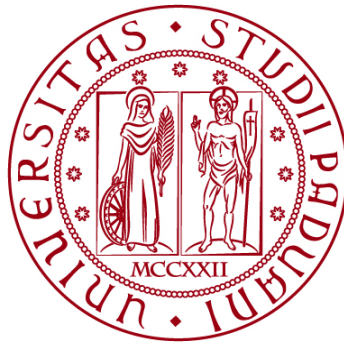


UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA CIVILE, EDILE E AMBIENTALE
Department Of Civil, Environmental and Architectural Engineering

Corso di Laurea Magistrale a ciclo unico in Ingegneria Edile-Architettura



TESI DI LAUREA

**INDIVIDUAZIONE DI PROTOCOLLI INTEROPERABILI PER IL
DIGITAL TWIN IN AMBITO STRUTTURALE:
PROPOSTA DI UNA METODOLOGIA BASATA SU
STANDARD OPENBIM IFC E NASTRAN**

Relatore: prof. Carlo Zanchetta
Correlatori: Ing. Giorgia Marcellino

Laureanda: Alice Pin

ANNO ACCADEMICO 2025-2026

SOMMARIO

ABSTRACT	1
INTRODUZIONE.....	3
CAPITOLO PRIMO ANALISI DELL'AMBITO DI RIFERIMENTO	5
1 FORMATI, STANDARD, MODELLI DATI.....	5
1.1 <i>Definizione di Software</i>	5
1.2 <i>Definizione di File</i>	5
1.3 <i>Formati proprietari e formati aperti</i>	5
1.4 <i>Definizione di Standard</i>	7
1.5 <i>Definizione di Modello Dati</i>	7
2 BIM PER LA STRUCTURAL ANALYSIS	10
2.1 <i>Definizione di BIM</i>	10
2.2 <i>LOD</i>	12
2.3 <i>Vantaggi del BIM</i>	12
2.4 <i>Formati Aperti per il BIM: lo Standard IFC</i>	14
2.5 <i>IFC Structural Domain</i>	18
3 FEM PER LA STRUCTURAL ANALYSIS.....	20
3.1 <i>FEM (Finite Element Method)</i>	20
3.2 <i>Nascita e diffusione del FEM</i>	21
3.3 <i>Software FEM</i>	21
3.4 <i>Formati per il FEM</i>	22
4 INTEROPERABILITA' BIM-FEM	24
4.1 <i>Definizione di Interoperabilità</i>	24
4.2 <i>Quadro Normativo</i>	25
4.3 <i>Interoperabilità BIM-FEM</i>	27
4.4 <i>Professionisti interessati</i>	27
4.5 <i>Vantaggi di una possibile soluzione</i>	28
CAPITOLO SECONDO ASPETTI METODOLOGICI NELL'INTEROPERABILITÀ BIM-FEM	29
1 ANALISI DELLA LETTERATURA	29
1.1 <i>Confronto di file</i>	29
1.2 <i>Link diretti</i>	33
1.3 <i>Database intermedi</i>	35
2 CONFRONTO TRA MODELLI DATI	37
2.1 <i>Rappresentazione geometrica e connessioni</i>	37
2.2 <i>Informazioni contenute e struttura dei dati</i>	38
2.3 <i>Diversi livelli di funzionalità</i>	39
3 OBIETTIVO DELLA RICERCA TESI	40
CAPITOLO TERZO METODOLOGIA E CASO STUDIO	43
1 NASTRAN	44
1.1 <i>Modello dati Nastran</i>	44
1.2 <i>Strand7</i>	48
2 STUDIO DELLO STANDARD IFC.....	50
2.1 <i>IFCCOLUMN</i>	50
2.2 <i>IFCBEAM</i>	61

3	POPOLAMENTO POPOLAMENTO DEL FILE NASTRAN.....	62
3.1	<i>GRID – PERCORSI</i>	62
3.2	<i>MAT1 – PERCORSI</i>	63
3.3	<i>PBEAM – PERCORSI</i>	64
3.4	<i>CBEAM – PERCORSI</i>	68
4	SVILUPPO DI UN APPLICATIVO.....	70
4.1	<i>Script</i>	70
5	VALIDAZIONE.....	73
5.1	<i>Revit</i>	73
5.2	<i>Materiale casi studio</i>	79
5.3	<i>Caso studio 1: colonna</i>	80
5.4	<i>Caso studio 2: trave</i>	82
5.5	<i>Caso studio 3: telaio</i>	83
5.6	<i>File dei casi studio</i>	85
CAPITOLO QUARTO RISULTATI.....		87
1	ANALISI CRITICA DEI RISULTATI.....	87
1.1	<i>Caso studio 1: risultati</i>	87
1.2	<i>Caso studio 2: risultati</i>	89
1.3	<i>Caso studio 3: risultati</i>	89
2	CRITICITA' DEL METODO.....	91
CAPITOLO QUINTO CONCLUSIONI E SVILUPPI FUTURI.....		93
APPENDICE 1 ARCHITETTURA IFC COLUMN E IFC BEAM		95
APPENDICE 2 SCRIPT.....		99
BIBLIOGRAFIA		107
SITOGRAFIA		111
INDICE DELLE TABELLE.....		113
INDICE ICONOGRAFICO.....		115

ABSTRACT

Il settore delle costruzioni affronta sfide crescenti in termini di efficienza, sostenibilità e qualità, che richiedono la collaborazione di diverse discipline. La frammentazione dei dati e la difficoltà di comunicazione tra i professionisti, che utilizzano software specifici e formati proprietari, generano dispersioni di informazioni, errori e rallentamenti. Per superare queste barriere, il Building Information Modeling (BIM) si propone come un approccio olistico per la gestione delle informazioni lungo il ciclo di vita dell'opera. Attraverso un modello digitale parametrico unico e condiviso, il BIM facilita la generazione automatizzata di elaborati grafici e promuove una collaborazione fluida tra architetti, ingegneri e impiantisti, consentendo verifiche in tempo reale e una gestione più efficace dei progetti.

