell'elemento. Per tal motivo, invece di impiegare il componente specifico delle proprietà della porta, si predilige in questo caso impostare a monte un **preferito** su *ArchiCAD*, che racchiuda al suo interno sia informazioni di carattere generale che una predeterminata geometria.



100 – Disponibilità delle proprietà di una porta: impostazioni di ArchiCAD (a sinistra) e di Grasshopper (a destra)

● Finestre

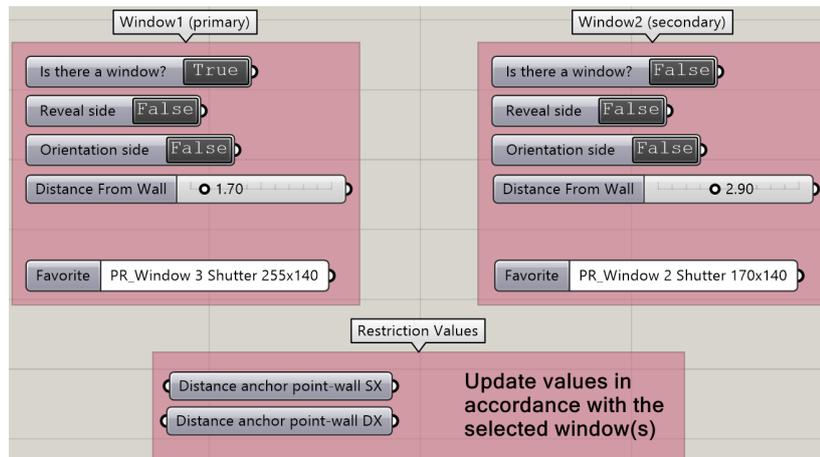


101 – Diagramma di flusso: elaborazione del componente “finestra”

La generazione di una finestra può avvenire solo in corrispondenza delle strutture verticali rivolte verso l'**esterno**, che, come anticipato, possono essere uniche o risvoltate ad angolo. Per questo motivo è necessario prevedere, all'interno dello *script*, la presenza di almeno due finestre, una per ogni possibile muro esterno, dando loro, come primo nodo, una **condizione di esistenza**. Pertanto, se le pareti a contatto con l'esterno sono due, si ha la possibilità di scegliere di "accendere" entrambe le finestre, oppure soltanto quella delle due desiderata. Viceversa, se la degenza ha un unico muro rivolto in facciata, soltanto una finestra rimane selezionabile, lasciando così la seconda parte dell'algoritmo "spento".

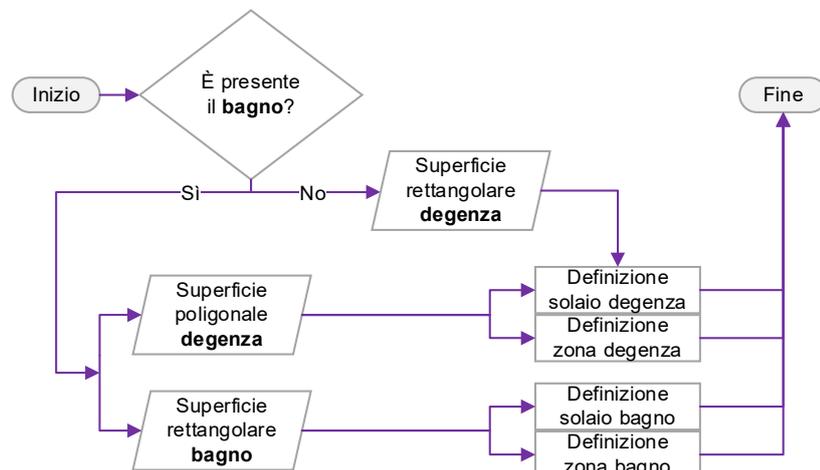
Una volta statuito quale finestra mantenere, restano dunque da definire gli stessi aspetti che caratterizzano la porta (vedi sopra). Anche in questo caso infatti è possibile scegliere il verso di apertura ed il filo della mazzetta, nonché stabilire la distanza dal muro, che nell'algoritmo è vincolata all'interno di un preciso intervallo. In questo caso, non essendoci standard per il dimensionamento delle finestre, è inoltre fornito all'utente l'onere di modificare i valori da cui dipendono tali **vincoli dimensionali**, in base all'ingombro del serramento preferito selezionato.

102 - Opzioni rese effettuabili su Grasshopper: finestre



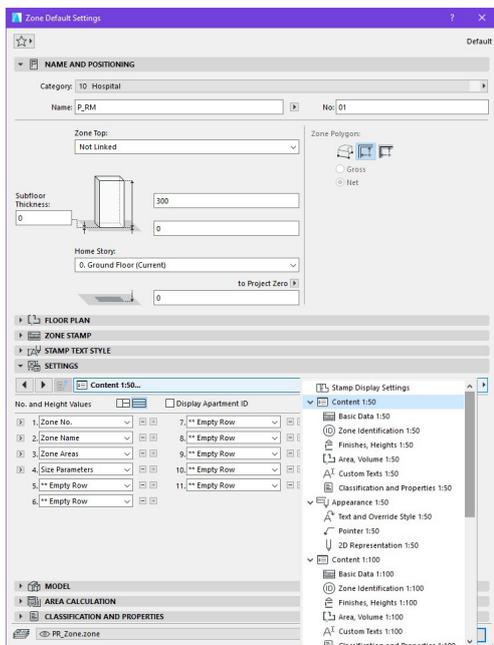
• Solaio, soffitto e zone

103 - Diagramma di flusso: elaborazione dei componenti "solaio", "soffitto" e "zona"

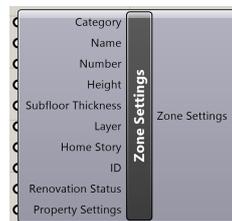


La porzione di algoritmo che genera il solaio lascia all'utente solo la definizione della stratigrafia. È di fatto pensato in modo da **calcolare in automatico**, in base all'eventuale presenza dell'area bagno, quale superficie risulti ricoperta dal locale degenza. Questa distinzione è resa necessaria dalla possibilità di **diversificare la stratigrafia** del bagno, solitamente con rivestimento in piastrelle, da quella della degenza, ricoperta da differenti superfici resistenti a lavaggio, disinfezione ed azione meccanica.

Lo stesso *input* geometrico che genera il solaio - o i solai - è impiegato per il componente delle **zone**, che, come per i serramenti visti in precedenza, possiede un ventaglio limitato di impostazioni su *Grasshopper*. Questa particolarità è dovuta stavolta al fatto che le proprietà di una zona su *ArchiCAD* vengono definite in base alla scala di visualizzazione. È pertanto necessario definire un **preferito** all'interno della piattaforma BIM, in cui specificare tali impostazioni, da poter poi impiegare nell'editor visuale.

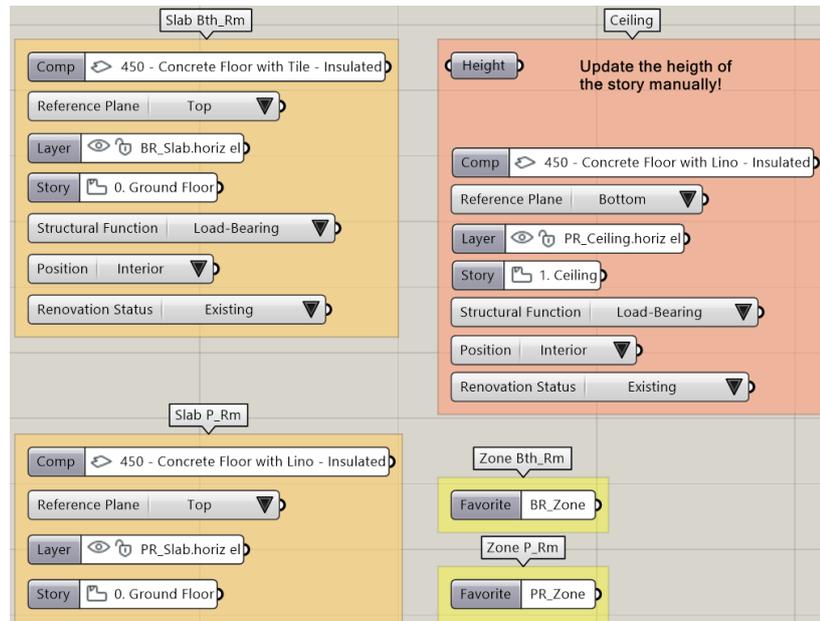


104 –Disponibilità delle proprietà di una zona: impostazioni di ArchiCAD (a sinistra) e di Grasshopper (a destra)



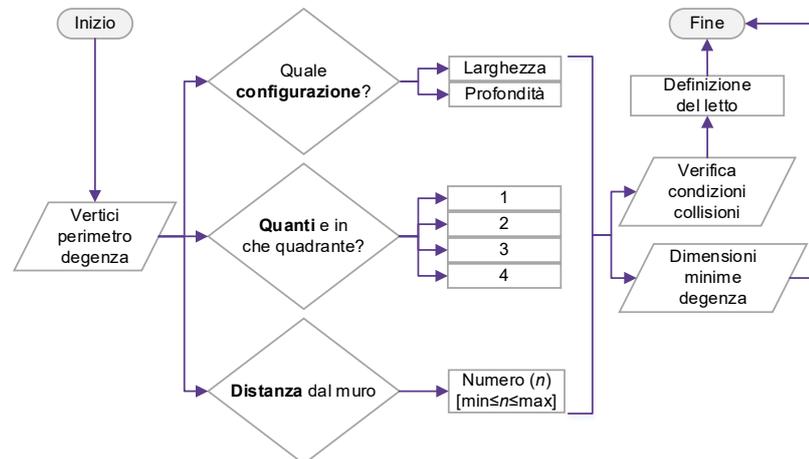
Partendo dalla definizione del solaio è inoltre realizzato il **soffitto**, come copia del poligono che costituisce l'intera area pavimentata. La sua **quota** impiega lo stesso *input* numerico (*slider*) dei muri, agevolando il flusso di lavoro. Sarebbe di fatto svantaggioso, oltre alla già citata incombenza della modifica manuale di tale valore, dover inoltre ripetere questa operazione per ogni componente che lo richieda.

105 - Opzioni rese effettuabili su Grasshopper: solai, soffitto e zone



106 - Diagramma di flusso: elaborazione del componente d'arredo "letto"

- Posti letto

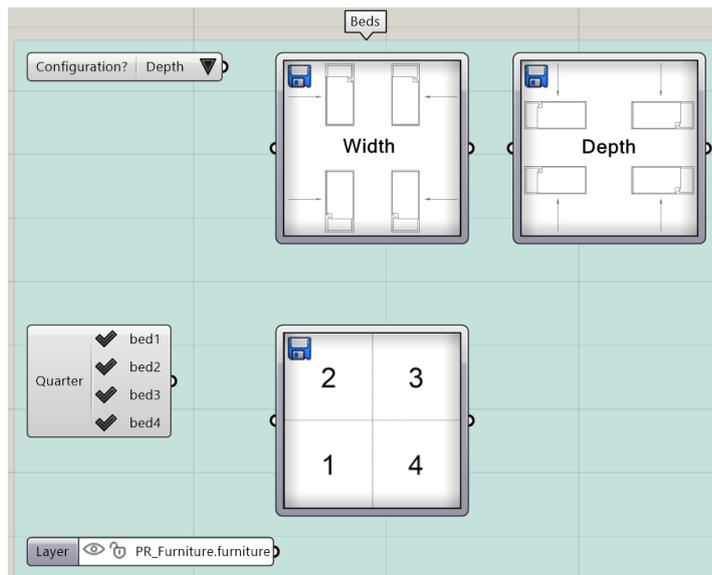


La caratteristica principale che contraddistingue una degenza, al di là dei suoi aspetti geometrico-volumetrici è il numero di posti letto al suo interno. Essendo oggetti di libreria, i letti sono legati ad un **punto di ancoraggio**, che è in questo caso realizzato come copia dei vertici del poligono che costituiscono la degenza. La direzione verso cui tali punti si muovono dipende dalla prima condizione posta, ovvero con quale **configurazione** si desidera predisporre la degenza.

Prendendo in esame gli schemi distributivi proposti nel Manuale dell'architetto [40], sono emerse due possibili soluzioni, che sono quindi riportate come prima opzione decisionale dell'algoritmo. Intendendo come "larghezza" un immaginario asse X di orientamento locale della degenza, e per "profondità" l'asse Y, la prima configurazione prevede che i letti siano disposti parallelamente alla dimensione di profondità della stanza. In questo modo essi si muovono perciò lungo la direzione della larghezza, mentre, viceversa, con la seconda configurazione, che vede i letti paralleli alla larghezza, gli oggetti si spostano in profondità.

La seconda condizione da porre, una volta stabilito l'orientamento dei letti, consiste nel determinare quanti pazienti effettivamente contenga la degenza. Per fare ciò, lo *script* visuale è qui sviluppato come una “**check list**” che, al contempo, risolve e racchiude due tipi di informazioni. Immaginando di suddividere la stanza in quadranti, numerati da 1 a 4, l'utente può scegliere in quali di questi posizionare i letti. Così facendo non viene determinato solo il loro punto di ancoraggio di riferimento, ma anche, indirettamente, la quantità di pazienti. Si risolve infatti, con un unico componente, la condizione che determina il **numero di posti letto** e una loro prima grossolana sistemazione all'interno della degenza.

La loro posizione può poi essere definita mediante degli *slider*, che, come di consuetudine, sono circoscritti nel loro movimento tra un minimo ed un massimo che dipendono dalla configurazione, dalla posizione e dalla quantità di letti. È infatti necessario in questo caso definire, oltre ai vincoli dimensionali che denotano la minima distanza dai muri, anche le condizioni che permettono il corretto spazio fra gli oggetti, come stabilito nel manuale [40].

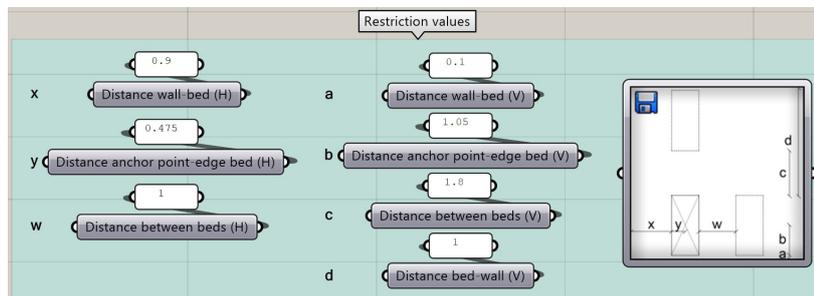


107 - Opzioni rese effettuabili su Grasshopper: posti letto

Una conseguenza implicita dei vincoli dimensionali legati ai posti letto descritti nella normativa è rappresentata dal fatto che la geometria della degenza stessa presenti un **dimensionamento minimo**, di cui però non è espresso alcun valore al di là della metratura. Grazie all'algoritmo creato è però semplice ricavare tali numeri basandosi sulla configurazione e sulla quantità dei posti letto, sulle dimensioni degli oggetti di libreria impiegati e sugli stessi vincoli inter-dimensionali della normativa. In questo modo l'utente ha un indice di grandezza affidabile su cui poter basare il progetto, dovendo comunque sempre tenere fede alle altre condizioni poste ed al buon senso. La modifica dei parametri che caratterizzano i letti diventa così strettamente legata alla geometria, che può quindi variare in base ad

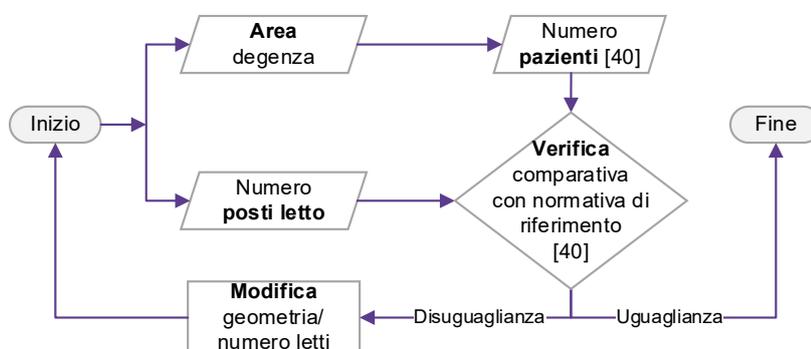
essi in quanto l'algoritmo dà priorità ai vincoli rispetto ai valori numerici immessi dall'utente. È quindi consigliabile definire in primo luogo i posti letto, per poi determinare con migliori risultati la geometria.

108- Vincoli dei posti letto su Grasshopper



109 - Diagramma di flusso: elaborazione delle verifiche da manuale [40]

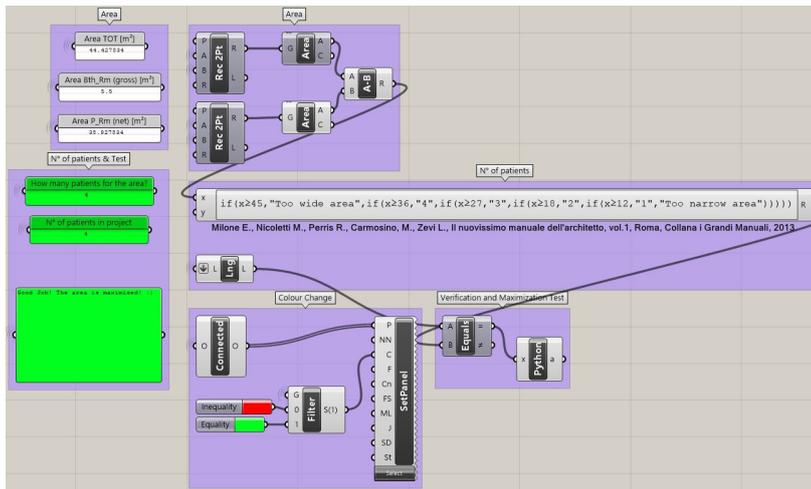
- **Calcolo e verifica di area e numero di posti letto**



Una volta definite le dimensioni della degenza e del numero di posti letto, un'apposita porzione dell'algoritmo è destinata a verificarne la coerenza da un punto di vista normativo. Per fare ciò è necessario **confrontare due valori**, di cui il primo deriva da una funzione che determina, in base all'area calcolata del poligono della degenza, il numero di pazienti che essa può ospitare da normativa [40]. Tale condizione nell'algoritmo viene così descritta:

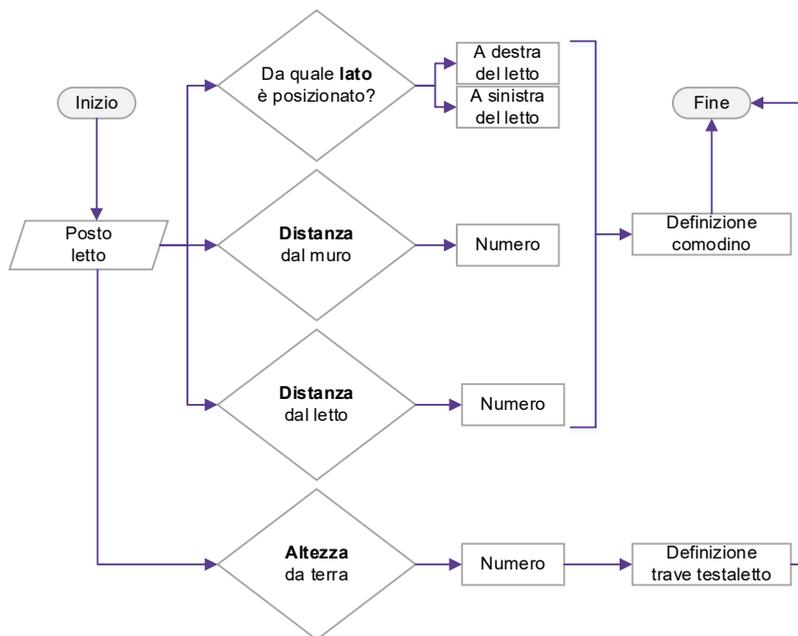
if (x ≥ 45, "Too wide area", if (x ≥ 36, "4", if (x ≥ 27, "3", if (x ≥ 18, "2", if (x ≥ 12, "1", "Too narrow area"))))

Il secondo valore rappresenta invece il numero di oggetti, ovvero la quantità di posti letto, che vengono selezionati all'interno della degenza. Questi due valori sono confrontati mediante un componente di **uguaglianza** che restituisce il proprio risultato ad un "panel", cioè ad una sorta di nota, che fa da tramite per comunicare all'utente se l'area è ottimizzata o se sia nuovamente necessario intervenire sulla geometria della degenza o sul numero di posti letto. Al *panel* è inoltre stato associato un componente di proprietà secondarie, appartenente al *plug-in MetaHopper*, grazie al quale se il riscontro del test risulta positivo, ovvero se i valori rispettano la normativa, è stato assegnato un **colore** verde, viceversa rosso.



110 - Calcolo e verifica di area e numero di posti letto su Grasshopper

- Arredo dipendente dai posti letto: **comodini e travi testaletto**



111 - Diagramma di flusso: elaborazione dei componenti d'arredo "comodino" e "trave testaletto"

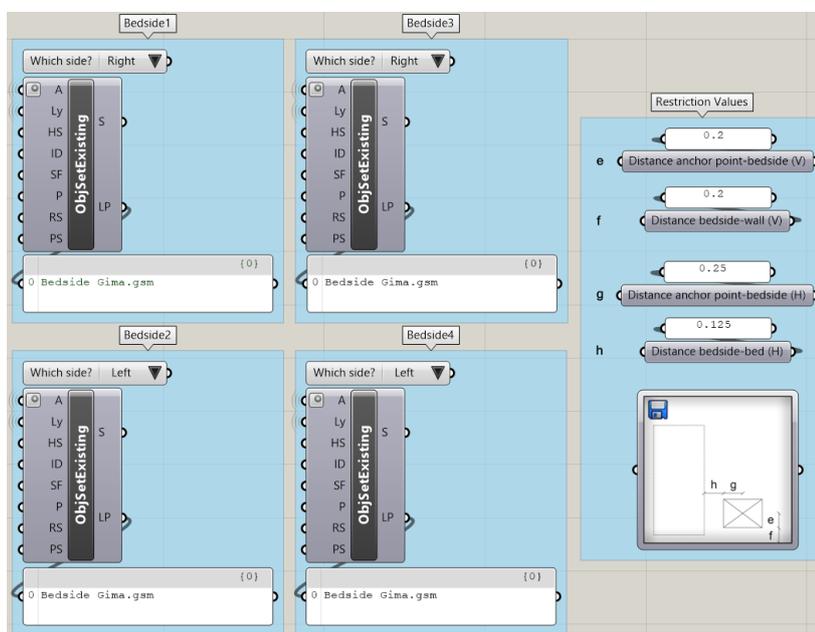
L'elenco di attrezzature minime presenti all'interno di una degenza sono, da normativa, relazionate ai posti letto. Questo legame è altresì mantenuto all'interno dell'algoritmo, dove gli arredi sono divisi in due tipologie: i primi dipendono dalla presenza e dalla posizione dei letti nello spazio, i secondi solo dalla loro quantità.