In questo contesto, la presente tesi di laurea esplora e sviluppa tecniche per un'interoperabilità efficace tra l'ambiente BIM, tramite il formato aperto Industry Foundation Classes (IFC), e i software di analisi strutturale agli elementi finiti (FEM), con particolare attenzione al formato proprietario NASTRAN. Nella ricerca, dopo aver definito i concetti di formato aperto, standard e modello dati e aver approfondito i quadri normativi BIM e FEM, si introducono il concetto di interoperabilità. Successivamente si analizza lo stato dell'arte delle metodologie per l'interoperabilità BIM-FEM, valutandone vantaggi e criticità e confrontando i modelli dati IFC e FEM, così da arrivare ad identificare una metodologia preferenziale per convertire un file IFC in un file FEM per l'analisi strutturale.

A questo punto, studiando approfonditamente le strutture dati di entrambi i tipi di file, è stato sviluppato un applicativo in Python, validato tramite casi studio diversificati. I risultati ottenuti dalla sperimentazione mettono in evidenza la validità della metodologia, che consente di convertire buona parte dei dati.

Altresì vi sono delle problematiche sia relative alla mappatura IFC, sia per quanto riguarda la stesura del codice convertitore, basato su conoscenze informatiche elementari, sia infine, per la semplicità dei casi studio proposti, che fungono solo da esempio e che vanno sicuramente resi più complessi in ottica di sviluppi futuri.

The construction sector faces growing challenges in terms of efficiency, sustainability, and quality, necessitating multidisciplinary collaboration. Data fragmentation and communication difficulties among professionals, who often use specific software and proprietary formats, lead to dispersed information, errors, and delays. To overcome these barriers, Building Information Modeling (BIM) offers a holistic approach to managing project information throughout its lifecycle. Through a single, shared parametric digital model, BIM facilitates the automated generation of graphic deliverables and promotes fluid collaboration among architects, engineers, and plant designers, enabling real-time verification and more effective project management.

In this context, this thesis explores and develops techniques for effective interoperability between the BIM environment, using the open Industry Foundation Classes (IFC) format, and Finite Element Method (FEM) structural analysis software, with a particular focus on the proprietary NASTRAN format. The research begins by defining concepts such as open format, standard, and data model, and delves into the regulatory frameworks for BIM and FEM, introducing the concepts of interoperability and digital twin. Subsequently, the state-of-art methodologies for BIM-FEM interoperability are analyzed, evaluating their advantages and disadvantages and comparing IFC and FEM data models, ultimately identifying a preferred methodology for converting an IFC file into a NASTRAN file for structural analysis.

At this stage, through an in-depth study of the data structures of both file types, a Python application was developed and validated using diverse case studies. The experimental results highlight the validity of the methodology, which allows for the conversion of a significant portion of the data. However, there are also challenges related to IFC mapping, the simplicity of the converter code (based on elementary programming knowledge), and the basic nature of the proposed case studies, which serve only as examples and require greater complexity for future developments.

INTRODUZIONE

Il settore delle costruzioni, in costante evoluzione, si trova oggi ad affrontare sfide sempre più complesse, dettate dalla crescente richiesta di efficienza, sostenibilità e qualità delle opere. La progettazione e la realizzazione di edifici moderni richiedono la collaborazione sinergica di un'ampia gamma di professionisti (architetti, ingegneri strutturisti, impiantisti, geotecnici) la cui interazione è fondamentale per il successo di qualsiasi progetto. Tuttavia, la natura multidisciplinare di questo processo porta con sé una serie di problematiche intrinseche, tra cui la gestione della frammentazione dei dati e la difficoltà di garantire una comunicazione fluida tra le diverse discipline.

Tradizionalmente, ogni professionista opera con strumenti e software specifici per la propria area di competenza, generando modelli e documenti spesso incompatibili tra loro. Questa eterogeneità si traduce in una notevole dispersione di informazioni, in lunghe tempistiche dedicate alla riconciliazione dei dati e, non di rado, nella comparsa di errori progettuali e interferenze in fase esecutiva. La condivisione dei file, ostacolata dall'uso di formati proprietari, diventa un passaggio che rallenta l'intero processo, compromettendo l'efficienza e aumentando i costi. La necessità di superare queste barriere ha spinto verso la ricerca di soluzioni che favoriscano la standardizzazione e l'apertura, con i formati aperti che rappresentano una risposta concreta a queste esigenze, garantendo la libertà di scambio di dati indipendentemente dal software di origine.

In questo contesto, il Building Information Modeling (BIM) emerge come un approccio olistico orientato alla gestione delle informazioni di un'opera edilizia lungo tutto il suo ciclo di vita. Attraverso la creazione di un modello digitale parametrico unico e condiviso, il BIM facilita non solo la generazione automatizzata di elaborati grafici tradizionali (piante, prospetti, sezioni) con una drastica riduzione degli errori, ma soprattutto promuove una collaborazione tra tutte le figure professionali coinvolte. Architetti, ingegneri e impiantisti possono operare sullo stesso modello, verificando in tempo reale la compatibilità delle proprie soluzioni e risolvendo tempestivamente eventuali conflitti o interferenze. Questo approccio collaborativo consente una gestione più efficace del progetto e un controllo più stringente sui tempi e sui costi.

Il BIM negli ultimi anni ha avuto un successo internazionale, e anche l'Italia sta rapidamente adeguando il proprio quadro normativo per accogliere questa metodologia innovativa. Il nuovo Codice dei Contratti Pubblici (D.Lgs. 36/2023) rappresenta un passo fondamentale in questa direzione, rendendo obbligatorio l'utilizzo del BIM a partire dal 1° gennaio 2025 per tutte le nuove opere pubbliche di valore superiore a un milione di euro. Questa disposizione sottolinea l'urgenza e l'importanza per il settore delle costruzioni italiano di allinearsi agli standard internazionali, abbracciando le potenzialità offerte dal BIM per migliorare la qualità e l'efficienza della progettazione.