La prima categoria comprende i comodini e le travi testaletto, che sono accomunati dall'aver il punto di ancoraggio originato come una copia di quello del letto a cui sono legati. In questo modo entrambi gli oggetti vengono definiti solamente se è presente il punto da cui hanno origine. Ciò sottintende non solo che essi siano generati indirettamente, in quanto **oggetti subordinati** ai posti letto, ma anche che ogni letto selezionato nella *check list* comprenda inevitabilmente sia comodino che trave testaletto.

La normativa non pone particolari restrizioni riguardo al posizionamento di attrezzature secondarie il cui ingombro, o faci-

lità di spostamento, non ostacola il passaggio di barelle e lettighe. È quindi doveroso porre delle condizioni che permettano un **grado di libertà** decisionale più elevato rispetto ai casi precedenti. Per i comodini l'utente può infatti deliberatamente scegliere la distanza dal muro e dal letto a cui è vincolato. Per ognuno di essi può inoltre definire il lato preferenziale del letto da cui vuole posizionarli.

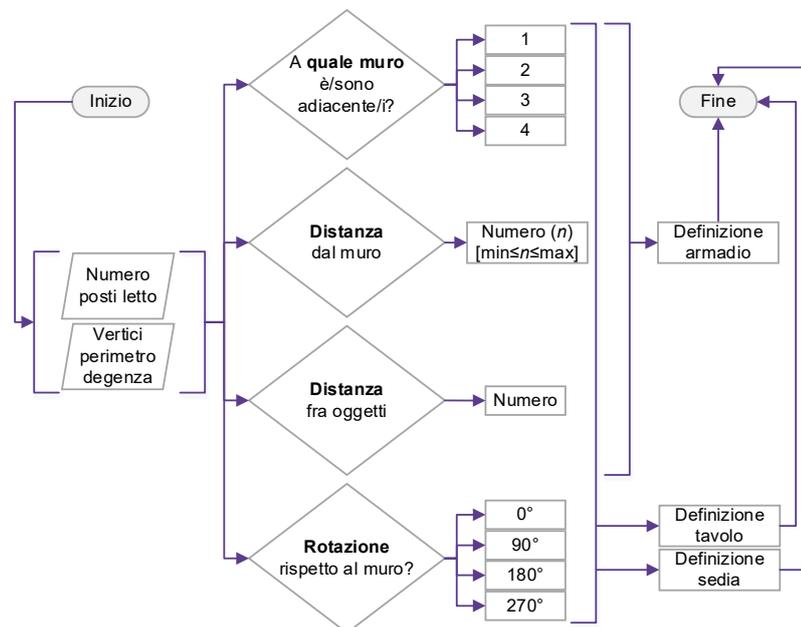
112 – Esempio di opzioni rese effettuabili su Grasshopper per l'arredo dipendente dai posti letto: comodini



Le travi testataletto, essendo generalmente progettate per stare centralmente sopra al letto del paziente, sono invece vincolate in mezzeria e l'unica opzione controllabile nello *script* è quella che riguarda la sua altezza da terra, che può di fatto variare in base al modello della stessa e del letto.

- Arredo dipendente dal numero di posti letto: **armadi, tavoli e sedie**

113 – Diagramma di flusso: elaborazione dei componenti d'arredo "armadio", "tavolo" e "sedia"



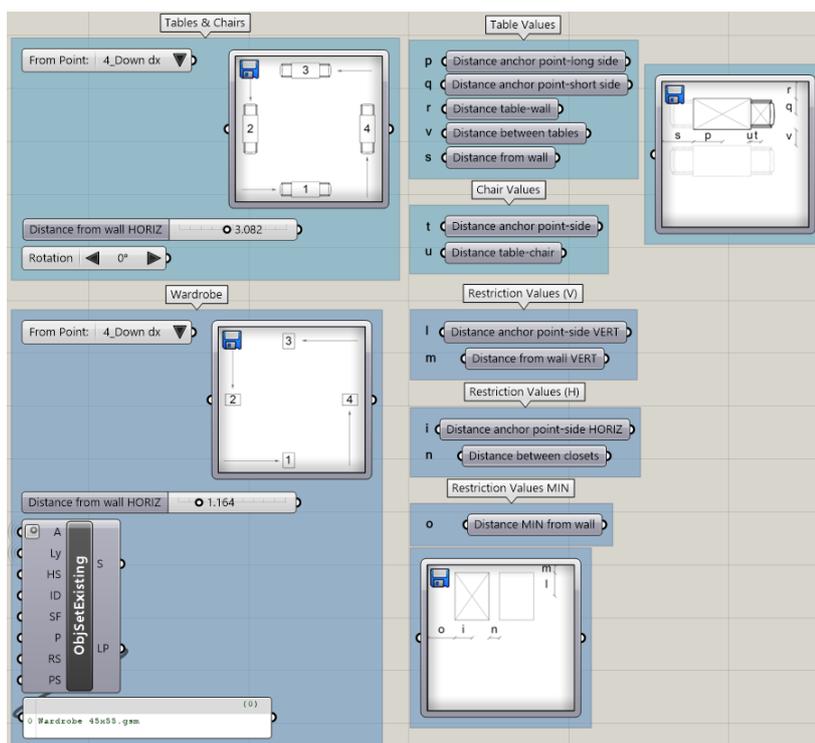
Gli armadi, i tavoli e le sedie appartengono alla seconda categoria di attrezzature dipendenti dai posti letto. Contrariamente al caso precedente, non sono però oggetti ad essi subordinati nello spazio, bensì il loro numero dipende dalla **quantità di letti** presenti nella degenza. Per rispecchiare tale equivalenza, nell'algoritmo il numero degli arredi in analisi si incrementa all'aumentare dei posti letto e viceversa.

Analizzando gli schemi distributivi proposti nel manuale dell'architetto [40], si può notare che, generalmente, armadi, tavoli e sedie sono posti perimetralmente all'interno della stanza. Pertanto, i loro punti di ancoraggio sono generati come **copie dei vertici** che circoscrivono la degenza.

Le tre tipologie di arredo sono accomunate dalle condizioni create nell'algoritmo allo scopo di definire a quale **muro** renderli adiacenti e a che distanza da esso devono collocarsi. Come in altri casi precedenti il margine entro cui avviene tale spostamento è vincolato parametricamente in base alla parete selezionata e alla sua lunghezza. Si ritiene inoltre necessario porre la possibilità di decidere a che distanza stiano gli oggetti fra loro, ovvero lo **spazio interposto** fra gli armadi, fra i tavoli e fra tavoli e sedie. Questi ultimi in particolare sono sviluppati all'interno dello *script* visuale come un "**blocco**" tale per cui non è possibile svincolare una sedia dal tavolo a cui essa è assegnata. Suddetta scelta dipende dal fatto che la normativa indichi un minimo di una sedia per paziente, individuandone la funzione non solo ad uso dell'ospite che si reca a visitare il ricoverato, bensì primariamente al pranzo del paziente stesso. Per questo motivo l'unione di tavolo e sedia è qui da intendersi come una **seduta per il pasto**, che coincide al tempo stesso con il quantitativo minimo di attrezzature sancito nel manuale [40]. La traduzione di questa condizione all'interno dell'algoritmo prevede la generazione di un solo tavolo fino ad un massimo di due posti letto, con annesse una o due sedie in base al numero di pazienti. Il secondo tavolo da pranzo, con le altre sedie, si genera allo stesso modo superando la precedente quota fino a colmare il numero massimo di pazienti consentito.

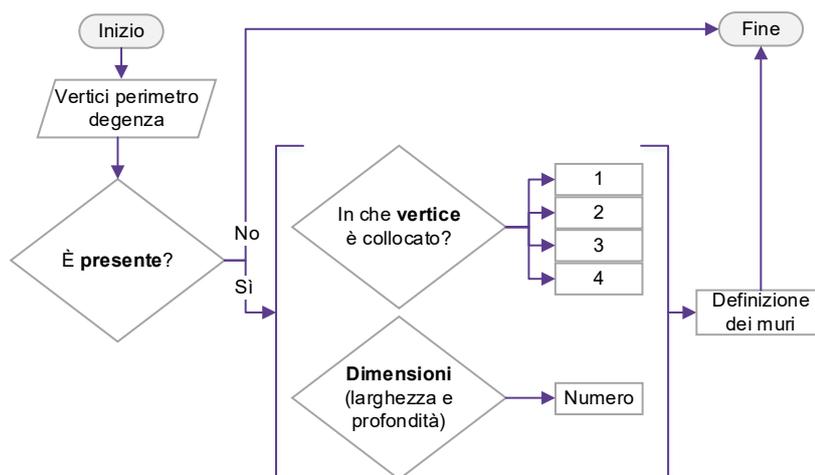
Per quanto riguarda tavoli e sedie, la maggiore libertà di posizionamento rispetto agli armadi, che sono obbligatoriamente rivolti con l'anta verso il centro della stanza, rende doveroso porre un'ulteriore condizione. Il blocco delle sedute presenta pertanto un'apposita porzione di *script* che sancisce l'**angolo di rotazione** in base al muro a cui esso è adiacente. Questo rende tale elemento, che concerne un considerevole ingombro all'interno della degenza, più **versatile** e adattabile allo spazio da modellare.

114 – Esempio di opzioni rese effettuabili su Grasshopper per l'arredo dipendente dal numero di posti letto: tavoli, sedie e armadi



115 – Diagramma di flusso: elaborazione del servizio sanitario

- Bagno annesso al locale degenza

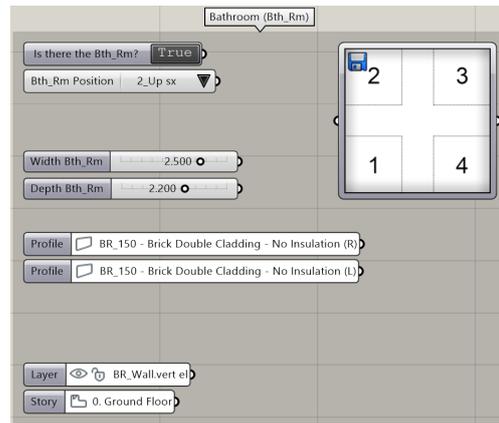


L'esigenza di includere un servizio igienico al locale degenza è, da manuale, resa obbligatoria ogni quattro posti letto. Questo implica che non tutte le stanze debbano possederne uno, pertanto, il primo requisito da porre, all'interno dell'algoritmo, è la **condizione di esistenza** che ne sancisca la presenza.

L'area del bagno, se aggiunta, viene generata grazie ad un poligono rettangolare i cui punti perimetrali sono costituiti da copie di uno dei vertici che racchiude la degenza. La prima opzione da selezionare è quindi in **quale angolo** collocare il servizio, scegliendo fra le quattro alternative rese disponibili nell'algoritmo.

Una volta definite anche le **dimensioni**, in larghezza e profondità dell'ambiente, è infine possibile stabilire, come già visto per i muri della degenza, la **stratigrafia** che ne caratterizza il perimetro. In questo caso però, le sole pareti editabili, da que-

sto punto di vista, sono le due generate dalla nuova porzione di *script* attivata, la quale permette di chiudere il poligono in analisi. Le altre due murature mantengono la loro appartenenza al perimetro principale generato, rimanendo modificabili esclusivamente come stratigrafie proprie della degenza. Ciò significa che non è possibile modificare i muri costituenti l'intero perimetro del bagno, in quanto le pareti, che prima dell'accertamento della condizione di esistenza del servizio appartenevano alla sola zona degenza, restano indifferenti a questa scelta.



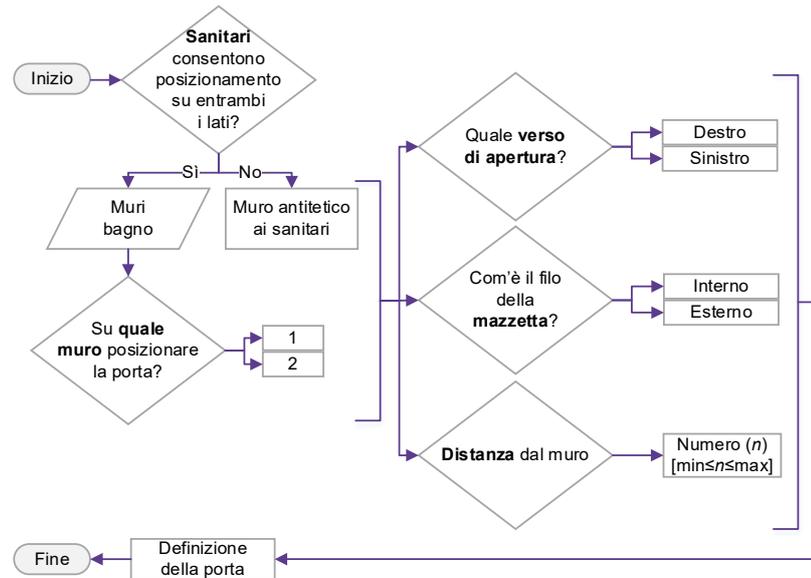
116 - Opzioni rese effettuabili su Grasshopper: bagno

Come analizzato per i muri perimetrali della degenza, si ricorda che la generazione di questi ultimi è definita in modo da mantenere il filo interno allo scopo di calcolare l'area netta della degenza stessa. Questa scelta, effettuata per mantenere monitorabile la superficie, soggetta a restrittivi vincoli normativi, ricade inevitabilmente sulla **metratura** del bagno, che risulta comprensiva dei due nuovi muri.

La presenza del bagno rappresenta uno dei **vincoli più frequenti** all'interno delle altre porzioni dell'algoritmo. Quando infatti viene generata la zona servizio, come anticipato, l'area della degenza si modifica di conseguenza. Questo implica che ogni componente appartenente alla degenza modifichi in modo parametrico l'**intervallo** entro cui potersi muovere in base non solo all'eventuale presenza del bagno, ma anche alla sua posizione spaziale. Inoltre, se un oggetto ha come origine lo stesso **vertice** del perimetro della degenza in cui è posizionato il bagno, il suo punto di ancoraggio deve essere programmato per copiare non il medesimo nodo da cui nasce il poligono del servizio, bensì quello che, sullo stesso asse, perimetra la zona degenza. In questo modo non solo si annulla la possibilità che tale oggetto possa sconfinare nell'area del bagno, ma viene inoltre fornita maggiore **consapevolezza delle distanze**, consentendo di calcolare le quote dal primo muro ad esso adiacente, anziché da quello del perimetro principale.

117 – Diagramma di flusso: elaborazione del componente “porta”

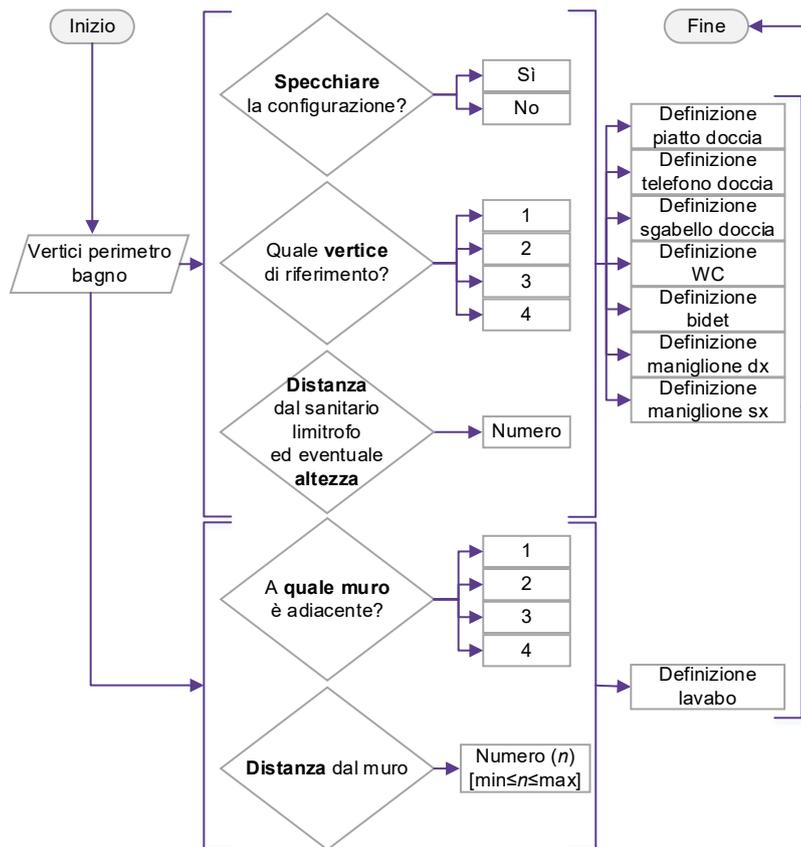
- Porta del bagno



Il posizionamento della porta del bagno può avvenire solo sui due muri interni della zona stessa che sono definiti, nel paragrafo precedente, come le uniche pareti proprie del servizio. Ciononostante esso dipende anche dalla disposizione interna dei **sanitari**, che ospita al suo interno, che verranno meglio definiti in seguito. Se di fatto tali oggetti sono allineati in adiacenza ad uno dei due muri opzionabili, la scelta risulta vincolata alla seconda parete rimanente. Se, al contrario, i sanitari sono a ridosso di uno dei muri della degenza, allora è lasciata maggiore libertà nella scelta del muro su cui posizionare la porta.

Una volta definita, in modo volontario o indirettamente, la parete che ospita il serramento, sono qui ripresentate le stesse condizioni ed opzioni già proposte per la porta d'ingresso della degenza (vedi sopra). Come per il precedente serramento, anche in questo caso il **senso di apertura** è posto come vincolo non modificabile, ad eccezione del fatto che da normativa esso deve essere rivolto verso l'esterno anziché verso l'interno.

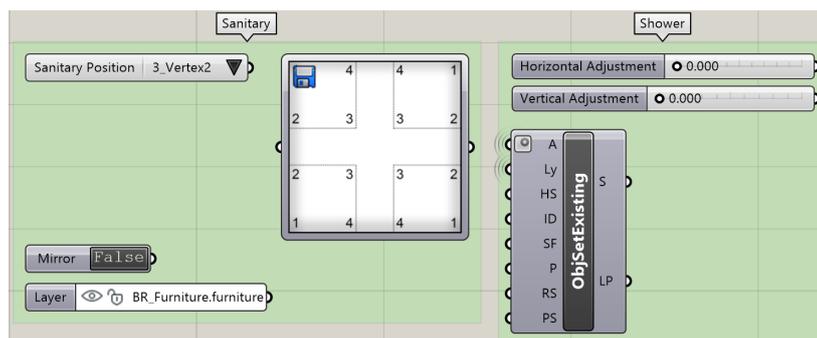
- **Attrezzature igienico-sanitarie del locale bagno**



118 – Diagramma di flusso: elaborazione dei componenti d'arredo “piatto doccia”, “telefono doccia”, “sgabello doccia”, “WC”, “bidet”, “maniglioni”

La normativa prevede l’inserimento di una serie di attrezzature igienico-sanitarie nel locale bagno. Non essendo specificata, però, nessuna particolare configurazione, nell’algoritmo viene sviluppata **una delle possibili distribuzioni**. Essa consiste nel gestire in un unico “blocco” tutti gli elementi, immaginandoli così disposti su un’unica parete, ad eccezione del lavabo, a cui è concesso invece un maggiore grado di libertà.

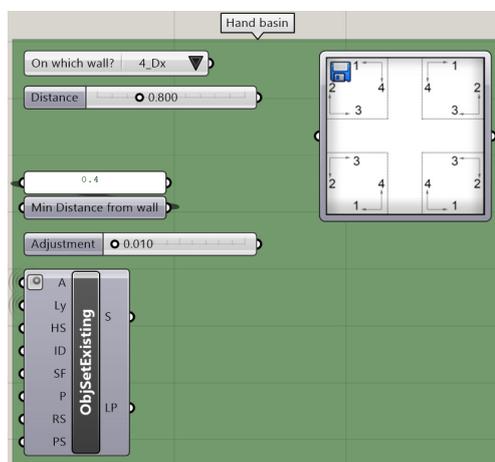
Il **blocco** comprende tutti gli elementi appartenenti al box doccia, il WC, il bidet ed i maniglioni di sostegno. L’intero gruppo viene definito a partire da uno dei quattro vertici appartenenti al perimetro del bagno, lasciando inoltre la libertà di specchiare tale configurazione rispetto al punto selezionato. Al fine di semplificare le operazioni compositive, in pianta le distanze fra i sanitari sono definite in modo parziale, ovvero misurando la quota dall’oggetto limitrofo, anziché quella totale che lo distanzia dall’origine.



119 - Opzioni rese effettuabili su Grasshopper: blocco attrezzature igienico-sanitarie, e porzione esemplificativa della doccia

Per quanto riguarda il **lavabo**, anch'esso possiede un punto di ancoraggio generato come copia di uno dei vertici perimetrali. Inoltre, come per precedenti soluzioni già analizzate, esso è libero di essere posizionato singolarmente su una delle quattro pareti opzionabili. È inoltre possibile spostare il lavabo all'interno di un intervallo vincolato in modo che il suo ingombro non sconfini all'esterno dell'area del bagno.

120 - Opzioni rese effettuabili su Grasshopper: lavabo



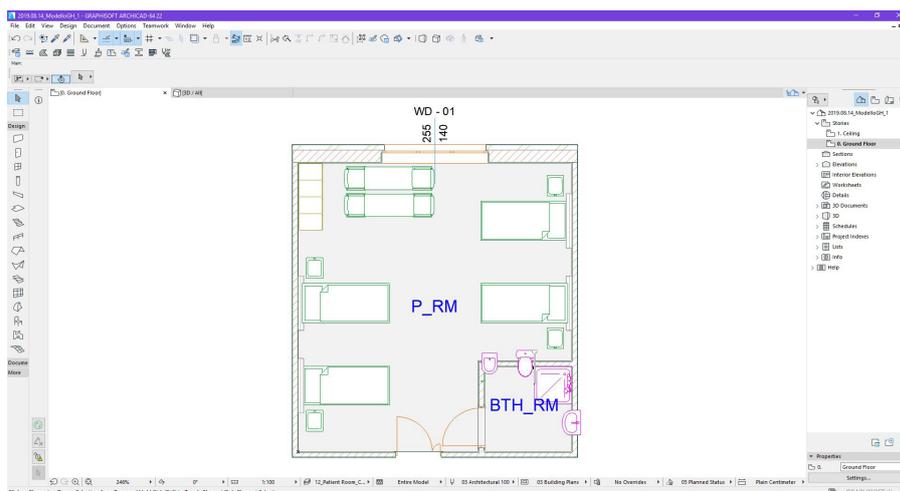
Essendo la **superficie del locale lorda**, è lasciata la possibilità di modificare la distanza degli oggetti dal muro nel caso in cui essi vengano posti su uno dei due muri generati a filo esterno del perimetro del bagno. Per come è stata configurata la totalità delle attrezzature, è necessario effettuare questo tipo di “**aggiustamento**” due volte, una per il blocco di sanitari e una per il lavabo. Questa ripetizione è resa necessaria dal fatto che i due gruppi di elementi siano svincolati fra loro, pertanto non si può dare per scontato che entrambe le tipologie necessitino contemporaneamente dell'operazione di pareggiamento dal muro.

2.2.2 Dallo script al BIM: revisione del modello

La modifica dei parametri resi editabili sul software di *visual scripting Grasshopper*, grazie alle condizioni poste all'interno dell'algoritmo sviluppato, consente un immediato riscontro su *ArchiCAD*, la piattaforma BIM scelta in questa fase del progetto di ricerca. Il *plug-in* impiegato permette infatti, mediante un sistema di *live-connection*, la semplificazione del flusso di lavoro delle prime fasi decisionali del progetto. Ciononostante, per ottenere una congrua visualizzazione degli elementi così generati, è necessario intervenire, non solo sul programma utilizzato per la parametrizzazione, ma anche sulla piattaforma BIM stessa. Sono infatti da considerare diversi aspetti, sia informativi che grafici, che rappresentano il limite oltre cui *Grasshopper* da solo non può operare. Per questo motivo, una volta effettuate le sopracitate scelte disponibili nell'algoritmo, diventa necessario porre le attenzioni alla piattaforma BIM su cui si genera il modello.

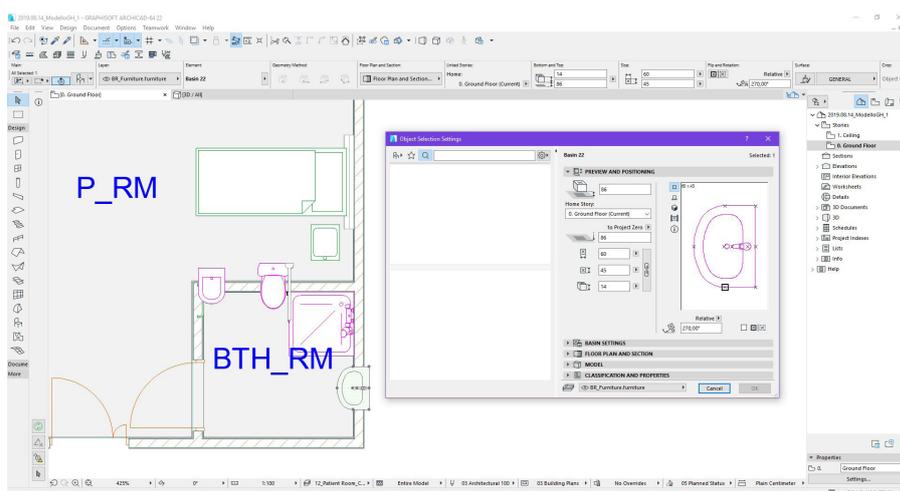
Operando tramite i soli parametri su *Grasshopper*, la definizione

del modello che così si ottiene su *ArchiCAD* presenta delle “anomalie” a cui è necessario porre rimedio manualmente.



121 – Generazione del modello da Grasshopper ad ArchiCAD: esempio di errori nella definizione geometrica

In primo luogo, la collocazione dei **punti di ancoraggio** potrebbe non rispettare i vincoli posti. Nell’algoritmo realizzato, infatti, generalmente gli oggetti sono pensati per avere l’origine posta centralmente all’ingombro proiettato in pianta. Questa scelta consente di facilitare la manipolazione dei parametri, potendo avere un più efficace controllo sulle singole condizioni rispetto ad una soluzione che non vede l’ancoraggio in mezzeria. Talvolta però gli oggetti di libreria perdono questa impostazione, che bisogna quindi ripristinare su *ArchiCAD*. Per fare ciò è sufficiente per prima cosa sbloccare l’elemento in questione, poiché ogni oggetto generato da *Grasshopper* viene, per impostazione predefinita, bloccato al fine di evitare modifiche accidentali degli stessi da parte dell’utente. Bisogna infine aprire le proprietà dell’oggetto e procedere con la selezione del nuovo punto di ancoraggio desiderato.

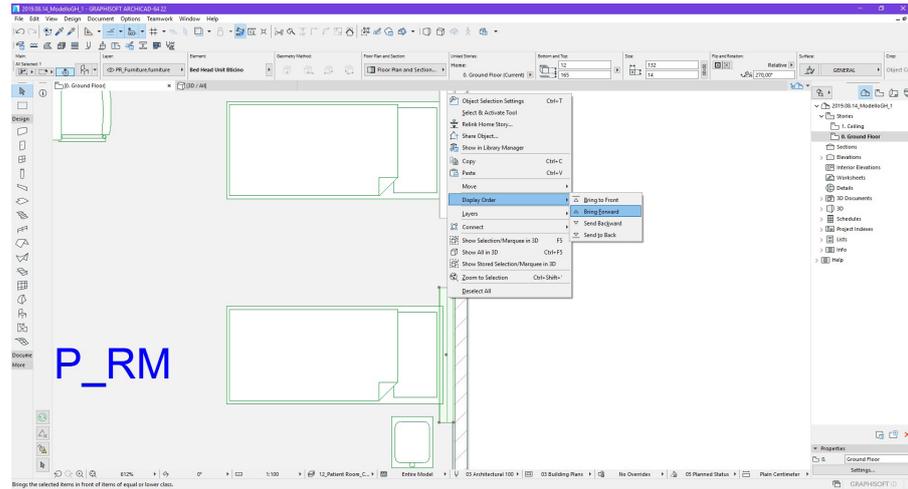


122 – Operazione di ripristino del punto di ancoraggio di un oggetto

La seconda operazione sugli oggetti di libreria su cui potrebbe essere necessario intervenire è l’**ordine di visualizzazione** in pianta. Alcuni arredi infatti, come nel caso delle travi testaleto, nonostante siano correttamente posizionati ad un’altezza superiore rispetto agli oggetti sottostanti, possono essere visualizzati ad un livello inferiore. In questo caso si procede a modificarne l’ordine di visualizzazione avendo cura di procedere “per *step*”, evitando l’opzione di posiziona-

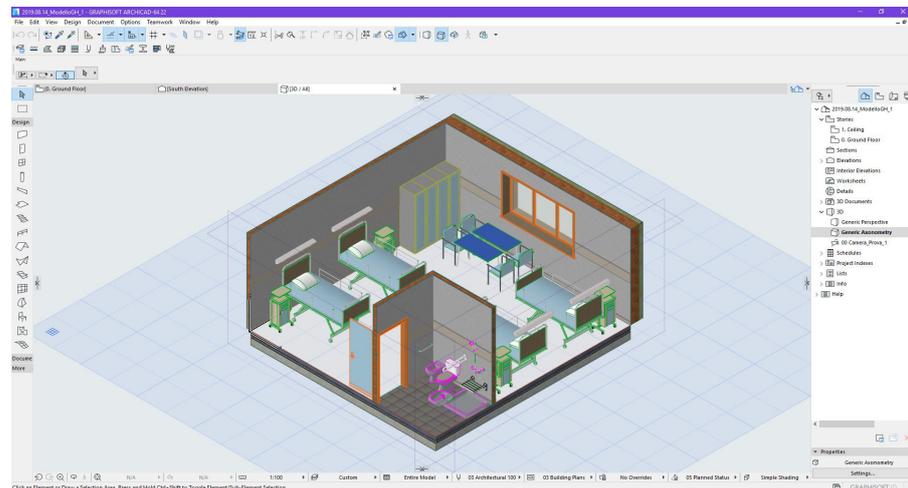
mento dell'oggetto in primo piano rispetto all'intero modello.

123 – Modifica dell'ordine di visualizzazione di un oggetto in pianta

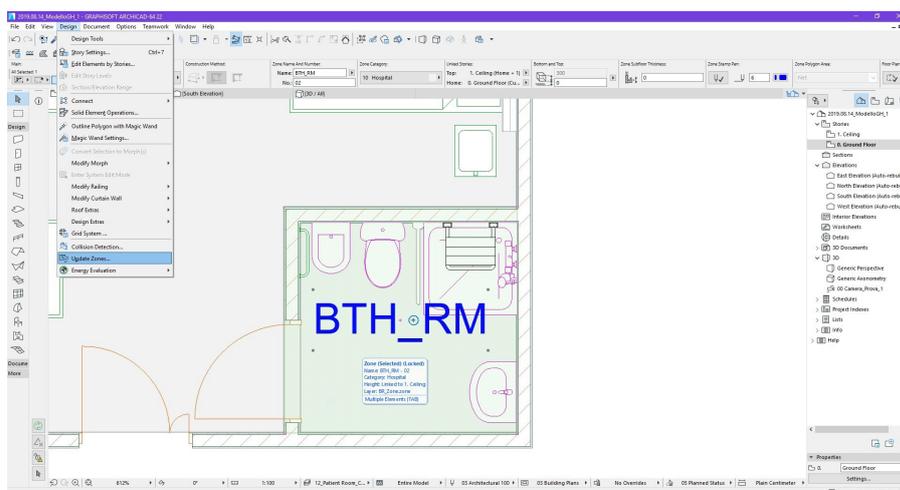


Le due operazioni appena descritte possono essere effettuate anche durante la fase di modifica dei parametri su *Grasshopper*, al fine di migliorare l'anteprima del modello generato. Altri provvedimenti richiedono invece di essere effettuati solo una volta terminata la connessione con il software di *visual scripting*. Questo dipende dal fatto che, come anticipato nel capitolo precedente, alcuni *input* geometrici siano necessariamente da perfezionare in una fase successiva. Fra queste ad esempio si trova la **definizione dei solai**, che vengono generati su *Grasshopper* tramite lo stesso perimetro che dà origine ai muri. Questi ultimi, essendo definiti a filo interno della degenza, risultano in questo modo senza appoggio, in quanto l'area del solaio termina esattamente nello stesso punto in cui si innalzano i muri. È quindi necessario agire direttamente sulle dimensioni dei solai su *ArchiCAD*, una volta terminata la fase di definizione dei parametri. Porre rimedio a questa incongruenza prima di aver terminato la connessione con *Grasshopper* non porterebbe infatti a nessun risultato. L'area pavimentata tornerebbe in pochi secondi a dare priorità ai parametri impostati nello *script* ripresentando nuovamente lo stesso problema. Questo "errore" è in realtà pensato per essere aggirato nella fase di inserimento della degenza all'interno di una struttura sanitaria. Si ipotizza infatti che i solai vengano definiti come un unico elemento del piano anziché singoli elementi suddivisi per area.

124 – Errore di generazione dei solai che lasciano i muri senza basamento



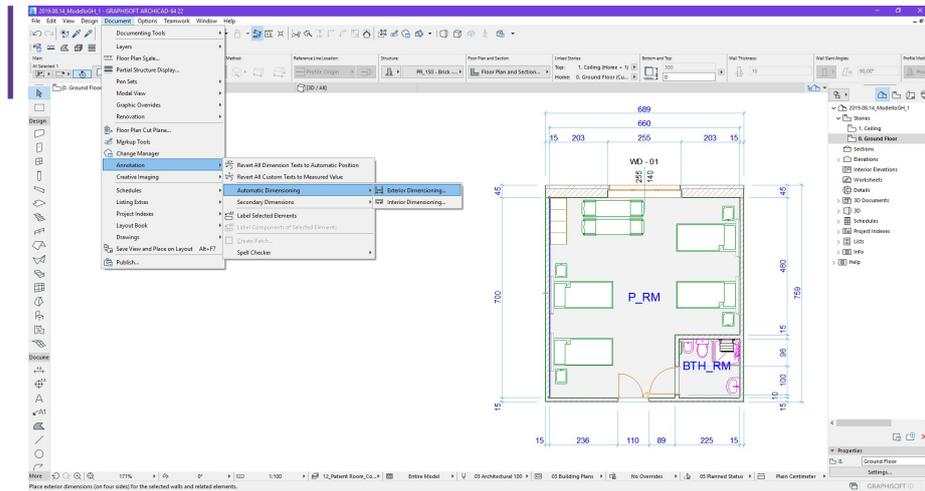
Un'analoga questione si riscontra nella **definizione della zona del bagno**, che viene generata, da *Grasshopper*, tramite il perimetro che origina i due muri con attinenza al servizio. Essa risulta pertanto comprensiva dell'area che costituisce l'impronta a terra di tali pareti. Anche in questo caso la correzione del problema si effettua direttamente su *ArchiCAD* e la sua reiterazione dipende dalla riattivazione della connessione con lo *script* che lo genera. Per rendere l'area netta si può agire in due modi: dopo aver sbloccato la zona è possibile modificarne manualmente la superficie di pertinenza oppure utilizzare lo strumento di aggiornamento automatico delle zone. Con il secondo metodo la piattaforma dimostra di saper calcolare correttamente l'area netta, anche se dal principio non è in grado di farlo in quanto vincolata ai parametri dell'algoritmo.



125 – Zona con calcolo dell'area lorda, vincolato ai parametri definiti su Grasshopper, prima della correzione dell'errore

Per completare infine le informazioni grafiche da visualizzare in pianta è possibile aggiungere le **quote** su *ArchiCAD*. Come per le precedenti, per ottimizzare il flusso di lavoro, questa operazione è preferibilmente da effettuare una volta definita come minimo la geometria sommaria della degenza. I nodi inseriti attraverso lo strumento di annotazione sono effettivamente dinamici e seguono parametricamente il punto a cui sono legati. Ciononostante, l'algoritmo che genera il modello è progettato per avere la possibilità di variare non solo le dimensioni, ma anche la presenza e la posizione di alcuni volumi, come quello del bagno annesso alla degenza. Pertanto, associando prematuramente dei nodi di quotatura ad un elemento così tanto variabile, si incorrerebbe nel rischio di dover eliminare o inserire nuovamente tali nodi, oppure di dover spostare le linee di quota, in quanto adiacenti ad un lato differente rispetto a quello della geometria a cui si riferiscono. Per ottimizzare ulteriormente il processo di annotazione, invece di inserire manualmente i nodi delle quote, è inoltre possibile sfruttare lo strumento di quotatura automatica.

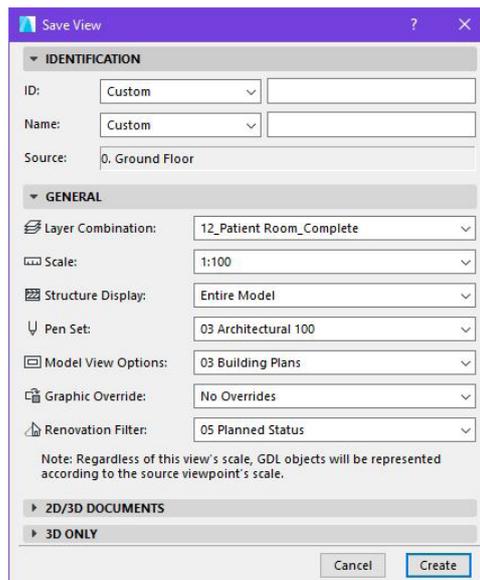
126 – Operazione di quotatura su ArchiCAD



2.2.3 Dallo script al BIM: opzioni di visualizzazione per la creazione di un *template*

La modifica delle incongruenze riscontrate nel modello BIM, generato tramite il software di programmazione visuale, non è sufficiente a rendere la visualizzazione del progetto consona alla rappresentazione di un disegno architettonico. È quindi necessario predisporre il modello attraverso la creazione di un **template**. Una volta definite tutte le impostazioni per la visualizzazione del progetto sarà infatti possibile, dopo aver svuotato il documento dalle geometrie generate da *Grasshopper*, salvarlo come file *.tpl* da impiegare come base per il trasferimento dei dati parametrici dell'algoritmo. La strada che porta alla creazione del *template* comprende le questioni che sono riassunte da *ArchiCAD* nella finestra di salvataggio di una vista. Essi rappresentano infatti tutti i punti da soddisfare per rifinire la visualizzazione di un modello.

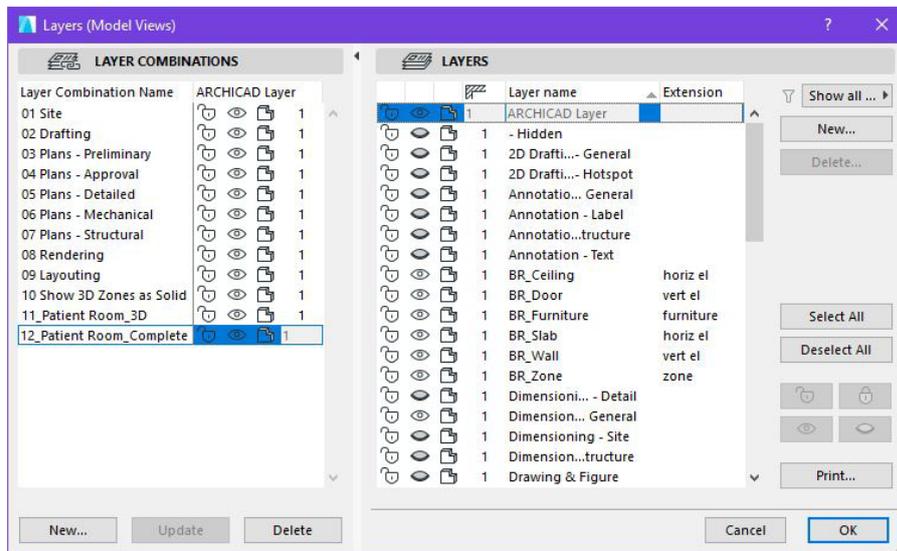
127 – Finestra di salvataggio di una vista: caposaldi per la creazione di un *template*



Il primo tema da affrontare è la **combinazione di lucidi**, che consente la gestione dei *layer* contenuti nel documento. I lucidi si comportano come dei fogli di carta da lucido, appunto, e rappresentano il "livello" su cui un oggetto giace. Essi sono impiegati per organizzare

in modo logico i vari elementi all'interno del progetto e le loro combinazioni permettono di ordinarli in gruppi capaci di memorizzarne lo stato, ovvero le impostazioni che ne sanciscono ad esempio la visibilità o il blocco.

In questo caso si decide di creare dei nuovi lucidi per gli elementi generati dalla piattaforma di *visual scripting*, e di impiegare alcuni di quelli già presenti nel *template* fornito assieme alla versione di *ArchiCAD 22* per gli oggetti di uso comune come le quote o le linee di sezione, creando così una nuova combinazione personalizzata.



128 – Esempio di combinazione di lucidi definito su ArchiCAD

Il secondo aspetto fondamentale da affrontare per una corretta visualizzazione del modello è quello dei **set di penne**. Ad ogni elemento, in *ArchiCAD*, viene automaticamente attribuito un valore numerico che serve a determinarne l'appartenenza ad una categoria di oggetti. Questo numero è visualizzabile all'interno delle proprietà di ogni elemento, ma è consigliabile non modificarlo in quanto il valore attribuito all'oggetto gli fornisce una specifica collocazione all'interno di una tabella, grazie alla quale è possibile gestire le penne.

	General	2D-Elements	OBJECTS	OPENINGS	Annotations	Text	WALLS	COLUMNS	SLABS	BEAMS	STAIRS	ROOFS	MESH	ZONES	Mechanical	CELLINGS	Electrical	Foreground Colors	Background Colors	Symbols
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
Structural Cut lines	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Non Structural Cut lines	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Separator lines	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
Uncut lines/Outlines	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
Foreground Color	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
Background Color	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
Structural Cut lines	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
Non Structural Cut lines	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
Separator lines	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
Uncut lines/Outlines	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
Foreground Color	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
Background Color	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255					

129 – Tabella della definizione delle categorie di elementi per la gestione delle penne [79]

Le colonne della tabella penne sono suddivise per elementi generali, a ciascuno dei quali è assegnato un gruppo di colori, ad esempio alla settima colonna c'è l'elemento "muri", a cui, nella tabella soprastante, è assegnato il rosso. Ogni riga identifica invece la funzione ancora più

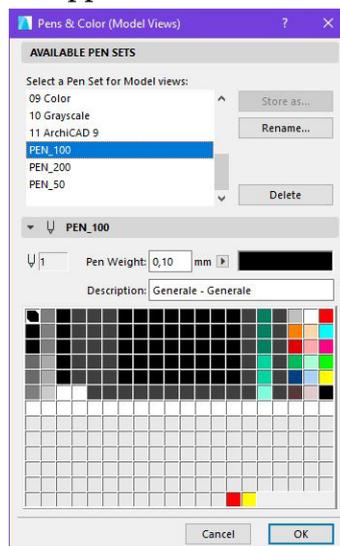
specifica del tipo di tratto che rappresenta l'elemento, ad esempio una linea assimilata ad un oggetto sezionato, che presenta quindi un maggiore spessore. L'incrocio di righe e colonne porta al valore numerico che identifica l'elemento specifico, al quale è possibile assegnare una penna.