In questo scenario, uno degli ambiti per l'implementazione del BIM riguarda l'interoperabilità con i modelli per analisi strutturali agli elementi finiti (FEM - Finite Element Method). L'obiettivo di questa tesi di laurea è proprio quello di esplorare e sviluppare tecniche valide che favoriscano un'interoperabilità efficace tra l'ambiente BIM e i software di analisi FEM, superando le attuali frammentazioni e ottimizzando il flusso di lavoro.

Il presente lavoro di ricerca si propone di approfondire la fattibilità e i benefici derivanti dall'integrazione tra il modello BIM, approcciato attraverso il formato aperto Industry Foundation Classes (IFC), standard internazionale per la condivisione di dati nell'edilizia, e il modello strutturale FEM. Per quest'ultimo, è stato scelto di concentrarsi sul formato proprietario NASTRAN, ampiamente diffuso e riconosciuto nell'ingegneria strutturale.

Il primo capitolo di questo elaborato sarà dedicato allo studio dell'ambito di riferimento. In questa sezione, si metteranno in evidenza le definizioni di formato aperto, di standard e di modello dati per avere un inquadramento dei concetti alla base dello studio dei formati IFC e per il FEM. Inoltre, verranno approfonditi i quadri normativi e contestuali in ambito BIM e FEM, introducendo nel dettaglio le strutture dei due tipi di modellazione. Infine, nel capitolo viene approfondito il concetto di interoperabilità e di digital twin, valutando la situazione attuale in ambito di interoperabilità BIM-FEM e i vantaggi che otterrebbero i progettisti da un'implementazione di questo flusso di lavoro.

Il secondo capitolo, attraverso uno studio dello stato dell'arte, va a ricercare delle possibili metodologie che possano essere adottate nell'ambito BIM-FEM. Inoltre, tra le diverse proposte pubblicate, si valuteranno i vantaggi e gli svantaggi di ciascuna, cercando di individuare una direzione preferenziale in cui sviluppare la sperimentazione. Tale studio coinvolge anche e soprattutto un approfondimento sul confronto tra il modello dati IFC e il modello dati FEM, dimostrando i punti di contatto e le differenze tra i due, nonché la complessità del confronto essendo queste ultime consistenti.

Nella terza parte la tesi si sviluppa proponendo una metodologia per convertire un file IFC di input in uno Nastran per l'analisi strutturale. Il metodo adottato studia in maniera approfondita la struttura dati e gli elementi presenti in un file Nastran, dopodiché studia la logica dati e le entità di un file IFC per popolare infine gli elementi Nastran con le informazioni prese da questo. Viene quindi sviluppato un applicativo in Python e validato il metodo attraverso casi studio diversificati.

Il capitolo quarto contiene un'analisi critica dei risultati ottenuti, mettendo in evidenza i difetti del metodo che andranno ripresi e migliorati nelle sperimentazioni future.

Infine, il quinto capitolo riprende tutte le osservazioni illustrate in fase di analisi di letteratura e in fase sperimentale e propone dei punti ai quali dare priorità in ottica di sviluppi futuri.

Capitolo Primo

ANALISI DELL'AMBITO DI RIFERIMENTO

1 FORMATI, STANDARD, MODELLI DATI

1.1 Definizione di Software

In ambito informatico, software viene definito come *l'insieme dei programmi che possono essere impiegati su un sistema di elaborazione dei dati* (Vocabolario Treccani, 2026). Il software contiene le istruzioni che il computer deve seguire affinché il programma venga eseguito correttamente.

Un software BIM è uno strumento che consente di creare una rappresentazione digitale 3D di un edificio, la quale viene integrata con dati fisici e funzionali. Questo tipo di software è parametrico, ovvero modificando il modello si aggiornano automaticamente tutte le viste, mantenendo le relazioni logiche impostate.

Un software FEM è uno strumento di simulazione utilizzato da ingegneri strutturalisti per eseguire l'analisi a elementi finiti, con l'obiettivo di risolvere sistemi complessi. Il metodo si applica scomponendo i corpi fisici in molteplici elementi di forma definita e di dimensioni contenute. I dati di *input* che vengono inseriti nel programma all'inizio dell'analisi FEM riguardano la caratterizzazione meccanica dei materiali, la scelta dei vincoli e dei carichi e la struttura della mesh.

1.2 Definizione di File

Un *file* è una sequenza di dati organizzati secondo schemi dove risiedono le informazioni che devono essere conservate in modo persistente, tipicamente in un disco o in una differente unità di I/O. Ogni file ha un nome e un'estensione che ne indica il tipo (es. .docx, .jpg, .mp4). I file possono essere di vari formati, ognuno progettato per uno scopo specifico, come ad esempio i documenti di testo, le immagini e i file audio. I file hanno la funzione di raccogliere e archiviare informazioni affinché queste possano essere facilmente recuperate e utilizzate quando necessario, rendendo più semplice la gestione dei dati (Matteo Cristani, 2002).

1.3 Formati proprietari e formati aperti

L'evolversi delle tecnologie e la crescente disponibilità e complessità dell'informazione digitale ha indotto la necessità di gestire sempre maggiori forme di informazione digitale (testo, immagini, filmati, ecc.) e di disporre di funzionalità più specializzate per renderne più facile la

creazione, la modifica e la manipolazione. Questo fenomeno porta all'aumento del numero dei formati disponibili e dei corrispondenti programmi necessari a gestirli nonché delle piattaforme su cui questi operano (Ministero Della Giustizia, 2014).