Le penne rappresentano quindi il modo in cui un elemento viene visualizzato, ovvero lo spessore ed il colore della linea che lo descrivono in una vista. È quindi sufficiente attribuire un differente colore ed uno spessore ad ogni specifico valore della tabella al fine di modificare la visualizzazione di un elemento senza doverne cambiare la categoria di appartenenza nelle proprietà dello stesso.

I set di penne, in analogia con la combinazione di lucidi, servono a salvare una serie di impostazioni che permettono di produrre velocemente disegni rappresentati diversamente partendo dallo stesso modello. Più set vengono personalizzati, più modalità di visualizzazione differenti possono essere ottenute.

Il set di penne impiegato nel file in analisi è pensato per ottenere documenti in scala di grigi, con spessori consoni alla scala grafica 1:100 con cui si desidera approcciarsi.

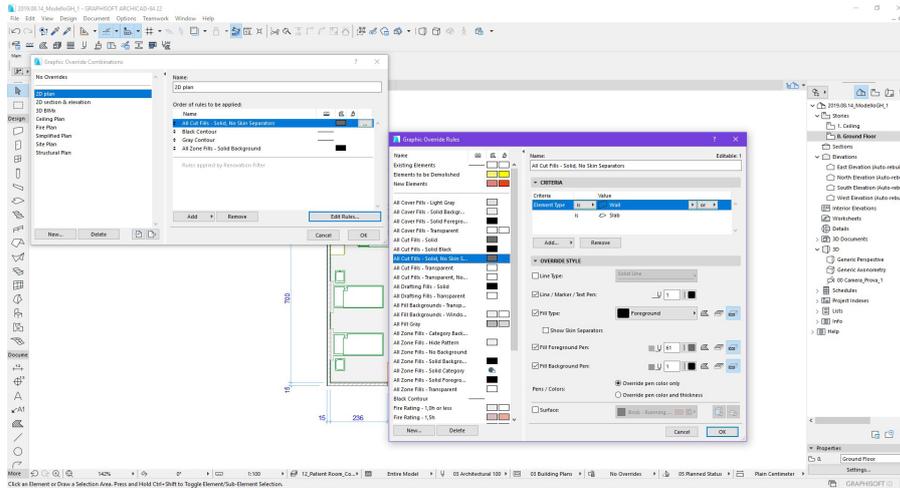
130 – Esempio di set di penne realizzato su ArchiCAD



L'ultima questione affrontata riguardo al tema di visualizzazione per la creazione del *template* è quella delle **sovrascritture grafiche**. Questo strumento è in realtà polivalente e può essere impiegato sia a scopi grafici, come in questo caso, che per verifiche o confronti visivi immediati riguardo alle informazioni degli elementi nel modello.

Una sovrascrittura è composta da una serie di regole, la cui priorità l'una sull'altra è definita dall'ordine in cui vengono posizionate. Ciascuna regola possiede criteri e stili editabili: i primi rappresentano gli elementi oggetto della sovrascrittura, i secondi individuano invece l'aspetto vero e proprio della sovrascrittura. Grazie a questa combinazione di impostazioni risulta quindi semplice rappresentare i differenti elementi in base al loro stato di ristrutturazione, individuare con colori diversi quelli che non rispecchiano determinati dimensionamenti, indicare con retini particolari stratigrafie che si desidera evidenziare o ancora contrassegnare gli elementi con determinate caratteristiche termiche, acustiche o antincendio.

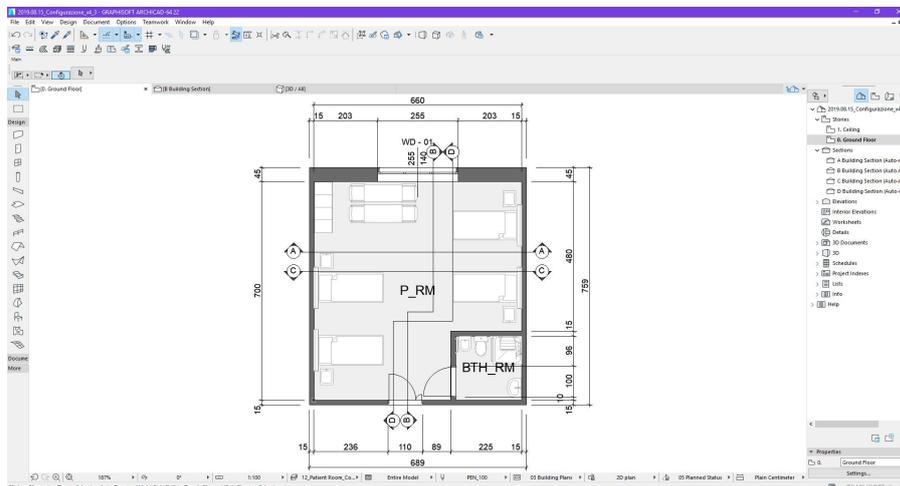
Nel file in analisi sono pensate tre differenti sovrascritture, messe in pratica rispettivamente in pianta, sezioni e viste tridimensionali, al solo scopo di visualizzare il modello in scala di grigi, con tratteggi pieni nelle porzioni sezionate.



131 - Esempio di sovrascrittura grafica realizzata su ArchiCAD: dettaglio di una regola, con i suoi criteri e stili

Un *template*, oltre alla predisposizione di impostazioni di visualizzazione, necessita anche della preparazione di una serie di viste, abachi e tavole. I primi due costituiscono delle **modalità di rappresentazione** di un modello BIM, le quali vengono poi organizzate all'interno di tavole che costituiscono il vero e proprio spazio carta digitale, pronto per essere stampato.

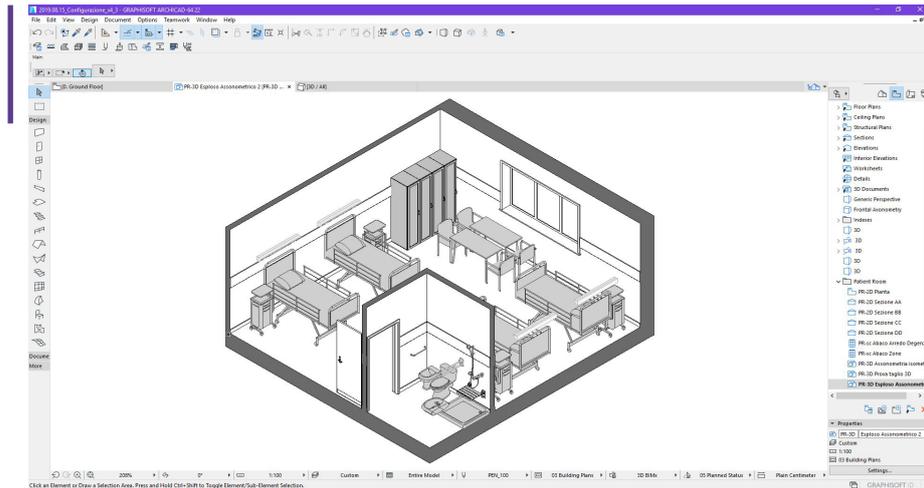
Le **viste 2D** che possono risultare utili al modello in analisi sono quelle delle piante, che vengono già generate automaticamente assieme alla creazione dei livelli, e quelle delle sezioni. I prospetti in questo caso non racchiuderebbero informazioni utili, essendo la degenza per ora estrapolata dal contesto della struttura sanitaria a cui dovrebbe essere destinata. L'indicazione, in pianta, delle linee che originano le viste delle sezioni può essere soggetta a piccoli cambiamenti in base alle volumetrie generate da *Grasshopper*. Ciononostante effettuare tale operazione per la creazione del *template* risulta comunque conveniente al processo di generazione degli elaborati. Le eventuali modifiche da attuare richiederebbero infatti pochi secondi, dopo i quali le sezioni sarebbero automaticamente aggiornate senza spreco di ulteriore tempo né risorse.



132 - Linee tracciate in pianta per la creazione delle viste di sezione

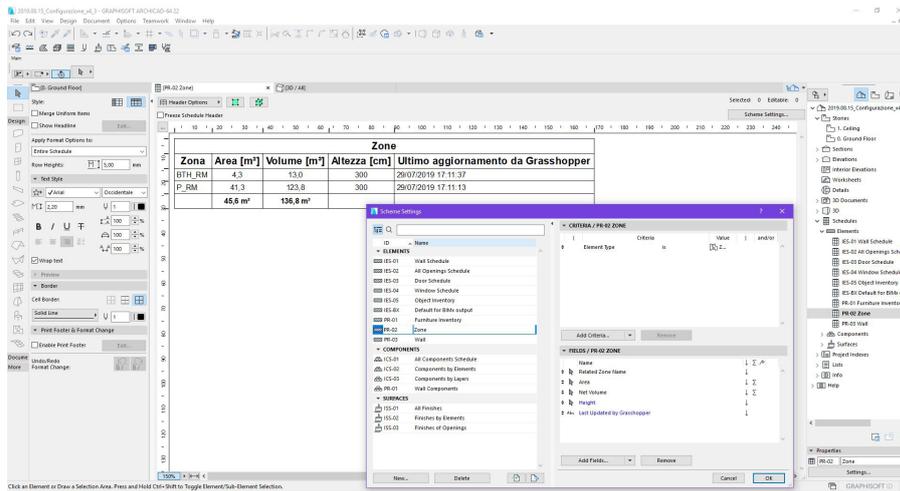
Per offrire una maggiore comprensione del modello potrebbe inoltre essere utile realizzare una **vista 3D** della degenza, grazie alla quale avere una miglior visione d'insieme degli spazi. Come per i prospetti però, in questo caso non sarebbe d'aiuto fornire un'immagine tridimensionale che rappresenti il modello dall'esterno. Può infatti giovare maggiormente uno spaccato assonometrico che, sezionandone due pareti, possa mostrare la disposizione degli arredi generati all'interno. Per fare ciò è possibile impiegare lo strumento che pone dei piani di taglio sul modello, avendo cura, come per le sezioni, di modificarne la posizione all'occorrenza.

133 – Esempio di vista 3D del modello BIM: spaccato assonometrico



Gli **abachi** sono delle tabelle che descrivono dal punto di vista informativo, e non prettamente grafico come per le modalità di rappresentazione appena descritte, determinati oggetti del modello. In *ArchiCAD* essi sono suddivisi in tre tipologie in base al tipo di contenuto che elencano, ovvero se trattano elementi, componenti o superfici. Ognuna di queste categorie comporta diversi criteri per la sua creazione, per questo l'utente deve conoscere a monte quale tipo di abaco e quali informazioni intende restituire.

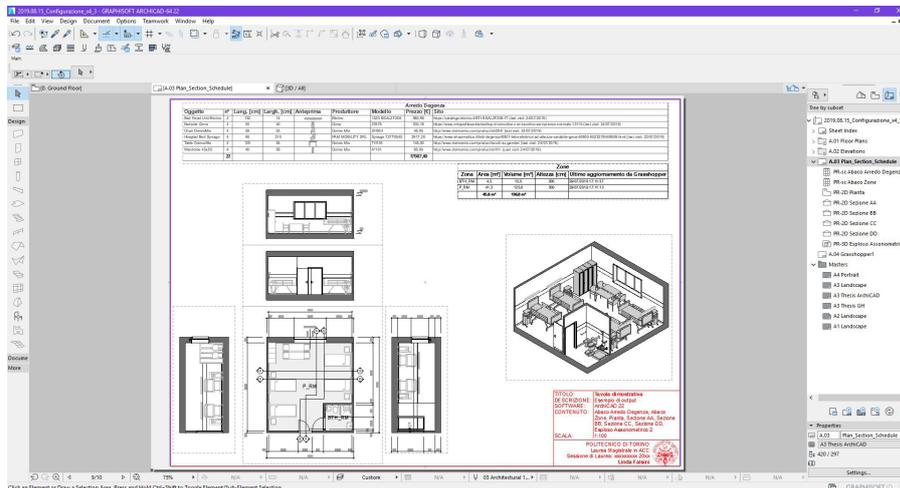
Per il *template* che deve contenere il modello della degenza si ritengono di rilevante importanza due abachi, entrambi atti a descrivere categorie di elementi. Il primo racchiude le informazioni relative alle zone, comprendendo i campi di area, volume altezza e l'ultimo aggiornamento effettuato dall'algoritmo di *Grasshopper*. Quest'ultimo campo è presente solo grazie all'installazione del *plug-in* che consente la *live-connection* col software, e risulta utile per comprendere se l'intero modello è aggiornato. Il secondo abaco riguarda invece gli arredi inseriti nella degenza, la cui motivazione e definizione verrà meglio spiegata nei capitoli successivi.



134 – Abaco delle zone, con dettaglio delle proprietà che ne definiscono criteri e campi

Una volta definite l'insieme delle viste che descrivono il modello, in tutte le sue sfaccettature, è infine opportuno predisporre una **tavola** che le possa contenere. Per fare ciò, dopo aver salvato ogni rappresentazione come vista, applicando le modalità di visualizzazione sopra descritte, è sufficiente trascinarle in un *layout*. Questo può contenere un cartiglio, realizzabile attraverso una pagina *master*, che può contenere sia testo statico, che rimane quindi sempre costante, sia dinamico, che invece varia in base al contenuto del *layout* stesso.

Anche in questo caso, come per alcune rappresentazioni, la disposizione e la dimensione delle finestre contenenti le viste può variare in base a quelle del modello generato. Il *template* è tuttavia realizzato basandosi su un modello di una degenza con quattro posti letto e un bagno annesso. Questo dovrebbe scongiurare l'evenienza di realizzare un modello di maggiori dimensioni partendo dallo stesso algoritmo.



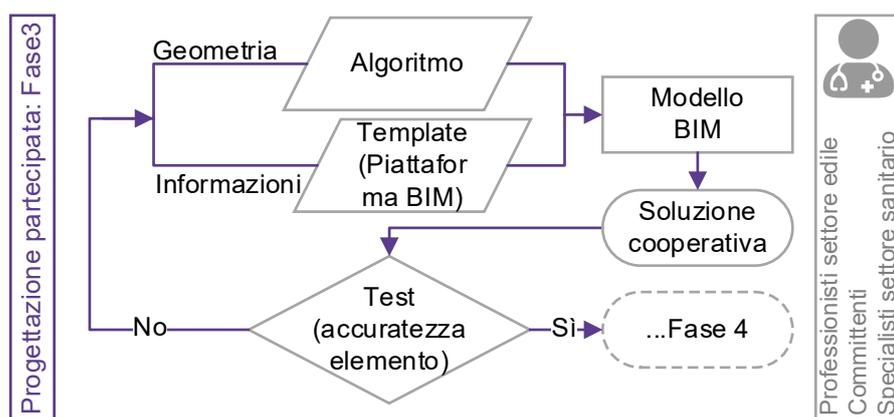
135 – Disposizione delle viste del modello all'interno di un layout di ArchiCAD

2.3 Fase 3: Approccio alla progettazione partecipata

La terza, nonché ultima, fase analizzata nell'ambito della presente ricerca, si colloca all'interno della tematica della **progettazione partecipata**. Questo implica che, come spiegato nei capitoli introduttivi, essa non costituisce in realtà un singolo *step* da "spuntare", ma fa parte dell'intero processo di progettazione. Pertanto anche il tassello principale di questa fase è rappresentato da un test, che, in questo caso, serve a verificare il grado di **accuratezza** del progetto in merito alla soddisfazione degli obiettivi iniziali e delle necessità di tutti gli attori coinvolti, compresi utenti finali, tecnici e professionisti del settore sanitario. Nel caso in cui la degenza, ottenuta mediante la modellazione parametrica illustrata nella fase precedente, non superi tale accertamento a pieni voti, è quindi opportuno provvedere ad effettuare le correzioni necessarie fintantoché tutti gli attori coinvolti non approvino il progetto. Solo a questo punto si può provvedere a passare alla fase successiva, qui solamente ipotizzata, che vede l'inserimento della stanza parametrica all'interno del modello BIM della struttura sanitaria a cui è destinata.

Questa fase eredita, dall'antecedente, l'algoritmo ed il *template*, grazie ai quali possono essere generati, nella piattaforma BIM, rispettivamente le geometrie e le informazioni, il cui dettaglio è già stato precedentemente esplicitato, del modello della degenza. Sono infatti questi due elementi ad essere oggetto di modifiche nel caso in cui il test non abbia esito positivo. Una volta completato il *template* in tutte le sue sfaccettature, il **modello BIM** costituisce di fatto la base dalla quale possono derivare diverse opzioni per ottenere risultati nell'ambito della progettazione partecipata, di cui, in questa ricerca, ne viene approfondita solamente una.

136 - Schema metodologico della fase 3. Per una visione d'insieme vd. diagramma di flusso dell'intera metodologia (fig. 24)



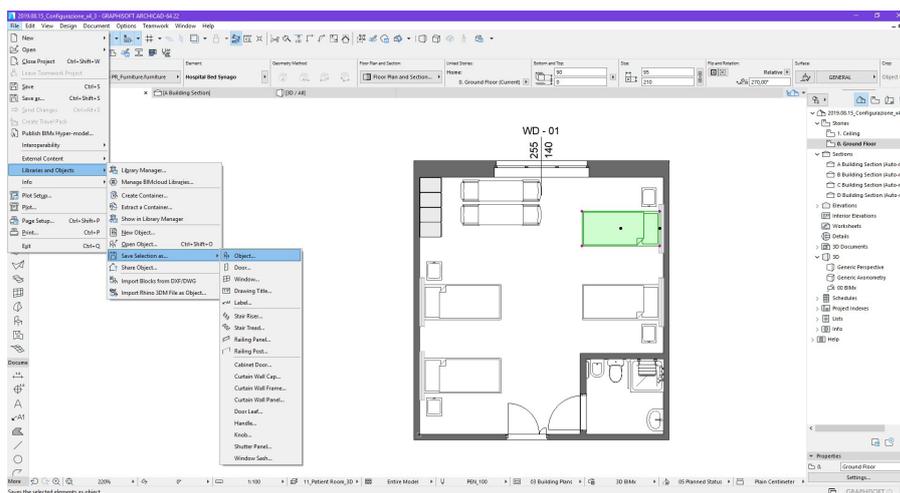
2.3.1 Gestione delle librerie

La preparazione di un *template* prevede sia la definizione di standard grafici, illustrati nella fase precedente, sia la predisposizione di alcuni dati riguardo agli oggetti contenuti nel modello. Essi comprendono sia informazioni riguardanti gli elementi architettonici, come le stratigrafie ed i profili delle murature impiegate, che sono già state analizzate assieme alla generazione, dall'algoritmo alla piattaforma BIM, degli elementi stessi, sia quelle relative agli oggetti d'arredo appartenenti alle librerie. La gestione di queste ultime è di fatto parte integrante dell'allestimento di un *template*, ma in questa tesi di ricerca trova la sua collocazione nella terza fase, in quanto risulta affine ai temi di condivisione e coinvolgimento degli attori all'interno del processo progettuale, oltre che a quelli della modellazione, che caratterizza la fase precedente.

La modalità con cui vengono organizzate le librerie deve essere **finalizzata agli obiettivi** del modello. È infatti consuetudine, in un *team* di progetto, salvare gli oggetti impiegati nel modello in una libreria esterna, cioè in una cartella, condivisa mediante la piattaforma di *teamwork* in dotazione con *ArchiCAD*. In questo modo, qualora un membro del *team* apportasse delle modifiche ad un elemento, gli altri professionisti con cui è condiviso il modello, gioverebbero dell'oggetto aggiornato senza dover cambiare file né ricaricare l'elemento. Nel caso in analisi però l'obiettivo non corrisponde ad una regolare condivisione, ed anche la metodologia adottata è differente. Si ricorda infatti che il modello BIM della degenza, generato grazie all'algoritmo del software di *visual editing* ed il *template*, ha come obiettivo quello di essere inserito all'interno di un modello BIM di una struttura sanitaria principale, comprensivo di differenti altri componenti ed elementi-stanza come quello in questione. È pertanto necessario fare una **distinzione tra gli oggetti** che sono propri della singola funzione della degenza e quelli che invece possono essere distribuiti nell'intero modello ospedaliero. Supponendo che il modello sia generato annettendo il servizio igienico alla degenza, esso contiene, infatti, al suo interno due tipi di oggetti, legati alle diverse tipologie di ambiente. Quelli appartenenti alla degenza possono essere considerati come oggetti "unici", appar-

tenenti alla specifica funzione a cui sono assimilati, mentre i sanitari sono, ipoteticamente, reperibili in altre zone del modello. Si ritiene quindi fondamentale gestire questi oggetti in **librerie differenti e separate**, in quanto una deve contenere arredi il cui collocamento è previsto solo all'interno delle degenze, oggetto di studio, mentre l'altra include oggetti comuni all'intero modello della struttura sanitaria.

La libreria predisposta ad assolvere la funzione di contenitore di oggetti specifici di un singolo modello è quella **interna**, che può quindi essere adottata per tutte le attrezzature sanitarie da collocare nella degenza. Questo tipo di libreria viene condivisa assieme al file del *template* (.tpl), senza quindi dover creare un documento (.pln) di *ArchiCAD*. Di contro, proprio per questa sua caratteristica, è consigliabile mantenere al suo interno solo gli oggetti effettivamente impiegati nel modello poiché, con l'aumentare della dimensione della libreria, si può incorrere in un appesantimento del file .tpl. Un particolare beneficio di questo sistema è rappresentato dalla semplicità con cui si può aggiungere informazioni ad un oggetto. È infatti sufficiente, dopo aver apportato le modifiche desiderate, salvare l'oggetto nella libreria per poterlo poi impiegare nel modello.

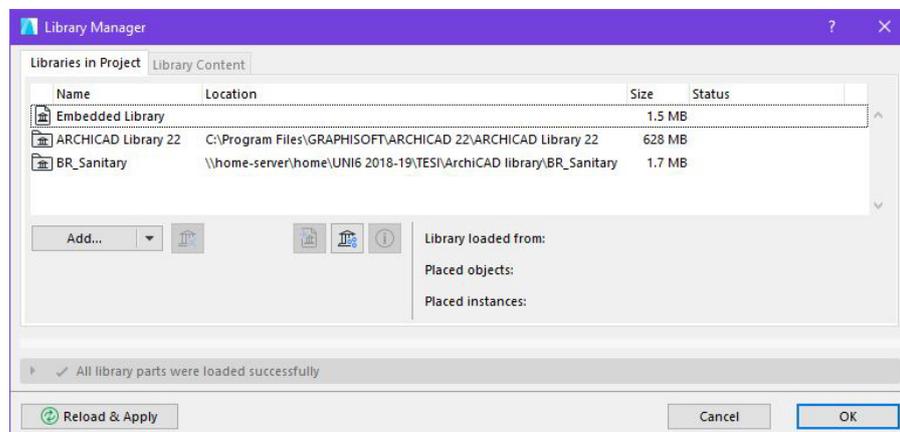


137 – Procedimento per il salvataggio di un oggetto all'interno della libreria interna di ArchiCAD

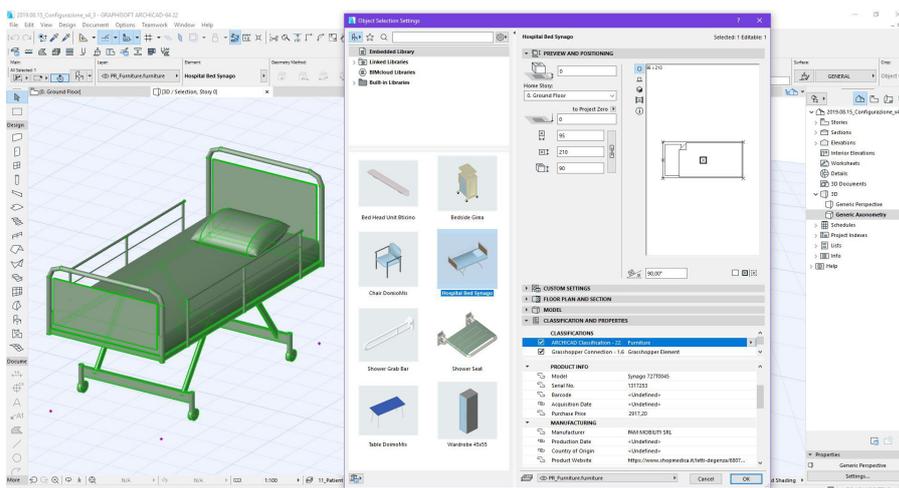
Le attrezzature igienico-sanitarie possono invece essere organizzate all'interno di una **libreria esterna di studio**, comune all'intero ipotetico *team* di progetto che lavora al progetto ospedaliero, come è abitudine nell'ambiente di condivisione dei dati (*Common Data Environment*). Una valida alternativa, che porta a simili risultati, benché comunque diverso e più contorto nella gestione delle modifiche, è l'accorpamento degli oggetti impiegati, all'interno di un file .lef, anch'esso da collegare al modello. Lo svantaggio di tale procedimento è che per poter gestire completamente un oggetto esterno è quasi inevitabile saperne modificare il **codice GDL** che lo contraddistingue. Esportando infatti un oggetto .gsm da *ArchiCAD*, per poterlo collocare nella cartella condivisa, esso perde molte delle proprietà immesse dall'interfaccia delle impostazioni. È perciò necessario che le informazioni, come il prezzo o il peso di un oggetto, debbano essere "scritte" direttamente fra i parametri dello *script* che compone l'elemento. Nel caso in analisi, sia per questa criticità, sia per la non specificità delle attrezzature sanitarie, che si suppongono comuni all'intero macro-modello,

tali oggetti possiedono un livello di dettaglio assimilabile ad un **LOD C**. Essi infatti risultano particolareggiati nella loro geometria, ma privi di informazioni di altro genere, ed è per questa ragione che l'abaco degli elementi, analizzato nei capitoli precedenti, risulta incentrato sui soli oggetti contenuti nella zona della degenza, a cui è associabile invece un **LOD D**.

138 – Gestione delle librerie impiegate all'interno del template per la generazione del modello della degenza

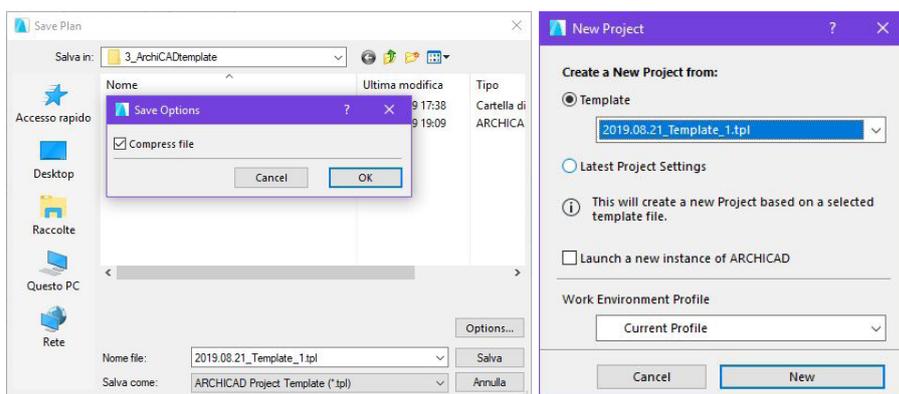


Nell'ottica di una progettazione partecipata efficiente, si ipotizza il coinvolgimento sia di tecnici specializzati, come un **ispettore sanitario** che si occupa della verifica degli standard all'interno della struttura ospedaliera, sia di utenti meno "esperti", come un **committente** che vuole esaminare invece aspetti economici o estetici del progetto. Per soddisfare le esigenze di entrambe le eventuali tipologie di attori è quindi necessario inserire una serie di **informazioni** chiave all'interno delle attrezzature sanitarie impiegate nel modello. È pertanto fondamentale la ricerca di analoghi oggetti sul mercato che trasformino un comune componente tridimensionale nella **rappresentazione di un oggetto reale**. Questo permette di fornire a tutti gli utenti che navigano il modello un'idea molto più chiara e veritiera di ciò che visualizzano. Si ritiene tuttavia superflua la modifica della geometria dell'oggetto modellato, che viene quindi mantenuta generica, in quanto la definizione dell'attrezzatura sanitaria è spesso oggetto di frequenti rivisitazioni, che comporterebbero numerose ore di lavoro per ogni modifica. Tale decisione costituisce la motivazione di un livello di dettaglio che non raggiunge l'oggetto specifico (LOD E), ma che si ferma a quello di un **oggetto dettagliato** (LOD D). È infatti sufficiente, anche per un utente non specializzato, prendere visione del sito a cui l'oggetto 3D fa riferimento, grazie all'inserimento del *link*, che porta alla pagina internet del suo produttore, fra le informazioni dell'elemento stesso. Le proprietà di un oggetto di ArchiCAD consentono inoltre di inserire molti dati legati al prodotto, alla sua fabbricazione ed alla sua manutenzione nell'arco del suo intero ciclo di vita. In questa ricerca questi aspetti non vengono presi in considerazione, nonostante la loro potenzialità, e quindi, per ogni attrezzatura sanitaria, sono inseriti modello, produttore, prezzo e *link* alla scheda tecnica.



139 – Schermata delle proprietà informative relative ad un oggetto

Con l'aggiunta delle informazioni agli oggetti e l'organizzazione di questi ultimi all'interno delle librerie, si chiude l'elenco di impostazioni che compone il *template* del modello della degenza. A questo punto è possibile svuotare il file dagli elementi generati attraverso l'algoritmo del software di *visual editing* e salvarlo come formato *.tpl*, per poterlo impiegare come *template* per ogni generazione della stessa tipologia. Aprendo un nuovo documento di *ArchiCAD* e selezionando il file *.tpl* creato, si potrà quindi accedere a tutti gli standard analizzati ed impostati.



140 – Salvataggio del file come template (a sinistra) e selezione del *.tpl* per la creazione di un nuovo documento di modellazione (a destra)

2.3.2 Requisiti per la progettazione partecipata

Nel capitolo precedente sono state analizzate le informazioni utili per la condivisione del modello ed il coinvolgimento di utenti esterni all'interno del processo progettuale. Si procede a questo punto con la ricerca della modalità con cui mettere in atto la vera e propria progettazione partecipata.

Come analizzato nei capitoli introduttivi, uno dei metodi più efficienti, e di più largo impiego, per il coinvolgimento intersettoriale in ambito edilizio, prevede l'adozione di tecnologie attinenti alla realtà virtuale, aumentata o mista. In particolare, *Graphisoft* prospetta sei differenti applicativi con cui la piattaforma di *ArchiCAD* può interagire specificamente per la **realtà virtuale**, ognuno dei quali ha proprie peculiarità, specifiche tecniche ed opzioni di interoperabilità. Si

Nel caso del modello della degenza in analisi, si propongono i seguenti **requisiti** in merito all'applicativo da selezionare per il soddisfacimento della progettazione partecipata:

- La **navigabilità** del modello tridimensionale, delle tavole e degli abachi, per avere una visione d'insieme delle zone progettate;
- La possibilità di visualizzare le **informazioni** attribuite agli oggetti, in modo da poter prendere visione delle rispettive schede tecniche ed interrogare il modello con le sue proprietà;
- La **misurabilità** delle distanze, con l'obiettivo di verificare che i vincoli dimensionali imposti sul software di *visual editing* corrispondano a quelli della normativa;
- La possibilità di vivere un'**esperienza immersiva** all'interno del modello, per avere una percezione maggiormente qualitativa della distribuzione degli oggetti nello spazio. Questo requisito differisce dalla sola navigazione per il grado di impatto emotivo sull'utilizzatore, che nel primo caso è ridotto al minimo, mentre nel secondo è massimizzato grazie anche all'impiego di appositi visori per la realtà virtuale immersiva;
- L'eventualità di poter beneficiare di **aggiornamenti automatici** tra la piattaforma di *ArchiCAD* e l'applicativo per la progettazione partecipata;
- La possibilità di segnare errori ed **annotazioni** direttamente tramite l'applicativo per avere un dialogo bidirezionale fra i progettisti e gli altri utenti con cui è condiviso il modello;
- L'**accessibilità** alla navigazione del modello anche in lontananza rispetto allo studio, ad esempio in cantiere, o anche la possibilità per un utente di prendere visione del progetto senza dover prendere un appuntamento presso lo stesso luogo fisico in cui lavorano i progettisti. In questo modo si rende implicita la volontà che tale applicativo possa essere usufruito senza costi aggiuntivi, o comunque riducendo al minimo le spese, da parte degli attori coinvolti, che non devono così dotarsi di onerosi dispositivi o software. Inoltre questo requisito, assieme al precedente, ben si applica a progetti di grande entità che coinvolgono utenti in più parti del mondo, come è facile che avvenga per una struttura sanitaria di grande entità.

Premettendo che non sono eseguiti test sui programmi analizzati, ma che viene impiegata la tabella sopra riportata, oltre agli specifici siti di ogni applicativo, come fonte per confrontare i vari software fra loro, è possibile verificare se ognuno di essi rispetti i requisiti richiesti o se vada invece scartato dallo studio.

La navigabilità del modello è solitamente una condizione imprescindibile dai software di questo genere, benché spesso difficilmente è possibile visualizzare anche tavole o abachi, nemmeno in modalità *desktop*. Tuttavia, è comunque una caratteristica che non sempre prevede l'indipendenza dal programma VR da cui deriva, che, in tal caso,

deve quindi essere installato sul computer per poter navigare il modello, contestando quindi l'accessibilità a tutti gli attori coinvolti. Questo esclude l'adozione di **VRcollab** e di **Lumion**, che non prevedono la creazione di file "portable standalone" (vedi fig. 141).

La possibilità di visualizzare le informazioni attribuite agli oggetti è un requisito difficile da assicurare, in quanto è uno dei motivi principali per cui si fa largo impiego di motori grafici intermediari e di appositi *script* per poter visualizzare i dati immessi attraverso la piattaforma BIM. Questo requisito non è infatti rispettato da **Twinmotion**, **Enscape** e **Fuzor**, che sono perlopiù improntati sulla realtà virtuale immersiva in tutti i loro aspetti grafici di *rendering* al fine di rappresentare un modello che sia il più vicino possibile alla realtà. Come già anticipato, invece, il tipo di rappresentazione in questa ricerca non è volta a questo obiettivo, bensì a quello di restituire un **modello concettuale** che possa fornire graficamente solo un'indicazione generica dell'oggetto ed il suo reale ingombro nello spazio.

Già dopo queste poche considerazioni, rimane **BIMx** come unico applicativo che possiede le caratteristiche poste come requisiti fondamentali. Oltre ad essere dotato di tutte le peculiarità per cui gli altri cinque software sono stati esclusi, è infatti in grado di soddisfare anche gli altri punti in elenco. Bisogna tuttavia sottolineare che alcune funzioni, come quella delle misurazioni, sono disponibili solo nella versione **BIMx PRO**, che è a pagamento, ma è comunque fruibile sul modello se a condividerlo è un professionista che possiede una regolare licenza di *ArchiCAD*. Altre operazioni invece, come le annotazioni e l'aggiornamento in *Live Connection* del modello, sono fruibili solo mediante il caricamento del file sulla piattaforma di condivisione, che in questa ricerca non è oggetto di analisi. Pertanto queste potenzialità, benché verificate l'esistenza, anche attraverso ricerche e tutorial online, non sono qui testate nella loro efficienza.

La scelta del software per approcciarsi alla progettazione partecipata ricade perciò su **BIMx**, in quanto rispecchia tutti i requisiti posti e offre anche ulteriori benefici, che verranno meglio approfonditi nel prossimo capitolo.

2.3.3 Progettazione partecipata: **BIMx**

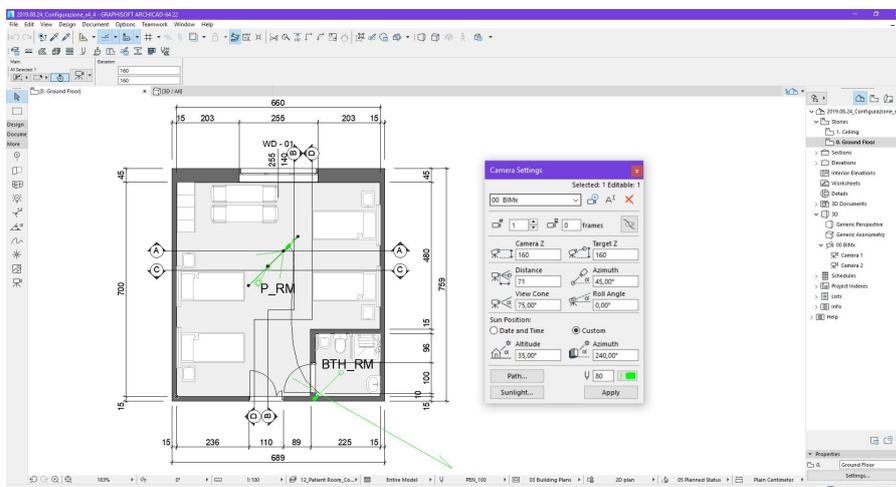
BIMx è un'applicazione che si basa su un "iper-modello", ovvero su una tecnologia sviluppata da *Graphisoft*, che consente di integrare la navigazione del progetto dell'edificio sia in 2D che in 3D [81]. Questo implica la possibilità di consultare al contempo tavole, documenti e abachi 2D, ma anche il modello 3D, sia in modalità immersiva che non, potendo, nel secondo caso, interrogare gli oggetti per conoscerne le informazioni che lo caratterizzano. Per usufruire di ciò, per un utente, è sufficiente possedere un **dispositivo mobile**, come un *tablet* o uno *smartphone*, per l'accesso, in ogni luogo, ad un'interfaccia intuitiva che non necessita di padroneggiare particolari competenze per il suo utilizzo. È inoltre possibile godere della modalità immersiva gra-

zie all'impiego di un **Google Cardboard 2.0**, un visore economico e semplice, costruibile anche autonomamente seguendo determinate specifiche, costituito sostanzialmente da un cartoncino piegato, dentro al quale posizionare lo *smartphone* per visualizzare il modello attraverso le lenti. È comunque possibile la navigazione VR senza visore, in modalità *desktop*, quindi non immersiva, attraverso la versione di *BIMx* per *Windows*, scaricabile dal sito di *Graphisoft*.



142 – Google Cardboard [82]

Grazie alla gestione dei parametri della degenza, attraverso il software di *visual editing*, e al *template*, precedentemente creato sulla piattaforma BIM, è possibile dar vita ad un modello su *ArchiCAD* che riproduca gli aspetti progettuali sia dal punto di vista geometrico che informativo, sia grafico. È possibile aggiungere a tale modello delle “telecamere” per poter in seguito individuare su *BIMx* delle viste selezionabili tra cui scegliere per partire con la navigazione immersiva del modello. Nel caso di una degenza con servizio sanitario annesso è quindi necessario posizionarne almeno due, una per zona, altrimenti può esserne sufficiente una sola nella degenza. Non vi è pertanto un limite massimo di telecamere posizionabili ed un progettista è libero di creare tutte le viste che ritiene necessarie per la comunicazione del modello con gli altri attori coinvolti. Con la versione *PRO* di *BIMx* questa operazione può essere fatta anche direttamente dall'applicazione, ma ciò non permette l'accesso a tutte le proprietà di posizionamento della telecamera disponibili invece su *ArchiCAD*.



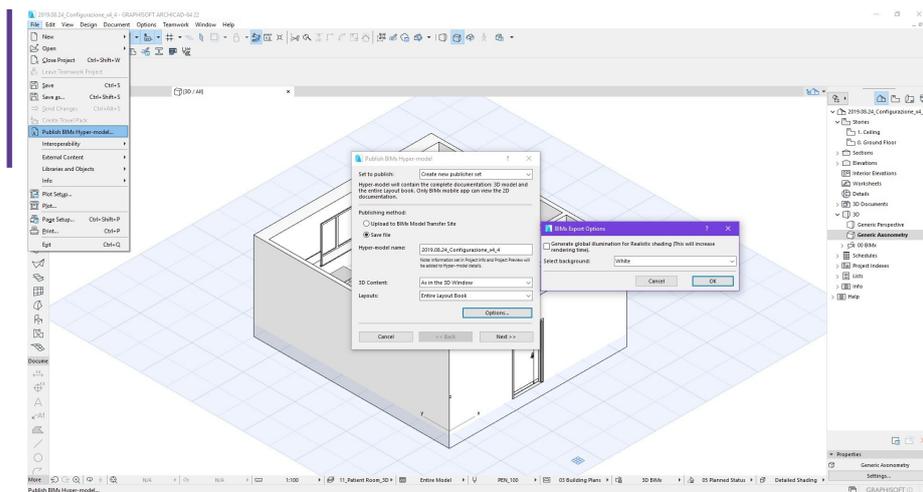
143 – Posizionamento telecamera su ArchiCAD e proprietà dello strumento per la gestione delle viste su BIMx

Per la realizzazione di un modello somigliante alla realtà, prima di creare l'iper-modello, bisognerebbe ancora impostare accuratamente luci e texture degli elementi modellati. Nella presente ricerca si ri-

corda tuttavia che lo scopo del progetto è quello di rappresentare un **modello concettuale**, che descriva l'ingombro degli oggetti e le loro rispettive informazioni, pertanto questi due aspetti non sono ritenuti oggetto di studio.

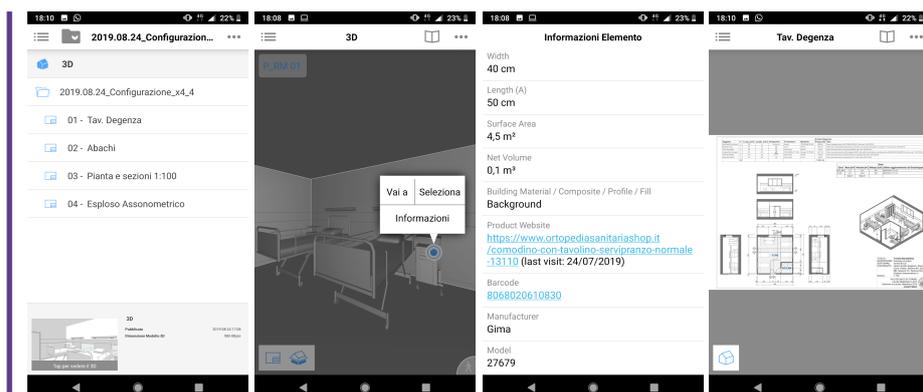
La creazione di un iper-modello non necessita dell'installazione di software o *plug-in* aggiuntivi, in quanto esso viene "esportato" come **output diretto** ed integrato da *ArchiCAD*. Si può scegliere se salvare il file come modello *.bimx* o caricarlo sulla piattaforma di *teamwork* online. Nel secondo caso, che non è però oggetto d'analisi approfondita, si ha la possibilità di avere accesso a due funzionalità aggiuntive, rispetto al salvataggio locale del file, che consentono un più ampio approccio alla progettazione partecipata. Esse derivano dalla forza della condivisione, grazie alla quale è possibile mantenere un **dialogo attivo** tra gli attori coinvolti, che possono, da un lato avere la certezza di navigare un modello sempre aggiornato (a patto che il progettista lo carichi dalla piattaforma BIM), dall'altro la possibilità di trasmettere i loro commenti ed annotazioni. Questa potenzialità permette di creare uno storico di tutte le segnalazioni inviate dai vari utenti, che dispongono quindi dei mezzi per lasciare note o promemoria, ed allegare eventuali *screenshot* riguardo alla porzione di modello a cui fanno riferimento nei messaggi.