Il formato di un file è la convenzione usata per interpretare, leggere e modificare il file. I formati si differenziano per il modo in cui ordinano e interpretano i bit, ovvero per come codificano e decodificano i dati binari che costituiscono il documento. Queste operazioni si svolgono grazie ad algoritmi matematici che il computer applica per leggere i dati contenuti nel file (Ministero Della Giustizia, 2014).

1.3.1 *Cos'è un Formato Proprietario*

Un formato è considerato proprietario quando la sua struttura interna di rappresentazione dei dati non è trasparente e le sue specifiche non sono liberamente accessibili al pubblico. Generalmente, questi formati vengono creati da aziende di software per codificare i dati specifici delle loro applicazioni. Di conseguenza, solo i prodotti sviluppati da tali aziende sono in grado di interpretare in modo completo e corretto le informazioni contenute in un file con questo tipo di formato. Inoltre, i formati proprietari possono essere tutelati da un brevetto e possono richiedere il pagamento di royalty a chiunque desideri utilizzarli.

1.3.2 *Definizione di Formato Aperto*

Un formato si dice "aperto" quando è conforme a specifiche pubbliche, cioè disponibili a chiunque abbia interesse ad utilizzare quel formato. La disponibilità delle specifiche del formato rende sempre possibile la decodifica dei documenti rappresentati in conformità con dette specifiche, anche in assenza di prodotti che effettuino tale operazione automaticamente. Questa condizione si verifica sia quando il formato è documentato e pubblicato da un produttore o da un consorzio al fine di promuoverne l'adozione, sia quando il documento è conforme a formati definiti da organismi di standardizzazione riconosciuti. In quest'ultimo caso, tuttavia, si confida che quest'ultimi garantiscono l'adeguatezza e la completezza delle specifiche stesse (Ministero Della Giustizia, 2014). Si riporta di seguito uno schema illustrativo rappresentante i rapporti logici tra le entità informatiche descritte in questo capitolo.

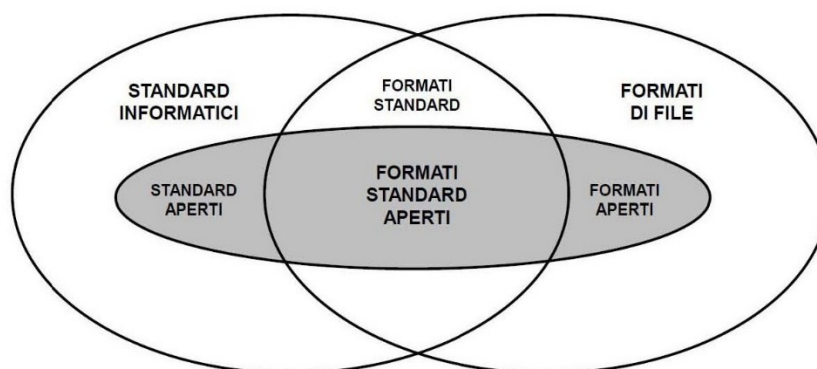


Figura 1 – Rapporto tra Standard e Formato. Fonte: (Simone Aliprandi, 2014)

Il focus di questa ricerca tesi è l'interoperabilità, ovvero lo scambio di dati. A questo proposito si possono valutare due modalità con cui è possibile realizzare il passaggio di informazioni: attraverso formati proprietari oppure con formati aperti.

Utilizzare formati proprietari vincola gli utenti all'uso di specifici software commerciali e in questo modo non solo viene limitata la flessibilità, ma viene anche introdotto il rischio di obsolescenza del formato, rendendo il file potenzialmente illeggibile nel tempo per il destinatario. Al contrario, i formati aperti promuovono l'accessibilità e la trasparenza, assicurando che i file possano essere letti e modificati universalmente e risultando la soluzione ideale per favorire l'interoperabilità tra sistemi. Inoltre, l'adozione di formati aperti svincola gli utenti dall'obbligo di utilizzare determinati software, promuovendo un mercato più competitivo ed equo, in cui la scelta degli strumenti progettuali non è dettata da restrizioni tecnologiche.

1.4 Definizione di Standard

Gli Standard sono l'insieme di regole e norme che definiscono come un formato vada strutturato, con lo scopo di garantire compatibilità e uniformità in un ecosistema digitale avente numerosi software. Grazie agli Standard, infatti, viene indicata quale struttura del formato sia più congeniale per ciascuna categoria di software (S. Aliprandi, 2010).

Gli Standard possono essere approvati da enti ufficiali, e dunque essere definiti *Standard de jure*, oppure possono derivare da convenzioni adottate nel tempo a livello commerciale, senza tuttavia essere approvate ufficialmente. Questi si definiscono *Standard de facto* (S. Aliprandi, 2010).

1.5 Definizione di Modello Dati

Un modello dati rappresenta una formalizzazione del modo in cui sono organizzati i dati, interconnessi all'interno di un sistema informativo, facilitandone la gestione e l'elaborazione.

La modellazione dei dati produce una rappresentazione concettuale, basata su principi di logica formale, che stabilisce le relazioni e le dipendenze tra le entità e i loro attributi. La coerenza e l'accessibilità delle informazioni, garantite da un modello dati ben definito, facilitano la manipolazione dei dati anche in ottica di confronto tra modelli aventi struttura e funzionamento diverso.

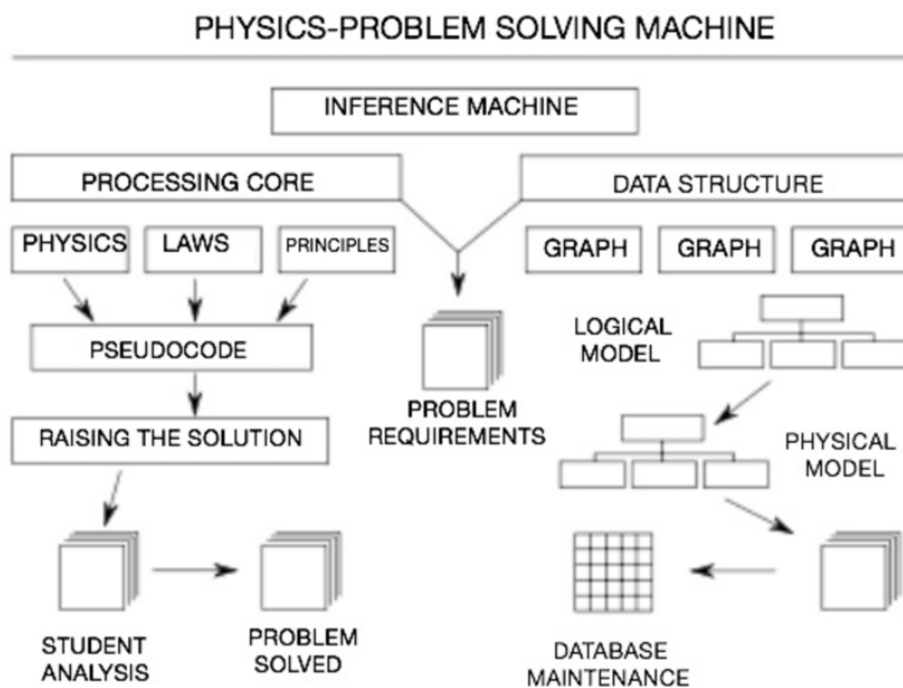


Figura 2 - Schema di un modello dati. Fonte: (Elena Cerro Prada & Luis B. López Vázquez, 2015)

1.5.1 Componenti

I modelli dati sono costituiti da elementi di dati, i quali insieme determinano le modalità di archiviazione e manipolazione delle informazioni. Di seguito un breve accenno a queste componenti (P. Atzeni et al., 2014):

- Entità: sono gli elementi principali della storia dei dati e vengono rappresentati sotto forma di tabelle in un database relazionale;
- Attributi: proprietà che descrivono le entità e vengono rappresentati come colonne di una tabella;
- Relazioni: sono le componenti che identificano come le entità sono collegate tra loro.
- Chiavi: sono elementi utilizzati per identificare in modo univoco una riga di una tabella.
- Vincoli: sono regole applicate ai dati con lo scopo di mantenerne la coerenza e l'accuratezza.

- Schema: descrizione logica dell'intero database. Definisce la struttura, i vincoli e le relazioni del database.

1.5.2 Livelli di modellazione dei dati

I dati sono organizzati secondo tre tipologie di modelli dati: il modello concettuale, il modello logico e il modello fisico. Questi livelli di modellazione hanno gradi differenti di astrazione e permettono di ridurre la complessità del modello dati scomponendolo in tre sezioni (P. Atzeni et al., 2014).

Il modello concettuale è una mappa astratta che serve ad individuare quali entità sono presenti all'interno del file, senza approfondire nei dettagli, con lo scopo di avere chiaro il quadro generale delle informazioni.

Il modello logico, invece, entra nel vivo delle specifiche proprie di ciascuna componente del file. Questo modello stabilisce le relazioni che intercorrono tra entità e attributi ed indica nel dettaglio la struttura con cui sono organizzati i dati.

Il modello fisico, infine, definisce dove memorizzare le informazioni. Questa fase prevede la creazione di un database in cui i dati verranno archiviati e gestiti (P. Atzeni et al., 2014).

L'immagine che segue è un esempio di organizzazione dei livelli di un modello dati, in cui i requisiti rappresentano le entità del file.

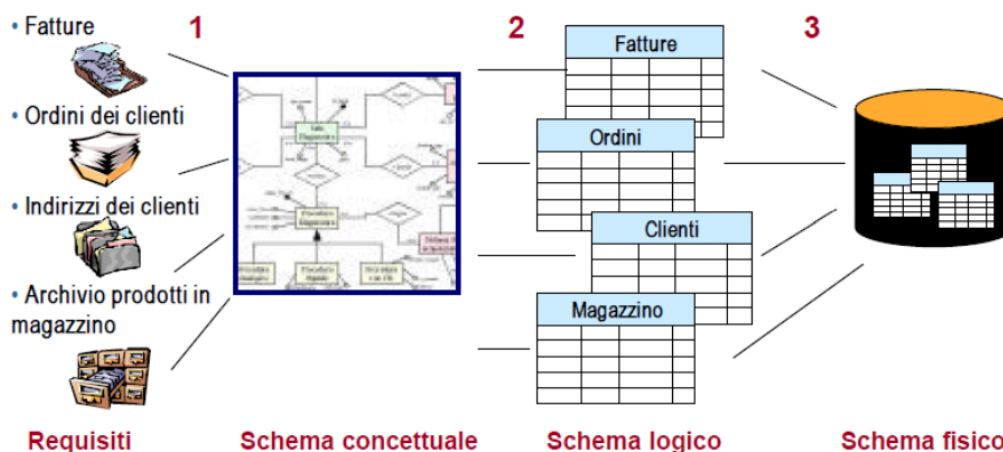


Figura 3 - Livelli di modellazione dei dati. Fonte: (Ing. Daniele Corti, 2013)

2 BIM PER LA STRUCTURAL ANALYSIS

2.1 Definizione di BIM

Il National Institutes of Building Science di Washington definisce il B.I.M. – acronimo anglofono di Building Information Model/Modeling – come: “la rappresentazione digitale di caratteristiche fisiche e funzionali di un oggetto”. Ossia una metodologia di lavoro che agevola in maniera significativa lo scambio di dati e informazioni tra i vari progettisti (E. Kennett, 2007).