144 – Creazione del modello BIMx (.bimx) da ArchiCAD per la progettazione partecipata

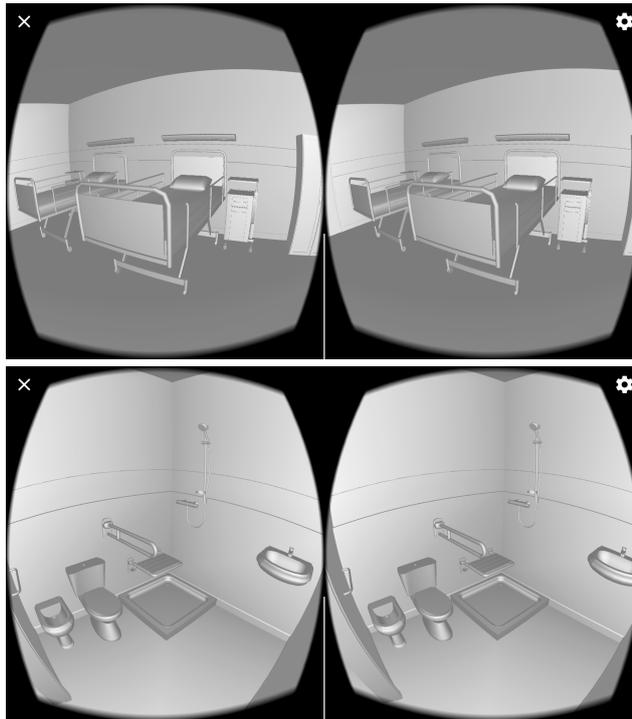


Tramite l'**applicazione** per *smartphone* (qui impiegata nella sua versione per *Android*), è possibile, dopo aver caricato l'iper-modello, navigare tra i vari documenti ed interrogare il modello con il semplice utilizzo del *touchscreen*.

145 – Utilizzo dell'app BIMx, in modalità navigazione. Da sinistra a destra: a) interfaccia di navigazione tra documenti; b) interrogazione di un oggetto del modello; c) informazioni restituite dall'oggetto interrogato; d) visualizzazione di un documento.



Dalle impostazioni è inoltre facile passare dalla semplice navigazione del modello, ad una realtà virtuale più **immersiva**, che, dopo essere stata selezionata, necessita l'utilizzo del *Google Cardboard*.



146 - Utilizzo dell'applicazione *BIMx*, in modalità realtà virtuale immersiva. Attraverso *Google Cardboard* ogni vista appare come un'unica immagine a 360°, mentre gli screenshot sdoppiano la visualizzazione perdendo anche l'idea

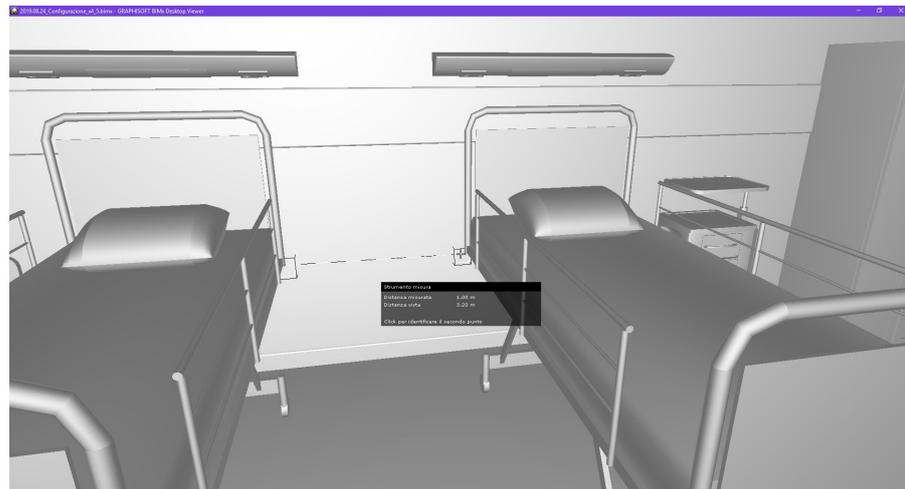
La versione di **BIMx Desktop Viewer** consente di usufruire delle stesse funzionalità della modalità in navigazione dell'applicazione, con l'accortezza di impiegare comandi da tastiera, che sono però resi ben noti nell'interfaccia del software.



147 - Utilizzo dell'applicazione *BIMx* (versione per Windows), in modalità desktop: elenco comandi da tastiera

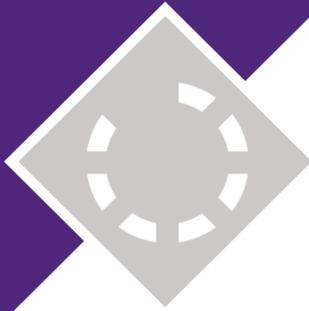
Tramite la modalità *desktop*, come per l'applicazione, è infatti possibile navigare liberamente, spostarsi da una vista all'altra ed interrogare il modello. Non è però possibile integrare un visore a questo software, quando usato da computer, lasciando così da parte l'immersione in realtà virtuale.

148 - Utilizzo dell'applicazione BIMx (versione per Windows), in modalità desktop. Esempio di interrogazione del modello: misurazione della distanza tra due attrezzature sanitarie, oggetto a vincoli da normativa.



L'approccio alla progettazione partecipata con *BIMx* permette agli attori coinvolti all'interno del processo edilizio di fruire di una semplice, ma efficace, rappresentazione del modello, che consenta, nel caso in analisi, di comprenderne gli **aspetti tecnici** principali, senza necessità di costose ed ingombranti attrezzature. Progettisti ed utenti, siano essi professionisti di altri settori o committenti, possono quindi così avere una base tramite cui sostenere un **dialogo bidirezionale**, non lasciando largo spazio alle incomprensioni. È inoltre possibile che tale comunicazione avvenga in separate sedi fra gli attori, o anche direttamente durante i sopralluoghi, grazie all'**accessibilità** che caratterizza tale metodologia.

Nel capitolo "5 Allegati digitali" è possibile scansionare un QR code (n. 4) per prendere visione di un video dimostrativo delle funzionalità di *BIMx* analizzate.



3

Risultati

3 Risultati

I risultati della presente ricerca sono da intendersi sia come sintesi delle criticità sia come riflessioni di ciò che è stato prodotto attraverso la metodologia esaminata nei capitoli precedenti. Pertanto, per un migliore approfondimento delle tematiche trattate, essi sono suddivisi nuovamente nelle fasi che caratterizzano il lavoro svolto. Prima di entrare nel merito, però, si riassumono, nella tabella sottostante, i motivi che hanno spinto all'elaborazione della metodologia sviluppata, come supporto per la piena comprensione dei risultati ottenuti, di seguito analizzati.

Obiettivo da raggiungere (What?)	Testare un nuovo approccio metodologico nell'ambito delle strutture sanitarie, che, in quanto pubbliche, dal 2019 in Italia hanno l'obbligo di essere progettate adottando il BIM, se di importo pari o superiore a 100 milioni di euro [2].
Come raggiungerlo (How?)	Sfruttare la ripetitività delle funzioni attinenti ai complessi sanitari e ragionare quindi su micro-modelli monofunzionali , adattabili parametricamente sia per geometria che informazioni, da inserire poi in un macro-modello ospedaliero.
Chi viene coinvolto (Who?)	Coinvolgere, sin dalle fasi iniziali del progetto, molteplici attori coinvolti nel processo, non solo del settore edilizio, ma anche sanitario, e perfino i committenti dell'opera, applicando le teorie della progettazione partecipata .
Motivo della ricerca? (Why?)	Avere un maggiore controllo delle singole funzioni progettate, potendo così ridurre tempi, costi e risorse dei progetti sanitari, anche grazie al dialogo costante con gli attori per la diminuzione di fraintendimenti ed errori.

3.1 Fase 1: La ricerca dell'approccio metodologico

La prima fase della ricerca è stata di fondamentale importanza al fine di definire quali decisioni prendere, e quali scartare, per l'effettivo raggiungimento degli obiettivi posti. Per fare ciò si è dimostrato necessario non solo condurre delle **ricerche** per approfondire la già presente letteratura sul tema, ma anche **testare** l'effettiva veridicità delle informazioni così ricavate.

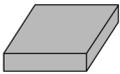
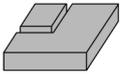
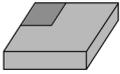
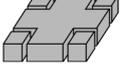
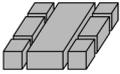
In primo luogo si è ipotizzato di impiegare direttamente due piattaforme BIM, selezionate fra quelle più diffuse statisticamente, per la creazione di un **modello parametrico** della camera di degenza che potesse essere **compatibile** con più software. *Revit* e *ArchiCAD* rappresentano quindi due campioni, scelti all'interno di una vasta popolazione, che servono per generalizzare, a livello teorico, i risultati ottenuti tramite i test effettuati, anche alle altre piattaforme BIM che non sono state prese in considerazione.

L'idea di parametrizzare direttamente attraverso una piattaforma BIM, che è intrinsecamente contraddistinto da parametri, era fondata sulla nozione che l'**IFC** rappresenti il formato di interscambio più utilizzato per l'ottimizzazione dell'interoperabilità tra piattaforme BIM, senza perdite né distorsioni di dati. Pertanto, si è ipotizzato che, modellando e parametrizzando la degenza su una delle due piattaforme, si potesse poi esportare e reimportare il modello all'interno della seconda. A questo punto si prevedeva quindi di testare entrambe le modalità di creazione dei parametri dei due software, per poter, in seguito, definire la presenza o meno di un flusso di interoperabilità preferenziale.

Le riflessioni su come realizzare un oggetto parametrico, che non fosse un elemento contenuto in un ambiente, come di consuetudine, bensì un'intera stanza, sono partite da *Revit*, piattaforma la cui interfaccia di creazione dei parametri è più **user-friendly**. La possibilità di utilizzare *ArchiCAD* in questo primo test è stata infatti momentaneamente esclusa, in quanto il primo ostacolo incontrato riguarda la modalità di parametrizzazione di questa piattaforma, che prevede la conoscenza dell'utilizzo del **codice GDL**.

Nella tabella sottostante sono riportate tutte le varianti elaborate, con le relative criticità riscontrate, che hanno portato infine alla defi-

nizione di due famiglie parametriche. Le ultime due soluzioni infatti, rispettivamente **ipotesi 6 e 7**, erano da considerarsi entrambe potenzialmente valide, in base al tipo di geometria del servizio igienico annesso alla camera di degenza. Nel primo caso infatti si prevede che il suo volume sia allineato con quello della stanza, mentre nel secondo fuoriuscente.

	Descrizione	Criticità
Ipotesi 1 	Modellazione e definizione dei parametri direttamente dal file di progetto , associando piani di riferimento agli elementi muro .	Possibilità di associare solo parametri globali e non di istanza, rendendone l'editabilità non immediata.
Ipotesi 2 	Modellazione e definizione dei parametri direttamente dal file di progetto , associando piani di riferimento ad una massa concettuale .	Necessità di aggiornare manualmente i muri associati alla massa ad ogni modifica dei valori dei parametri.
Ipotesi 3 	Modellazione di due masse concettuali "ad incastro" allo scopo di adibire la maggiore alla funzione di degenza , e la minore a servizio igienico .	Poca flessibilità degli elementi e difficoltà nella gestione dei parametri. Linkando il prototipo ad un modello si dovrebbero realizzare tanti file quante differenze sono presenti fra le singole degenze.
Ipotesi 4 	Modellazione di due masse concettuali inserite una dentro l'altra in sovrapposizione, all'interno della stessa famiglia .	Limitazioni a livello geometrico , dovute ai vincoli inseriti per rendere l'oggetto parametrico.
Ipotesi 5 	Modellazione di una massa concettuale "a croce" nei cui cavedi perimetrali inserire altre quattro masse da adibire, una in alternativa all'altra, allo scopo di servizio igienico.	La modifica dei parametri comporta la perdita dei vincoli della massa cruciforme principale.
Ipotesi 6 	Scomposizione della massa principale in ulteriori tre masse, definendo un totale di sette masse all'interno della famiglia.	Le possibili variabili geometriche prevedono che i volumi dei servizi siano sempre allineati a quello del corpo centrale.
Ipotesi 7 	Definizione di un maggior numero di parametri per rendere le masse adibite ai servizi sporgenti rispetto al corpo centrale.	Scarsa immediatezza nella gestione dei parametri e difficoltà nel calcolo dell'area dei locali e del numero di posti letto.

Una volta definito un prototipo con un sufficiente numero di tipologie di **parametri** differenti, quali **dinamici, dimensionali e di testo**, si è potuto procedere con i test di interoperabilità per verificare l'efficacia dell'importazione dei dati su *ArchiCAD* della famiglia così definita in *Revit*. Per effettuare l'esportazione è stato impiegato un file di progetto (.rvt) in cui sono state inserite la famiglia derivata dall'ipotesi 6 e altri elementi di differente natura, al fine di ottenere più riscontri possibili, comprendenti anche gli aspetti di **editabilità di componenti e geometrie**, partendo da un'unica analisi. Sono in realtà stati eseguiti differenti test sfruttando non solo l'esportazio-

ne e l'importazione diretta tra le due piattaforme, impiegando diverse impostazioni e versioni di IFC, ma anche due *plug-in* pensati appositamente per questo tipo di interoperabilità. In ognuna delle verifiche eseguite, tuttavia, il risultato non è stato all'altezza delle aspettative, dimostrando notevoli **perdite di dati**, specialmente riguardo ai parametri dinamici, aspetto invece fondamentale per il progredire della ricerca.

Nella tabella sottostante si riassumono i risultati dei test svolti, aggiungendo, oltre agli scambi di dati da *Revit* ad *ArchiCAD*, anche gli analoghi effettuati nel flusso direzionale opposto. Si è voluto di fatto verificare se, nonostante le difficoltà che caratterizzano la parametrizzazione all'interno della piattaforma *Graphisoft*, sarebbe valsa la pena lo sforzo di apprendimento del linguaggio GDL per avvalersi dell'esportazione della degenza, stavolta da *ArchiCAD*, a *Revit*. I test hanno, però, avuto il medesimo **esito negativo**, persino reimportando il modello nel software stesso in cui è stato originato, dimostrando come il formato di interscambio IFC debba ancora evolversi per ottenere un'interoperabilità tra piattaforme che salvaguardi l'editabilità di oggetti e l'esportabilità di parametri, malgrado le peculiarità che contraddistinguono inevitabilmente i differenti software.

Flusso interoperabilità	Versione IFC	Formato file	Importazione geometria	Editabilità geometria
Revit → ArchiCAD	2x3 coordination view	.ifc /.ifczip	✓	✓
	4 reference view		✗	-
	2x3 coordination view	.ifcxml	✓	✓
Revit → Revit	2x3 coordination view	.ifc /.ifczip	✓	✓
ArchiCAD → Revit	general export	.ifc /.ifczip	✓	✓
	revit export for reference model		✓	✗
	revit mep export		✓	✗
	revit structure export		✓	✓
	general export	.ifcxml	✗	-
	revit export for reference model		✗	-
	revit mep export		✗	-
revit structure export	✗	-		
ArchiCAD → ArchiCAD	general export	.ifc /.ifczip	✓	✓

149 – Tabella riassuntiva dei test d'interoperabilità svolti tra *Revit* ed *ArchiCAD*

Importazione arredo	Editabilità arredo	Importazione parametri di testo	Editabilità parametri di testo	Importazione parametri dinamici	Mantenimento host
✓	✗	✗	-	✗	✗
✓	✗	✗	-	✗	✗
✓	✗	✗	-	✗	✗
✓	✗	✓	✓	✗	✗
✓	✗	✓	✓	✗	✓
✓	✗	✗	-	✗	✗
✓	✗	✗	-	✗	✗
✗	-	✓	✓	✗	✓
✗	-	✗	-	✗	✗
✗	-	✗	-	✗	✗
✗	-	✗	-	✗	✗
✗	-	✗	-	✗	✗
✓	✗	✗	-	✗	✗

A questo punto della ricerca è stato indispensabile rivisitare il tipo di approccio metodologico da impiegare per proseguire con lo sviluppo dell'analisi metodologica. Escludendo la possibilità di confidare nell'interoperabilità diretta tra le piattaforme BIM, bisognava quindi cercare dei software che facessero da “intermediari” tra la gestione dei parametri e la loro corretta esportazione ed importazione. Dopo alcune indagini, la scelta è ricaduta su **Grasshopper**, grazie alla presenza di numerosi *plug-in* che consentono la connessione ed il trasferimento di dati di natura geometrico-parametrica, e non statiche importazioni, all'interno delle piattaforme BIM. Le fonti di tali applicativi aggiuntivi sono tuttavia di duplice natura, e da questo dipende la loro reperibilità ed affidabilità. Il primo tipo è sviluppato da note *software house*, come per il *plug-in* nato dalla collaborazione tra *Graphisoft* e *McNeel & Associates*, mentre il secondo è solitamente ideato da utenti privati, come quelli di *WhiteFeet Software* e *Moethu*, che li rendono disponibili online. Questa seconda tipologia è di fatto la **meno stabile**, in quanto sono progetti che dipendono da singole persone, o piccoli studi, che non hanno come obiettivo principale lo sviluppo di software, ma che se ne occupano solo nel tempo che rimane loro disponibile dalle principali mansioni svolte. Per questo motivo, molte delle ricerche effettuate riguardo ai potenziali *plug-in* impiegabili, hanno portato a scarsi risultati, proprio perché spesso, negli articoli, vengono utilizzati alcuni di questi applicativi che però nel tempo non sono più stati aggiornati rispetto alle nuove versioni dei programmi con cui devono comunicare. Anche se alcuni risultano ancora scaricabili, sono perciò inutilizzabili se non con una versione di *Revit* o *ArchiCAD* ormai obsoleta, visti i continui progressi tecnologici che assecondano la crescente tematica del BIM. Un'altra criticità rappresenta la possibilità di avere supporto e servizi di assistenza, che risulta notevolmente più elevata per i *plug-in* delle *software house*, in confronto a quelli meno ufficiali provenienti da “utenti comuni”. In più, le funzionalità di questa tipologia di *plug-in* risultano spesso limitate, rispetto a quelle sviluppate dalle *software house*, obbligando l'utente ad utilizzarne più di uno per ottenere un risultato efficiente. È stato infatti proprio questo il caso sperimentato, in quanto l'utilizzo di *Hummingbird* o *Grevit* singolarmente non ha portato ad una soluzione completa per *Revit*, contrariamente all'impiego della *Live Connection* di *ArchiCAD*.

I differenti *plug-in* selezionati sono stati testati per verificarne, non solo l'efficacia del trasferimento dei dati dal software di *visual editing* alla piattaforma BIM, ma anche la sua efficienza. Ciò che ne è derivato, riassunto nella tabella sottostante, è stato un risultato soddisfacente per quanto riguarda i test applicati ad *ArchiCAD*, con il quale è possibile instaurare una **connessione in tempo reale** rispetto alle modifiche dei parametri effettuate tramite *Grasshopper*. Viceversa, su *Revit*, il successo di questa metodologia si è dimostrato più limitato. La “natura ufficiosa” dei *plug-in* impiegati, ha portato a simili risultati, ma con delle tempistiche ed un impiego di risorse maggiori. A dimostrazione di questo sono i numerosi **svantaggi** riscontrati, tra cui:

- l'impossibilità di verificare se il modello BIM sia aggiornato alle ultime modifiche, in quanto esse non avvengono in tempo reale, come invece accade per *ArchiCAD*;
- la mancanza di componenti che attingano direttamente all'elenco di famiglie presenti su *Revit*, il che obbliga l'utente a scriverne manualmente il nome, innalzando la possibilità di incorrere in errori e rischiando nella mancata lettura degli oggetti, in caso la piattaforma abbia una lingua differente rispetto a quella impiegata per la denominazione dell'elemento;
- nel caso di *Hummingbird*, la generazione di numerosi file CSV, uno per ogni componente creato, rischia di compromettere un'ordinata disposizione dei file di progetto aggiornati, caratteristica fondamentale per evitare confusione ed imprecisioni in uno studio;
- nel caso di *Grevit*, l'accortezza di dover scalare l'unità di misura del modello in una metrica che rientri nel sistema internazionale (SI), essendo il *plug-in* creato basandosi sul piede.

Queste complicazioni hanno comportato gravi mancanze e criticità, che hanno però permesso di comprendere che l'approccio metodologico scelto può comunque essere valido per entrambe le piattaforme, in quanto, rispetto al primo analizzato, ha portato quantomeno a dei **risultati efficienti**. Il problema consiste tuttavia nel maggiore **impiego di risorse** richieste dal metodo, per il quale, in fase di ricerca se ne vogliono comunque approfondire le potenzialità, ma, in ambito reale, non sarebbe da escludere un secco diniego per il presumibile dispendio di tempo che ne comporterebbe.