È bene sottolineare che il BIM non è un prodotto o un software, come in molti sostengono, ma è un processo lavorativo innovativo, una metodologia basata su un modello dell’edificio che contiene ogni informazione relativa alla costruzione.

Come riportato ne Il futuro del disegno con il BIM per ingegneri e architetti: “Il BIM è dunque costituito dall’insieme dei processi applicati per realizzare, gestire, ricavare e comunicare informazioni tra soggetti a livelli differenti, utilizzando dei modelli creati da tutti i partecipanti al processo edilizio, in tempi diversi ed anche per scopi non uguali tra di loro, per garantire qualità ed efficienza attraverso l’intero ciclo di vita di un manufatto.” (Anna Osello, 2012).

Per tanto, il BIM mette in pratica processi di programmazione e progettazione per ottimizzare tutte le fasi dalla realizzazione alla gestione, demolizione e dismissione di un edificio, raccogliendo digitalmente tutte le informazioni relative al suo intero ciclo di vita. In altre parole, potrebbe essere considerato come un enorme database informativo impostato su un modello digitale tridimensionale. Parte integrante all’interno di tutti i processi BIM Based è la collaborazione, meglio nota come interoperabilità, tra le diverse figure professionali coinvolte nel processo edilizio. Questa peculiarità favorisce una modellazione integrata e multidisciplinare in quanto permette in qualsiasi momento a tutti gli attori coinvolti di consultare, modificare e scambiare le informazioni sul e del modello BIM (Michele Zucco, 2022).



Figura 4 - Rappresentazione del concetto di BIM. Fonte: (Michele Zucco, 2022)

Riferendosi alla normativa italiana UNI 11337-6, il BIM è strutturato secondo sette dimensioni: 1D è il concept design, che coinvolge l'organizzazione degli elementi; 2D sono gli elaborati come piante, prospetti e sezioni; 3D individua la rappresentazione tridimensionale del prodotto; 4D equivale all'analisi della durata o tempi (programmazione); 5D riguarda l'analisi dei costi (computi, stime e valutazioni economiche); 6D coinvolge la fase di gestione dell'opera (uso, manutenzione e dismissione); 7D è relativa alla valutazione della sostenibilità (sociale, economica e ambientale) (UNI, 2017).

Vi sono poi tre dimensioni aggiunte successivamente: 8D riguarda la sicurezza in fase di progettazione ed esecuzione dei lavori; 9D riguarda le informazioni da assegnare al modello per ottimizzare la realizzazione dell'opera e gestire in maniera efficiente le risorse; 10D coinvolge i dati che integrano nel modello nuove tecnologie, dati fisici, commerciali o ambientali attraverso la digitalizzazione con lo scopo di industrializzare il mondo delle costruzioni, incrementando la produttività.



Figura 5 - Le dimensioni del BIM. Fonte: (Gabriel Fagundes Lobato de Araújo, 2018)

2.2 LOD

Nel settore dell'architettura e dell'ingegneria i livelli di dettaglio, i cosiddetti LOD (Level Of Detail) definiscono con precisione il livello di approfondimento delle informazioni contenute all'interno del modello BIM. Rappresentano quindi un punto di riferimento definito dal committente che permette a tutti i soggetti coinvolti di costruire il modello BIM con lo stesso grado di definizione dei contenuti lungo tutte le fasi in cui si articola il processo di concepimento e realizzazione dell'opera.

Nella normativa italiana UNI 11337-4:2017, si distinguono i livelli di dettaglio tra LOD, LOG e LOI: i LOD sono intesi come i livelli di sviluppo degli oggetti digitali e sono composti da LOG e LOI. I LOG rappresentano gli attributi geometrici degli oggetti, mentre i LOI sono gli attributi informativi degli oggetti. In normativa, i LOD vengono categorizzati in sette tipologie, ciascuna identificata da una lettera (UNI, 2017):

- 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.

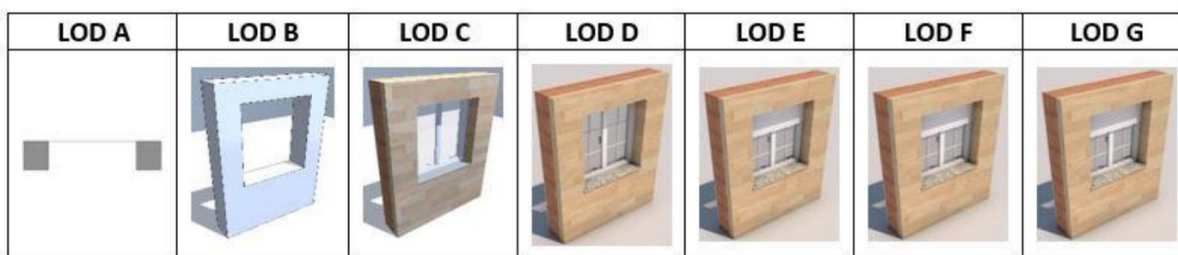


Figura 6 - LOD. Fonte: (Gabriel Fagundes Lobato de Araújo, 2018)

2.3 Vantaggi del BIM

In generale, l'adozione di un metodo di modellazione BIM nel processo progettuale ed esecutivo/gestionale dell'opera vede diversi vantaggi per i professionisti. In breve, è possibile affermare che il BIM faciliti la collaborazione tra entità diverse, riducendo di conseguenza la possibilità di compiere errori. Il flusso automatizzato, inoltre, permette di ottimizzare le risorse, ponendo il progetto in maniera molto competitiva sul mercato.

2.3.1 Collaborazione tra professionisti

Come anticipato in precedenza, nella progettazione e costruzione di un'opera edilizia concorrono molteplici figure, le quali hanno competenze diverse e abitualmente fanno uso di strumenti di modellazione tra i più svariati.

Negli ultimi anni con lo sviluppo dei software e delle nuove applicazioni all'interno del mondo delle costruzioni si sta cercando di andare verso un progetto altamente integrato che riesca a risolvere molte delle criticità attualmente presenti in fase di progettazione. In questo scenario si inseriscono i programmi BIM, che sono la base per l'ottimizzazione del processo edilizio. Uno dei principali obiettivi del BIM è quello di far collaborare tutte le figure coinvolte in una sola piattaforma di condivisione, eliminando quindi la replicabilità del dato. È evidente come il modello BIM si ponga al centro del processo fungendo da piattaforma di condivisione e di regolazione del flusso di informazioni. Tramite il BIM è possibile ottenere un processo interoperabile dove in tempo reale tutti i professionisti possono vedere quali variazioni subisce il progetto (Kejdi Luku, 2023).

2.3.2 Controllo delle interferenze e riduzione degli errori

La digitalizzazione del modello consente la sovrapposizione di diversi progetti appartenenti allo stesso edificio, come ad esempio quelli architettonico, strutturale e impiantistico, i quali possono talvolta essere collegati e visualizzati contemporaneamente. In questo modo risultano evidenti fin da subito eventuali errori progettuali o interferenze tra i modelli, che quindi possono essere corretti per rendere i progetti compatibili. Essere in grado di avere dei progetti completi e dettagliati consente di arrivare alla fase esecutiva con la certezza che non si incorrerà in imprevisti, i quali, se corretti in corso d'opera, risultano decisamente più onerosi.

Inoltre, i software BIM sono parametrici, per tanto, quando si effettua una modifica, tutte le viste vengono aggiornate automaticamente. In questa maniera ci si assicura che gli elaborati siano coerenti tra di loro, evitando il rischio di dimenticare di estendere le modifiche a tutte le tavole di progetto.

2.3.3 Ottimizzazione costi, tempi e risorse

Il BIM permette ai progettisti di ridurre i tempi di redazione dei progetti, di limitare gli errori, di rispettare le previsioni stimate con la conseguente riduzione di costi, tempi e risorse. Il flusso di lavoro viene ottimizzato e questo aumenta il rendimento: più si lavora velocemente e prima si ottiene un ritorno di capitale.

Si stima una riduzione dei tempi di realizzazione di progetti BIM fino al -27% rispetto a quelli tradizionali. Il risparmio di tempo si traduce, poi, in risparmio economico. L'impiego del

BIM ha dimostrato un abbattimento dei costi totali di progetto compreso tra il 5 e il 20% del costo totale: questo principalmente evitando le varianti in corso d'opera (Kejdi Luku, 2023).

2.4 Formati Aperti per il BIM: lo Standard IFC

I formati aperti per il BIM sono alla base del concetto di OpenBIM e permettono di scambiare i dati tra software diversi senza perdere informazioni. Il più diffuso e maturo è IFC, ma ne esistono anche altri come ad esempio SAF, BCF, COBie, gbXML.

Con il termine OpenBIM si intende un approccio universale per la collaborazione durante le fasi di progettazione, realizzazione e messa in esercizio degli edifici basato su standard e flussi di lavoro aperti. Lo standard utilizzato principalmente in OpenBIM è l'IFC (Chuck Eastman, 2016).

Il formato file IFC, acronimo di "Industry Foundation Classes", è un modello di dati aperto standardizzato. Creato e gestito dall'organizzazione internazionale buildingSmart, è il formato di scambio dati preferenziale per tutto il mondo della progettazione BIM oriented e per lo scambio di dati nell'industria delle costruzioni (C. Eastman et al., 2016).

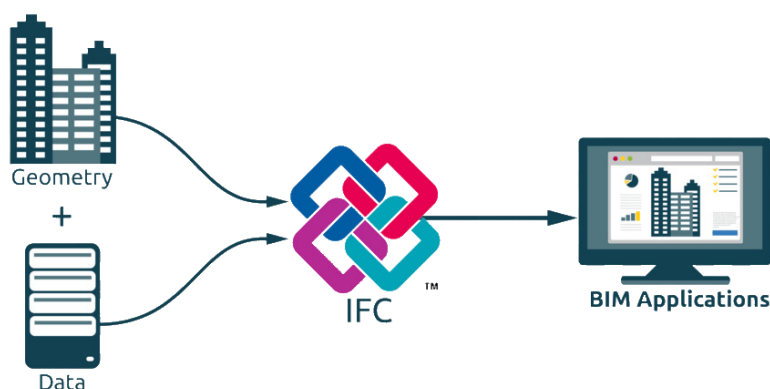


Figura 7 - Schema concettuale dell'utilizzo di IFC. Fonte: (Inojosa et al., 2023)

2.4.1 Definizione di IFC

In generale, IFC, o "Industry Foundation Class", è una descrizione digitale standardizzata dell'ambiente costruito, inclusi edifici e infrastrutture civili. La specifica dello schema IFC è il principale prodotto tecnico di buildingSMART International. Più specificamente, lo schema IFC è un modello di dati standardizzato che codifica, in modo logico:

- l'identità e la semantica (nome, identificatore univoco leggibile meccanicamente, tipo di oggetto o funzione);
- le caratteristiche o gli attributi (come materiale, colore e proprietà termiche);
- le relazioni (inclusi luoghi, connessioni e proprietà) tra:
 - oggetti (come colonne o lastre)
 - concetti astratti (performance, costing)

- processi (installazione, operazioni)
- persone (proprietari, designer, appaltatori, fornitori, ecc.).

La specifica dello schema può descrivere come viene utilizzata una struttura o installazione, come è costruita e come viene gestita. IFC è in grado di definire i componenti fisici di edifici, prodotti prefabbricati, sistemi meccanici / elettrici, nonché i modelli più astratti per l'analisi strutturale, per l'analisi energetica, per la suddivisioni dei costi, per la programmazione dei lavori e molto altro ancora (buildingSMART International, 2026).