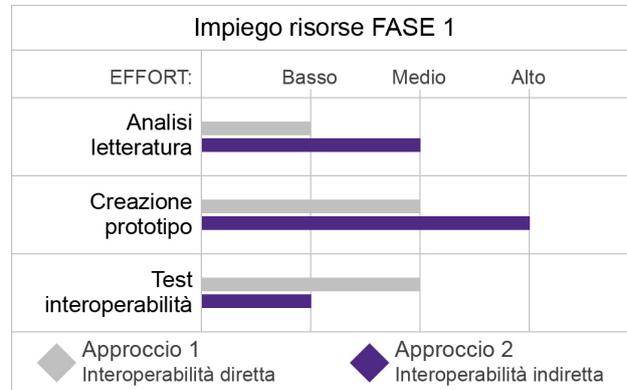
Plug-in	Piattaforma BIM	Muro	Solaio	Zona	Serramenti	Arredo
Grasshopper - ArchiCAD Live Connection	ArchiCAD	✓	✓	✓	✓	✓
Hummingbird	Revit	✓	✓	✓	✗	✓
Grevit		✓	✓	✗	✓	✓

150 – Esito dei test effettuati sui componenti analizzati nel trasferimento dei parametri da Grasshopper alle piattaforme BIM

In conclusione, la prima fase della ricerca non ha subito portato ai risultati sperati, tuttavia, con un secondo tentativo è stata intrapresa una nuova strada, che è poi di fatto stata approfondita nelle fasi successive. Questo nuovo approccio ha dimostrato di avere grandi **potenzialità**, ma ha anche dato prova di essere molto **più impegnativo** rispetto al primo. Nell'immagine sottostante sono state riassunte le componenti principali che hanno caratterizzato l'intera fase, confrontando l'impiego di risorse richiesto per ciascun approccio. Come anticipato, la ricerca teorica è stata più esaustiva e di facile reperimento nel primo caso, benché le sue tesi siano presto state confutate, mentre, al contrario, l'avanzamento tecnologico dei software ha contribuito a rendere obsoleti molti degli studi e degli articoli accademici più recenti, riguardanti il secondo approccio. Allo stesso modo, la creazione del modello semplificato, a parità di conoscenze dei programmi impiegati, è stata più complessa e prolissa nell'ultima metodologia analizzata. Per quanto riguarda, invece, i test svolti, con l'interoperabilità diretta

fra le piattaforme BIM si è trattato di un processo mediamente lungo, in cui il numero di operazioni da effettuare era compreso in un ampio ventaglio di opzioni. Al contrario, la verifica del trasferimento dei parametri geometrici dallo *script* visuale alle piattaforme BIM, essendo privo di parametri complessi, è consistito in piccole e semplici operazioni volte al controllo della corretta generazione delle geometrie.

151 – Panoramica delle risorse impiegate per le sezioni che costituiscono la prima fase della ricerca: confronto tra i due approcci testati



3.2 Fase 2: Il paradosso dell'approccio parametrico

La seconda fase della ricerca è quella che ha richiesto la maggior quantità di tempo all'interno dell'intera tesi. Il suo sviluppo è stato incentrato sulla creazione dello *script* che costituisce l'insieme delle operazioni volte alla generazione del modello parametrico. Come conseguenza della fase precedente infatti, non solo è stata completamente abbandonata l'ipotesi dell'interoperabilità diretta caratteristica del primo approccio analizzato, ma è stato ulteriormente escluso l'approfondimento che vedeva interagire l'algoritmo generato su *Grasshopper* con *Revit*. Nell'ottica della progettazione partecipata, che, nonostante sia trattata come fase finale, ha influenzato di fatto l'intera ricerca, uno degli obiettivi impliciti della tesi era dimostrare la possibilità che si potesse sfruttare un **unico modello parametrico su molteplici piattaforme BIM**, ipotizzando l'eventualità in cui gli attori coinvolti nel processo edilizio lavorino con software differenti fra loro. In tal senso lo scopo può essere ritenuto raggiunto grazie agli esiti dei test d'interoperabilità dimostrati nella fase precedente. Pertanto si è scelto, per motivi pratici e tempistici, di condurre la restante parte della ricerca basandosi sulla sola interazione tra *Grasshopper* ed *ArchiCAD*.

La creazione dell'**algoritmo**, per la generazione del modello parametrico della camera di degenza e dell'eventuale servizio annesso, è stata impostata basandosi su due fattori fondamentali. Il primo è quello tradizionalmente in uso e corrisponde alla definizione degli **aspetti geometrici**, che costituiscono i volumi ed il posizionamento dei singoli componenti, mentre il secondo è l'introduzione della descrizione dei **vincoli** che caratterizzano questi ultimi. Tale dicotomia ha contraddistinto l'intero approccio rendendolo contemporaneamente flessibile e rigido.

Ciò che caratterizza un software di *visual editing*, come *Grasshopper*, è la possibilità di dar vita alle forme più disparate, avendo come unico limite teorico la creatività, e come limite pratico la conoscenza delle formule e delle funzioni matematiche che compongono le configurazioni desiderate. Questo è di fatto il motivo principale per cui tali programmi vengono solitamente impiegati, specie se in associazione con piattaforme BIM, la cui generazione di **oggetti complessi** e pa-

rametricamente editabili nel tempo risulta molto complessa. La definizione dell'algoritmo di una degenza, che possa essere adattato sia in base alla superficie disponibile sia al numero di posti letto che deve ospitare, porta con sé una serie di **restrizioni geometriche**, che dipendono dalla normativa di riferimento adottata. Tali vincoli fanno sì che si debbano inevitabilmente limitare i movimenti di oggetti e componenti nello spazio in modo da "guidare" e facilitare la progettazione della camera. In sostanza, l'elasticità, che caratterizza la modellazione parametrica, si scontra con la necessità di rispettare delle regole, dando vita al "**paradosso**" che sancisce l'apparente contraddizione intrinseca della metodologia analizzata.

L'esigenza di irrigidire un sistema che è naturalmente flessibile è nata dall'idea di dover rendere la gestione dell'algoritmo sufficientemente semplice da essere utilizzato autonomamente da un utente professionista che non l'abbia creato. In questa fase si suppone infatti che, in un ambiente reale, gli attori coinvolti siano appartenenti sia al **settore informatico** (IT: *Information Technology*) sia all'industria delle costruzioni. Così facendo si ottimizzerebbero tempi e risorse impiegati per la parte di creazione dello *script*, essendo curata da appositi professionisti. Questa è una delle caratteristiche che contraddistingue maggiormente l'approccio analizzato rispetto al primo testato nella fase precedente: la collaborazione con attori esterni al settore edilizio permette che i risultati così ottenuti siano ottimizzati e funzionali alla definizione specifica della camera di degenza. Ne consegue che il punto di forza di tale approccio consista proprio nei vincoli, che limitano le collisioni geometriche dei componenti dell'algoritmo. Più ne vengono creati, più facile diventa il compito di modellare la degenza, dovendo operare solo entro i limiti di opzioni predeterminate anziché decidere deliberatamente dimensioni e posizionamento dei singoli oggetti. Inoltre, la definizione dei vincoli è composta da una serie di **parametri e valori numerici** che possono essere facilmente adattati a una normativa differente da quella con cui sono stati creati, ovvero il manuale dell'architetto [40]. Questa caratteristica giustifica, almeno in parte, le **tempistiche** impiegate per l'allestimento dell'algoritmo che, nonostante richieda un lungo periodo di preparazione, esso viene compensato dalla possibilità di essere aggiornato a differenti normative ed al susseguirsi dei loro relativi aggiornamenti nel tempo.

Il vero valore aggiunto, grazie all'implementazione dei vincoli all'interno dell'algoritmo, consiste nella definizione di una serie di **regole "non visibili"** create con lo scopo di minimizzare la probabilità di incorrere in errori ed incongruenze. Si è scelto di vincolare infatti solamente gli aspetti che costituiscono il rispetto delle normative di sicurezza e delle direttive sanitarie, come la distanza minima che deve incorrere fra i letti dei pazienti, o il senso di apertura delle porte. Tali restrizioni rappresentano dettagli che spesso rischiano di passare inosservati e che, se mal progettati, possono contribuire alla rivisitazione della distribuzione in fasi più avanzate del progetto, quando comportano un maggior impiego di costi e risorse. Questo implica che tutto ciò che non rientra in questa tipologia di regole non è stato rite-

nuto necessario motivo di essere vincolato. Pertanto, in fase di generazione del modello, da una parte si trovano delle invisibili **linee guida** che permettono di non uscire dai confini della normativa, dall'altro viene mantenuta una libertà di progettazione tale da rendere possibili collisioni tra oggetti facili da vedere quanto da correggere nell'immediato. I vincoli rendono quindi l'algoritmo una **solida base semi-automatizzata**, in quanto, da un lato vi è l'automatismo dettato dalle restrizioni poste, dall'altro tale meccanismo è messo a punto dagli utenti stessi che intendono adoperarlo. Essa permette una modellazione più precisa e sicura, rispetto ad una tradizionale in cui invece la responsabilità del rispetto della normativa dipende unicamente dalla prudenza posta dall'utente in fase di progettazione. Nello specchio sottostante sono riassunti i risultati principali che contraddistinguono l'approccio metodologico analizzato da uno tradizionale.

Modellazione tradizionale	Modellazione algoritmica
Ogni degenza modellata singolarmente	Unico algoritmo adattato alle necessità di ogni degenza
Consultazione costante della normativa di riferimento	Unica messa a punto iniziale dei parametri costituenti la normativa
Verifica manuale del rispetto della normativa	Impossibilità di oltrepassare i vincoli imposti da normativa
Assenza di un metodo rigoroso	Presenza di una metodologia precisa, aggiornabile nel tempo e versatile all'adattamento ad altre normative
Alta probabilità di errore	Bassa probabilità di errore

Il metodo analizzato, che prevede l'utilizzo del *plug-in Grasshopper – ArchiCAD Live Connection*, non è stato creato per lo scopo con cui è stato invece impiegato in questa tesi di ricerca. Esso è infatti incentrato, come anticipato, sulla potenzialità di realizzare modelli dalle forme complesse, che sarebbero difficilmente ottenibili direttamente attraverso la piattaforma BIM. Per questo motivo si sono presentate delle **imperfezioni** “inattese” che hanno parzialmente alterato un processo progettuale che sarebbe altrimenti stato più immediato e preciso. È di fatto reso necessario, prima di implementare l'aspetto informativo del BIM a quello geometrico-parametrico dell'algoritmo, fare un passaggio intermedio di “pulizia” in cui correggere queste imprecisioni. Su *ArchiCAD* bisogna quindi modificare incongruenze, come la correzione del punto di ancoraggio di un elemento, o l'ordine di visualizzazione degli oggetti, che non vengono rappresentati correttamente. A questo genere di criticità si aggiunge inevitabilmente la necessità di realizzare un **template** che consenta di ridurre i tempi di progettazione impostando come standard la qualità grafica con cui il modello viene rappresentato, il livello di dettaglio e la quantità di informazioni contenute al suo interno.

La produzione dello *script*, che compone l'algoritmo per la generazione del modello, e quella del *template*, che ne consente una corretta visualizzazione ed implementazione informativa, sono accomunate dal processo progettuale che caratterizza la metodologia del BIM. Si può infatti notare come i tempi di preparazione del modello, rispetto a quelli di creazione e gestione, siano molto più elevati, in accordo con la teoria espressa dal **diagramma di MacLeamy**. È infatti evidente come un approccio del genere costituisca un grande impiego di risorse nelle fasi iniziali, ma al contempo possa farne risparmiare altrettante nelle fasi successive, durante le quali è possibile concentrarsi su altri aspetti del progetto.

3.3 Fase 3: Un approccio olistico alla progettazione

Le tematiche trattate nella terza fase della ricerca investono in realtà tutta la tesi, in quanto riguardano l'insieme dei concetti imprescindibili che hanno tenuto le redini invisibili del suo intero svolgimento. In primo luogo, grazie al tipo di progettazione partecipata esaminata, è possibile contestualizzare la metodologia analizzata all'interno del processo edilizio. Si tratta infatti di un approccio attuabile in **fase progettuale**, sia come strumento di progettazione stesso sia come verifica delle varie modifiche attuate in collaborazione con i diversi attori che interagiscono al progetto. La partecipazione è prevista, pertanto, come principio alla base del processo, in quanto elemento chiave per il miglioramento dello stesso e per la riduzione di incongruenze ed incomprensioni nell'arco delle sue fasi più avanzate. Essa implica di fatto che il dialogo tra professionisti di settori differenti, ma anche quello con i committenti, sia sempre attivo e facilitato dalle soluzioni messe in atto, di cui nella presente ricerca ne è stata analizzata solamente una.

Un banale esempio di un processo progettuale in fase decisionale, che mette in atto l'approccio analizzato, può essere rappresentato da un gruppo di architetti ed uno di strutturisti che collaborano per impostare una degenza. I primi configurano la camera in modo da avere tre letti, di cui uno posto frontalmente agli altri, e ipotizzano una luce tra i muri divisorii pari a 6,20 m. Dopo aver preso visione del modello così generato gli strutturisti possono però decidere che, in base all'area del complesso sanitario in cui dovranno essere poste le degenze, la maglia strutturale non può essere superiore a 6,00 m e inviano nuovamente il modello, con tale annotazione, al team di architetti, i quali possono facilmente verificare, grazie all'algoritmo preimpostato, che per ottenere tale risultato è necessario o diminuire i numeri dei letti per stanza, o decidere di porre le degenze in una differente ala della struttura. Si può ben immaginare come effettuare modifiche del genere possa richiedere una grande quantità di risorse per uno studio, se effettuata in una fase troppo avanzata del progetto, e che quindi, grazie ad un'attiva partecipazione di tutti gli attori coinvolti che permette un **dialogo costante**, è possibile chiarire ogni genere di dubbio ed incomprensione sul progetto, prima che diventi troppo oneroso porvi

rimedia. La progettazione partecipata, in questa accezione di cooperazione, può tuttavia essere applicata non solo a progetti *ex-novo*, ma anche per verifiche effettuate su strutture esistenti. Nonostante la modellazione algoritmica analizzata non sia un processo bidirezionale, in cui lo *script* sarebbe in grado di “leggere” i parametri di un modello, è comunque possibile, dopo aver effettuato un sopralluogo, impostare all’interno dell’algoritmo i valori rilevati per ottenere un immediato riscontro sulla verifica del rispetto delle normative.

Nella fase finale della ricerca, oltre ad approfondire le tematiche della progettazione partecipata, sono stati anche analizzati i requisiti che hanno reso indispensabile l’utilizzo di un ulteriore applicativo per la realtà virtuale. Si pone infatti come limite del programma di editor visuale l’incapacità di trasmettere una **percezione qualitativa** degli ambienti stessi che permette di generare. Se infatti da un lato i parametri consentono un matematico accertamento dei vincoli da normativa, dall’altro necessitano di essere implementati con un differente sistema di visualizzazione. In puro ambito progettuale tale strumento è rappresentato dalla piattaforma BIM, che lega le geometrie, generate attraverso la manipolazione dei parametri, alle informazioni contenute nel *template* che caratterizzano ogni oggetto del modello. Nell’ottica della progettazione partecipata questo non rappresenta tuttavia una risposta soddisfacente per tutti gli attori coinvolti, che necessitano di venire a conoscenza di entrambi i sopracitati aspetti del modello, anche senza l’obbligo di possedere una licenza del software. È quindi stato necessario l’utilizzo di **BIMx**, che ha racchiuso al contempo le esigenze di visualizzazione delle informazioni tecniche, per i professionisti, di dati economici, per committenti e finanziatori e di ingombri spaziali e volumi per la restituzione di una visione d’insieme del progetto in analisi, sia a schermo che in realtà virtuale immersiva tramite visore.

I risultati ottenuti dipendono inevitabilmente dalla scelta dello strumento e delle tecnologie impiegate, che, nella presente ricerca, sono state mirate a precisi requisiti a scopo dimostrativo, ma che possono includere molteplici altre soluzioni differenti. La scelta della piattaforma BIM e del software impiegato per la parametrizzazione, unitamente a quella in merito all’approccio alla progettazione partecipata, rappresentano pertanto variabili che consentirebbero di conseguire esiti diversi, per quanto affini.



4

Conclusione e sviluppi futuri

4 Conclusione e sviluppi futuri

Il futuro dell'industria delle costruzioni è volto al *Building Information Modeling*, metodologia che sta applicando un grande cambiamento a tutti i suoi settori, integrando vantaggi a lungo termine per l'intero ciclo di vita di un edificio o infrastruttura. Tale trasformazione comporta tuttavia ancora molta confusione da parte dei professionisti, che vedono non solo l'incontestabile sviluppo tecnologico, ma anche la **rottura degli schemi** che hanno contraddistinto la modalità lavorativa degli ultimi decenni. In questo contesto è perciò fondamentale condurre ricerche grazie alle quali sperimentare nuovi approcci metodologici che aiutino a chiarire quale direzione intraprendere per stare al passo con l'annunciato progresso.

La presente ricerca si è occupata di una piccola porzione del problema generale che interessa il **settore sanitario**, le cui strutture richiedevano una metodologia in grado di gestire una vasta quantità di funzionalità, complicati sistemi impiantistici ed innumerevoli dati, ancora prima dell'imposizione del BIM negli appalti pubblici. Si è trattato infatti di sfruttare la ripetitività funzionale, caratteristica di questi complessi, per analizzare un nuovo approccio metodologico che implicasse la possibilità di ottimizzarne la progettazione. In particolare è stata affrontata la tematica della funzione di **degenza**, applicando le linee guida provenienti dal manuale dell'architetto [40]. Bisogna infatti sottolineare che l'approfondimento è stato attuato con maggior interesse alla metodologia rispetto alla specificità dell'ambiente creato, che risulta quindi generico e può essere ulteriormente implementato in base al reparto a cui la degenza è destinata e alle singole esigenze progettuali.

I risultati ottenuti permettono di riflettere sullo stato dell'arte dello sviluppo tecnologico che, per quanto stia facendo passi da gigante, non sempre risulta all'altezza delle necessità di rinnovo dell'industria delle costruzioni. Si è riscontrata infatti una forte correlazione tra gli obiettivi posti e la modalità con cui raggiungerli, influenzando così anche gli esiti delle ipotesi testate. Nonostante la promozione di software e formati di interscambio *Open Source*, il **monopolio delle software house** prevale sulla libertà di scelta dei programmi da impiegare e sul modo con cui questi interagiscono fra loro. È questo di fatto il

principale motivo del fallimento della prima ipotesi analizzata nella ricerca e delle difficoltà riscontrate nell'integrazione fra modellazione parametrica e BIM.

Ad influenzare ulteriormente i risultati ottenuti sono state anche le **competenze** richieste per affrontare l'approccio metodologico analizzato. Lasciando da parte il supporto teorico, è stato infatti necessario apprendere come autodidatta le funzionalità e potenzialità di tutti gli applicativi proposti, compromettendo anche le tempistiche dell'intera ricerca. Si lascia quindi supporre la possibilità che professionisti specializzati o utenti più esperti conquisterebbero risultati più efficienti partendo dalle medesime premesse ed obiettivi di base. La scelta di impiegare una piattaforma BIM di cui non si conosceva il linguaggio di programmazione è solo un esempio dell'azzardo a cui si è deciso di esporsi per condurre la presente tesi di ricerca.

Nonostante le numerose criticità riscontrate, gli esiti conseguiti rappresentano una base solida per ulteriori approfondimenti in merito all'efficientamento dei tempi e delle risorse richiesti per la progettazione di complessi sanitari. L'approccio analizzato, come anticipato, ha trattato una singola funzione, ma è possibile applicarlo a qualunque ambiente che richieda una **ripetitività** costante in un progetto, come sale operatorie o, cambiando destinazione d'uso, anche aule all'interno di un edificio scolastico. La versatilità della metodologia rappresenta infatti uno dei suoi punti di forza, sia per campo di applicazione sia per adattabilità nel tempo. Inoltre, la decontestualizzazione di un ambiente dall'interezza del modello a cui appartiene lo rende il **prototipo ideale** per specifiche analisi e studi, come verifiche energetiche, illuminotecniche, acustiche, ma anche per la realizzazione di piani di manutenzione e gestione o stime economiche.

Le potenzialità offerte dalla presente ricerca per l'ottimizzazione della progettazione e della gestione dell'intero ciclo di vita di un manufatto sono tuttavia ancora più ampie per quanto riguarda l'implementazione tecnologica in merito all'**approccio partecipativo**. Questo aspetto è stato infatti ridotto a semplici applicativi, allo scopo di dimostrare uno dei possibili flussi di lavoro adottabili. Tuttavia, grazie all'impiego di motori grafici come *Unreal Engine* o *Unity* sarebbe possibile implementare l'interazione con i parametri direttamente tramite soluzioni di realtà virtuale immersiva. Per fare ciò bisognerebbe programmare degli appositi componenti che rielaborino gli *script* visuali in modo da poterli gestire direttamente tramite dei *controller* ed un visore, che restituisca nell'immediato la percezione visiva di ciò che viene editato. Questo consentirebbe un ampliamento della consapevolezza degli spazi che renderebbe ancora più visibili le "incongruenze" che non sono state vincolate tramite lo *script*. Inoltre si può così incrementare il livello di cooperazione, rendendo più immediato il dialogo fra gli attori del processo.

Lo sviluppo tecnologico è ormai imprescindibile supporto all'avanzamento delle metodologie che caratterizzano l'industria delle costruzioni. Si potrebbe anche aggiungere che senza il primo probabilmente

il secondo non sarebbe mai avanzato quanto oggi è possibile intravedere. In conclusione, questa ricerca ha voluto, nel suo piccolo, analizzare una delle possibilità che tale progresso, nell'attuale contesto tecnologico, potrebbe portare all'interno del settore edilizio. Ciononostante, la relazione fra lo sviluppo dell'industria delle costruzioni e quello tecnologico si sta affinando ininterrottamente, lasciando largo spazio a studi che, in futuro, porteranno con tutta probabilità a risultati sempre più efficienti per quanto riguarda la sostenibilità degli edifici, la loro gestione nel tempo e, di conseguenza, anche la qualità di vita delle persone che ne fruiscono.



5

Allegati digitali

5 Allegati digitali

I seguenti *QR code* contengono immagini e video, a scopo dimostrativo e di approfondimento, relativi ad alcuni capitoli della metodologia di ricerca. Per un'efficiente visualizzazione dei contenuti si consiglia l'utilizzo delle applicazioni "Google Drive" e "YouTube".

2.1 Fase 1

2.1.5 Parametrizzazione ed interoperabilità orizzontale indiretta



1 – Script complessivo ottenuto tramite i test dell'approccio parametrico della prima fase di ricerca.

2.2 Fase 2

2.2.1 Definizione dell'algoritmo e dei vincoli progettuali



2 – Script delle porzioni di algoritmo definite nella seconda fase di ricerca.

2a – Muri esterni ed interni

2b – Porta d'ingresso della degenza

2c – Finestre

2d – Solaio, soffitto e zone



2e – Posti letto

2f – Arredo dipendente dai posti letto: comodini e travi testaletto



2g – Arredo dipendente dal numero di posti letto: armadi, tavoli e sedie

2h – Bagno annesso al locale degenza



2i – Porta del bagno

2l – Attrezzature igienico-sanitarie del locale bagno





3 – Video dimostrativo della modellazione parametrica di camere di degenza, attraverso la connessione tra Grasshopper ed ArchiCAD

2.3 Fase 3

2.3.3 Progettazione partecipata: *BIMx*



4 – Video dimostrativo dell'applicazione di *BIMx* alla progettazione partecipata

Riferimenti: bibliografia e sitografia

- [1] A. Marra, «*Il futuro dell'edilizia passa per il BIM* (EdilPortale)», 15 febbraio 2018. Disponibile Online: https://www.edilportale.com/news/2018/02/focus/il-futuro-dell-edilizia-passa-per-il-bim_62317_67.html. [Consultato il giorno 01 settembre 2019].
- [2] Decreto Ministeriale D.M. MIT n. 560, 1 dicembre 2017.
- [3] D. C. Engelbart, «*Augmenting Human Intellect: A Conceptual Framework*», Summary Report AFOSR-3233, Menlo Park, California, 1962.
- [4] M. Carradori, «*BIM – Building Information Modeling: i principi di una rivoluzione* (Bis-Lab – Building Innovation & Skills-Lab)», 28 febbraio 2017. Disponibile Online: http://www.bis-lab.eu/2017/02/28/bim-building-information-modeling-i-principi-di-una-rivoluzione/#_edn2. [Consultato il giorno 27 gennaio 2019].
- [5] A. Osello, *Il futuro del disegno con il BIM per Ingegneri e Architetti*, Palermo: Dario Flaccovio Editore, 2012.
- [6] C. Eastman e Others, «*An Outline of the Building Description System*», Research Report n. 50, Carnegie-Mellon University, Pittsburgh, 1974.
- [7] C. Eastman, P. Teicholz, R. Sacks e K. Liston, *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, Hoboken, New Jersey: John Wiley & Sons, Inc., 2011.
- [8] Cooperative Research Centre for Construction Innovation, «*National Guidelines for Digital Modelling*», Brisbane, Australia, 2009.
- [9] Digital Construction, (bips), «*3D Working Method 2006*», Ballerup, Denmark, 2007.
- [10] National Institute of Building Science, «*National Building Information Modeling Standard. Version 1 - Part 1: Overview, Principles and Methodologies. Transforming the building supply chain through open and interoperable information exchanges*», United States, 2007.
- [11] Graphisoft, «*Open BIM*», Disponibile Online: https://www.graphisoft.com/it/soluzioni/open_bim/about_bim/. [Consultato il giorno 22 gennaio 2019].
- [12] D. Quadretti, «*Il BIM Building Information Modeling secondo Charles Eastman* (888 Software Products)», 27 giugno 2017. Disponibile Online: <https://blog.888sp.com/bim-building-information-modeling-se->

- condo-charles-eastman/*. [Consultato il giorno 22 gennaio 2019].
- [13] D. Davis, «*The Macleamy curve* (“DanielDavis” blog)», 15 ottobre 2011. Disponibile Online: <http://www.danieldavis.com/macleamy>. [Consultato il giorno 28 gennaio 2019].
- [14] M. M. Bocconcino, M. Del Giudice e F. Manzone, *Il disegno e l'ingegnere. Il disegno e la produzione edilizia tra tradizione e innovazione*, Torino: Levrotto e Bella, 2016.
- [15] «*IFC, cos'è: caratteristiche, vantaggi e importanza della certificazione IFC* (Biblus-net)», 30 novembre 2017. Disponibile Online: <http://biblus.acca.it/focus/ifc-cose-e-quali-sono-i-vantaggi/>. [Consultato il giorno 4 febbraio 2019].
- [16] L. Nissim, «*IFC: cos'è? e come è fatto?* (iBIMi)», 30 dicembre 2015. Disponibile Online: <http://www.ibimi.it/ifc-cose-e-come-e-fatto/>. [Consultato il giorno 17 gennaio 2019].
- [17] «*IFC cos'è e a cosa serve? Qual è il legame BIM-IFC?* (BibLus BIM)», 22 maggio 2017. Disponibile Online: <http://bim.acca.it/legame-bim-ifc/>. [Consultato il giorno 4 febbraio 2019].
- [18] C. C. Rizzarda e G. Gallo, *La sfida del BIM. Un percorso di adozione per progettisti e imprese*, Milano: Tecniche nuove, 2017.
- [19] M. Bolpagni, «*The many faces of 'LOD' (BIM Think Space)*», 19 luglio 2016. Disponibile Online: <https://www.bimthinkspace.com/2016/07/the-many-faces-of-lod.html>. [Consultato il giorno 7 febbraio 2019].
- [20] Norma UNI 11337-4:2017, «*Edilizia e opere di ingegneria civile - Gestione digitale dei processi informativi delle costruzioni - Parte 4: Evoluzione e sviluppo informativo di modelli, elaborati e oggetti*».
- [21] «*Level of Development specification guide* (BIM Forum)», novembre 2017. Disponibile Online: https://bimforum.org/wp-content/uploads/2017/11/LOD-Spec-2017-Guide_2017-11-06-1.pdf. [Consultato il giorno 13 febbraio 2019].
- [22] PAS 1192-2:2013, «*Specification for information management for the capital/delivery phase of construction projects using building information modelling* (BSI)», London, United Kingdom, 2013.
- [23] E. Sattanino, «*I LOI nelle scale di LOD USA, UK, ITA* (Ingenio)», 12 febbraio 2018. Disponibile Online: <https://www.ingenio-web.it/18926-i-loi-nelle-scale-di-lod-usa-uk-ita>. [Consultato il giorno 12 febbraio 2019].
- [24] F. Sabatini e V. Coletti, «*Dizionario di Italiano “il Sabatini Coletti”*», Corriere della Sera, 2008. Disponibile Online: https://dizionari.corriere.it/dizionario_italiano/P/parametro.shtml. [Consultato il giorno 19 febbraio 2019].
- [25] «*Design parametrico e progettazione algoritmica* (FabLab Venezia)», Disponibile Online: <https://www.fablabvenezia.org/design-parametrico-e-progettazione-algoritmica/>. [Consultato il giorno 19 febbraio 2019].
- [26] L. Bertazzoni, «*La progettazione Feng Shui per l'architettura parametrica* (Teknoring)», 8 ottobre 2015. Disponibile Online: <https://www.teknoring.com/guide/guide-architettura/la-progettazione-feng-shui-per-larchitettura-parametrica/?sso=J1AR6QOEACV-DKLM8PFRGDJ9JHO>. [Consultato il giorno 21 febbraio 2019].

- [27] A. Tedeschi, *Architettura parametrica. Introduzione a Grasshopper: il plug-in per la modellazione generativa in Rhino*, IIa edizione, Brienza (PT): Le Penseur, 2010.
- [28] A. Tedeschi, «*Il processo è più importante del risultato, AAD Algorithmic Aided Design* (Arturo Tedeschi)», Disponibile Online: http://www.arturotedeschi.com/wordpress/?page_id=1475 . [Consultato il giorno 19 febbraio 2019].
- [29] «*Computational design, algoritmi generativi e modellazione parametrica* (Digital Arts & Manufacturing Academy)», 2 settembre 2016. Disponibile Online: <https://www.dama.academy/computational-design-algoritmi-generativi-e-progettazione-parametrica/>. [Consultato il giorno 27 febbraio 2019].
- [30] «*BIM e progettazione algoritmica: connubio di successo Rhino-Grasshopper-ARCHICAD* (Edilportale)», 19 settembre 2016. Disponibile Online: <https://www.edilportale.com/news/2016/09/bim-news/bim-e-progettazione-algoritmica-connubio-di-successo-rhino-Grasshopper>. [Consultato il giorno 25 febbraio 2019].
- [31] Autodesk, «*Prodotti Revit: Dynamo*», Disponibile Online: <https://knowledge.autodesk.com/it/support/Revit-products/learn-explore/caas/CloudHelp/cloudhelp/2018/ITA/Revit-Customize/files/GUID-F45641B0-830B-4FF8-A75C-693846E3513B-htm.html>. [Consultato il giorno 11 marzo 2019].
- [32] A. Tedeschi, Conferenza Graphisoft Italia (a cura di): «*Oltre il BIM: nuove forme di collaborazione con i principi di progettazione parametrica*», 28-29 marzo 2018, Milano. Disponibile Online: <https://www.youtube.com/watch?v=mcuWlg-H7CA>. [Consultato il giorno 26 febbraio 2019].
- [33] G. Ramírez, Conferenza Graphisoft Italia (a cura di): «*Oltre il BIM: nuove forme di collaborazione con i principi di progettazione parametrica*», 28-29 marzo 2018, Milano. Disponibile Online: <https://www.youtube.com/watch?v=xuffIemQp-4>. [Consultato il giorno 26 febbraio 2019].
- [34] B. Casella, «*Il BIM: uno strumento innovativo per le strutture ospedaliere* (Health online)», 10 luglio 2018. Disponibile Online: <https://healthonline.healthitalia.it/bim-uno-strumento-innovativo-le-strutture-ospedaliere/>. [Consultato il giorno 12 marzo 2019].
- [35] D. Barista, «*BIM for Healthcare. Early adopters of building information modeling technology are seeing great benefits when designing and building hospitals* (Building Design Construction)», 1 febbraio 2007. Disponibile Online: <https://www.bdcnetwork.com/bim-healthcare>. [Consultato il giorno 17 gennaio 2019].
- [36] F. Cappelletti, T. Dalla Mora, F. Peron, P. Romagnoni e P. Ruggeri, *Una panoramica sul Building Information Modelling (BIM)*, Milano: Ai-CARR, 2014.
- [37] «*BIM in Healthcare* (Healthcare WSP)», Disponibile Online: <http://healthcare.wsp-pb.com/building-information-modelling-bim/>. [Consultato il giorno 3 marzo 2019].
- [38] P. Veloso, G. Celani e R. Scheeren, «*From the generation of layouts to the production of construction documents: An application in the custo-*

- mization of apartment plans», *Automation in Construction*, n. 96, pp. 224-235, 2018.
- [39] «Parco della Salute di Torino, la gara al via entro fine mese (TorinOggi)», 11 gennaio 2019. Disponibile Online: <http://www.torinoggi.it/2019/01/11/leggi-notizia/argomenti/politica-11/articolo/parco-della-salute-di-torino-la-gara-al-via-entro-fine>. [Consultato il giorno 25 marzo 2019].
- [40] E. Milone, M. Nicoletti, R. Perris, M. Carmosino e L. Zevi, «*Il Nuovissimo Manuale Dell'architetto*», vol. 1, Roma, M.E. Architectural Book and Review, 2013, pp. 225-259 (sezione B).
- [41] G. Gennaro, «*Il vero ruolo del Parco della Salute di Torino* (Piemonte informa, Regione Piemonte)», 18 gennaio 2019. Disponibile Online: <https://www.regione.piemonte.it/web/pinforma/notizie/vero-ruolo-parco-della-salute-torino>. [Consultato il giorno 25 marzo 2019].
- [42] «*Ecco come sarà il futuro Parco della Salute di Torino* (TorinOggi)», 12 aprile 2018. Disponibile Online: <http://www.torinoggi.it/2018/04/12/leggi-notizia/argomenti/eventi-11/articolo/ecco-come-sara-il-futuro-parco-della-salute-di-torino.html>. [Consultato il giorno 15 ottobre 2018].
- [43] «*L'Ospedale Karolinska Solna in Svezia* (BIM blog)», 23 settembre 2017. Disponibile Online: <https://bimblog.net/2017/09/23/lospedale-karolinska-solna-in-svezia/>. [Consultato il giorno 25 marzo 2019].
- [44] G. Gennaro, «*Accordo di programma per il Parco della Salute di Torino* (Piemonte informa, Regione Piemonte)», 15 novembre 2017. Disponibile Online: <http://www.regione.piemonte.it/pinforma/sanita/1613-accordo-di-programma-per-il-parco-della-salute-di-torino.html>. [Consultato il giorno 15 ottobre 2018].
- [45] M. Lehtinen e M. Zappia, «*La progettazione impiantistica BIM-based per l'ospedale svedese New Karolinska Solna* (Ingenio)», 22 settembre 2018. Disponibile Online: <https://www.ingenio-web.it/21210-la-progettazione-impiantistica-bim-based-per-lospedale-svedese-new-karolinska-solna>. [Consultato il giorno 25 marzo 2019].
- [46] «*Il BIM al servizio degli appalti puliti: Università e Politecnico di Torino propongono un modello per il progetto parco della salute* (Systema)», 25 febbraio 2016. Disponibile Online: <https://systemasrl.it/modello-bim-appalti-puliti-universita-politecnico-torino-parco-salute/>. [Consultato il giorno 28 marzo 2019].
- [47] G. Gennaro, «*Entro fine gennaio la gara per il Parco della Salute di Torino* (Piemonte informa, Regione Piemonte)», 11 gennaio 2019. Disponibile Online: <https://www.regione.piemonte.it/web/pinforma/notizie/entro-fine-gennaio-gara-per-parco-della-salute-torino>. [Consultato il giorno 25 marzo 2019].
- [48] J. Briselli, K. Dowd e O. Elizarova, «*Participatory Design in Practice* (UX Magazine)», 14 dicembre 2017. Disponibile Online: <https://uxmag.com/articles/participatory-design-in-practice>. [Consultato il giorno 18 agosto 2019].
- [49] K. Lewin, «*Action Research and Minority Problems*», *Journal of Social Issues*, vol. 2, 1946.
- [50] I. Iscoe e L. C. Harris, «*Social and Community Interventions*», *Annual*

- Review of Psychology, n. 35, pp. 333-360, 1984.
- [51] A. Coppo e C. Tortone, *Partecipazione e Empowerment. La progettazione partecipata intersettoriale e con la comunità*, Dors: Centro Regionale di Documentazione per la Promozione della Salute, Azienda Sanitaria Locale di Collegno e Pinerolo (A.S.L. TO3), Regione Piemonte, 2011.
- [52] E. B.-N. Sanders, «*From User-Centered to Participatory Design Approaches*», in *Design and the Social Sciences*, J.Frascara (Ed.), Taylor & Francis Books Limited, 2002.
- [53] S. Gladkiy, «*User-Centered Design: Process And Benefits (UX Design)*», 20 luglio 2018. Disponibile Online: <https://producttribe.com/ux-design/user-centered-design-guide>. [Consultato il giorno 2 settembre 2019].
- [54] F. Palatucci, «*Realtà Aumentata e Realtà Virtuale in edilizia: 5 pronostici per il futuro (Graitec)*», 6 agosto 2018. Disponibile Online: <https://www.graitec.it/blog/entry/5-pronostici-per-il-2018-su-realta-aumentata-e-realta-virtuale>. [Consultato il giorno 20 agosto 2019].
- [55] «*BIM, realtà virtuale e realtà aumentata (Biblus-net)*», 30 agosto 2017. Disponibile Online: <http://biblus.acca.it/focus/progettazione-architettura-bim-realta-aumentata-e-realta-virtuale-come-cambia-il-modo-di-progettare-con-levoluzione-tecnologica/>. [Consultato il giorno 11 febbraio 2019].
- [56] A. Salice, «*Gli italiani amano la tecnologia ma fingono di conoscerla (Dday)*», 27 maggio 2016. Disponibile Online: <https://www.dday.it/redazione/20098/gli-italiani-amano-la-tecnologia-ma-fingono-di-conoscerla>. [Consultato il giorno 20 agosto 2019].
- [57] «*Realtà Virtuale (Intelligenza Artificiale)*», Disponibile Online: <http://www.intelligenzaartificiale.it/realta-virtuale/>. [Consultato il giorno 23 gennaio 2019].
- [58] K. O'Connell, «*4 Tips to Get Started With Virtual Reality in Architecture (Redshift, by Autodesk)*», 16 novembre 2016. Disponibile Online: <https://www.autodesk.com/redshift/virtual-reality-in-architecture/>. [Consultato il giorno 20 agosto 2019].
- [59] «*Las mejores aplicaciones para reformar tu casa (Info Casas)*», 30 agosto 2018. Disponibile Online: <https://www.infocasas.com.uy/blog/aplicaciones-reformar-casa>. [Consultato il giorno 20 agosto 2019].
- [60] M. De Mori, «*Realtà virtuale, aumentata o mista? (Focus)*», 6 dicembre 2015. Disponibile Online: <https://www.focus.it/tecnologia/digital-life/realta-virtuale-aumentata-o-mista>. [Consultato il giorno 20 agosto 2019].
- [61] R. Negri, «*Una partnership per la Mixed Reality (BIM portale)*», 12 aprile 2018. Disponibile Online: <https://www.bimportale.com/una-partnership-per-la-mixed-reality/>. [Consultato il giorno 20 agosto 2019].
- [62] E. Dicosola, «*BIM, realtà aumentata e realtà virtuale: quale futuro per l'edilizia? (Building by Close-up Engineering)*», 13 aprile 2019. Disponibile Online: <https://buildingcue.it/bim-realta-aumentata-e-realta-virtuale-quale-futuro-per-ledilizia/12325/>. [Consultato il giorno 20 agosto 2019].

- [63] «*Virtual, augmented and mixed reality are the 4th wave* (Digi-Capital)», 11 luglio 2016. Disponibile Online: <https://www.digi-capital.com/news/2016/07/virtual-augmented-and-mixed-reality-are-the-4th-wave/#.WaZrI5CzG31>. [Consultato il giorno 20 agosto 2019].
- [64] A. Malleon, «*National BIM Survey: summary of findings*», in NBS, National BIM Report 2018, Riba Enterprise Ltd, 2018.
- [65] Piattaforma online «*BIM object*», Disponibile Online: <https://www.bimobject.com/it>. [Consultato il giorno 17 novembre 2018].
- [66] Graphisoft, «*Downloads: IFC Model Exchange with ARCHICAD for Revit*», Disponibile Online: <https://www.graphisoft.com/downloads/interoperability.html>. [Consultato il giorno 17 novembre 2018].
- [67] Graphisoft, «*ArchiCAD Connection Add-In*», Disponibile Online: <https://graphisoft.akamaized.net/cdn/ftp/techsupport/downloads/interoperability/GRAPHISOFT%20ARCHICAD%20Connection%20for%20Revit%202018.pdf>. [Consultato il giorno 22 novembre 2018].
- [68] «*We talk to BIM6x about their Revit RVT translator for ARCHICAD: From BIM6x* (WorldCAD Access)», 11 dicembre 2017. Disponibile Online: <https://www.worldcadaccess.com/blog/2017/12/we-talk-to-bim6x-about-their-Revit-rvt-translator-for-ArchiCAD.html>. [Consultato il giorno 29 settembre 2018].
- [69] Autodesk, «*Prodotti Revit: Impossibile importare o collegare alcuni file IFCXML in Revit*», Disponibile Online: <https://knowledge.autodesk.com/it/support/Revit-products/troubleshooting/caas/sfdcarticles/sfdcarticles/ITA/Fail-to-import-or-link-some-IFCXML-files-in-Revit.html>. [Consultato il giorno 30 novembre 2018].
- [70] Grasshopper Development team, *The Grasshopper Primer*, IIIa edizione, Mode Lab, 2014.
- [71] Piattaforma online «*food4Rhino*», Disponibile Online: <https://www.food4Rhino.com/>. [Consultato il giorno 4 dicembre 2018].
- [72] Lmnts, «*Lyrebird* (Grasshopper3D)», Disponibile Online: <https://www.grasshopper3d.com/group/lyrebird>. [Consultato il giorno 5 dicembre 2018].
- [73] J. Hiroshi, «*Chameleon* (Grasshopper3D)», Disponibile Online: <https://www.grasshopper3d.com/group/chameleon>. [Consultato il giorno 5 dicembre 2018].
- [74] K. K. Sobon, «*Mantis Shrimp* (Grasshopper3D)», Disponibile Online: <https://www.grasshopper3d.com/group/mantis-shrimp>. [Consultato il giorno 5 dicembre 2018].
- [75] Graphisoft, «*Downloads: Rhino - Grasshopper - ARCHICAD Toolset*», Disponibile Online: <https://www.graphisoft.com/downloads/addons/interoperability/rhino.html#live-connection-plugin>. [Consultato il giorno 22 novembre 2018].
- [76] Piattaforma online “food4Rhino”, «*Hummingbird* (by WhiteFeet Software)», Disponibile Online: <https://www.food4Rhino.com/app/hummingbird>. [Consultato il giorno 5 dicembre 2018].
- [77] Piattaforma online “food4Rhino”, «*Grevit - Grasshopper Native BIM* (by moethu)», Disponibile Online: <https://www.food4Rhino.com/app/Grevit-Grasshopper-native-bim>. [Consultato il giorno 5 dicembre

- 2018].
- [78] A. Sedlenieks, «*Doors import problem (Grasshopper3D)*», 6 gennaio 2016. Disponibile Online: <https://www.grasshopper3d.com/group/hummingbird/forum/topics/doors-import-problem>. [Consultato il giorno 5 dicembre 2018].
- [79] Graphisoft, «*User manuals: Predefined Pen Sets for Specific Functions*», Disponibile Online: <https://helpcenter.Graphisoft.com/user-guide/64857/>. [Consultato il giorno 14 agosto 2019].
- [80] Graphisoft, «*Knowledgebase: Virtual Reality (VR) for ARCHICAD*», 5 marzo 2019. Disponibile Online: <https://helpcenter.graphisoft.com/knowledgebase/86526/>. [Consultato il giorno 18 giugno 2019].
- [81] Graphisoft, «*BIMx*», Disponibile Online: <https://bimx.ArchicAD.com/it/#what-is-bimx>. [Consultato il giorno 18 giugno 2019].
- [82] «*Google Cardboard: cos'è e come funziona? (Regiverse: Virtual Reality made easy)*», 28 febbraio 2018. Disponibile Online: <https://regiverse.com/blog/2018/google-cardboard-cose-funziona/>. [Consultato il giorno 24 agosto 2019].