2.4.2 Versioni IFC

IFC è un formato aperto in continua evoluzione e per tanto negli anni ha subito modifiche e aggiornamenti, che si sono tradotti in nuove versioni dello standard. Le versioni attualmente in uso sono IFC2X3, IFC4 e, la più recente, IFC4X3. Le versioni precedenti, come IFC2X2, sono supportate dai software, ma raramente utilizzate.

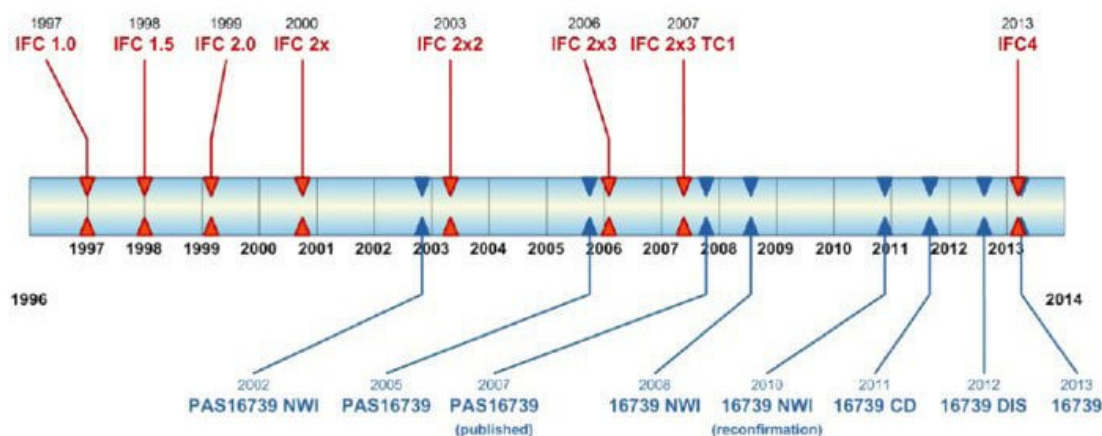


Figura 8 - Evoluzione temporale delle versioni di IFC. Fonte: (Giovanna Acampa et al., 2018)

IFC2X3 è una versione ampiamente consolidata e ancora in uso, tuttavia risulta superata. Infatti, nel 2013 è stato rilasciato IFC4 (anche detto IFC2X4), il quale è un'evoluzione di IFC2x3. Nato con lo scopo di migliorare le funzionalità della versione precedente, ha introdotto importanti rielaborazioni delle specifiche IFC e ha migliorato la coerenza delle esportazioni.

IFC4X3, rilasciato nel 2024, e ulteriormente rielaborato in IFC4X3_ADD2 (senza modifiche allo schema e alla specifica), risulta essere la versione in assoluto più recente di IFC. La sua evoluzione, tuttavia, si è concentrata sul rendere lo Standard accessibile anche al mondo delle infrastrutture e quindi le modifiche che sono state apportate si focalizzano sull'introduzione di nuove classi specifiche per strade, ferrovie, ponti e opere idrauliche, nonché sullo sviluppo dell'interoperabilità BIM-GIS.

2.4.3 IFC 4X3_ADD2

Con la versione IFC4x3_ADD2 viene dato un notevole slancio alla digitalizzazione delle infrastrutture in formato aperto, estendendo i benefici IFC alle infrastrutture che si sviluppano attraverso il territorio come strade, ferrovie, porti e vie navigabili, ponti e gli elementi comuni tra tutti questi.

Per identificare quali dati possono essere scambiati attraverso l'esportazione del modello in IFC4X3_ADD2, bisogna fare riferimento alle sue Model View Definition specifiche. Le Model View Definitions (MVD) sono sottoinsiemi (o filtri intelligenti) dello schema IFC che definiscono quali parti del modello di dati dovrebbero essere incluse in uno scambio di informazioni (Han Liua et al., 2019).

A differenza della policy per IFC4 che prevedeva 2 MVD, quella per IFC4X3_ADD2 prevede attualmente 3 livelli di implementazione per IFC: *Reference View*, *Alignment Based View* e *Design Transfer View*. Si tratta di livelli graduali che aggiungono funzionalità più avanzate alle implementazioni.

La *Reference View* ha lo scopo di supportare il coordinamento delle discipline di progettazione architettonica, dell'analisi strutturale e dei servizi di costruzione. Supporta in particolare la verifica dei conflitti e la risoluzione di problemi derivanti da errori nella modellazione geometrica.

La *Design Transfer View* è stata progettata per supportare il trasferimento dei dati del modello utili alle ulteriori attività di progettazione, analisi, stima e gestione delle strutture. Questa vista è più dettagliata.

La vista *Alignment Based View*, introdotta con questa versione e non presente in IFC4, si concentra sul posizionamento degli elementi basato su tracciati lineari, con lo scopo di facilitare uno scambio di dati che richieda una precisa definizione geometrica del tracciato (orizzontale, verticale, inclinato) e il posizionamento lineare relativo degli elementi (es. posizionare pali, barriere o conci di un ponte lungo un asse) (buildingSMART International, 2026).

2.4.4 Modello dati di IFC4X3_ADD2

I dati contenuti in IFC4X3_ADD2 si organizzano secondo uno schema standardizzato ben preciso e basato su un linguaggio e un'architettura definiti direttamente dal sito ufficiale di buildingSMART come segue.

La specifica IFC include termini, concetti ed elementi dei dati che derivano dall'uso in discipline, settori e professioni del settore delle costruzioni e del facility management. Termini e concetti sono espressi in maniera semplice, mentre gli elementi all'interno della specifica dei dati seguono una convenzione di denominazione:

- i nomi degli elementi dati per tipi, entità, regole e funzioni iniziano con il prefisso "Ifc" e continuano con le parole inglesi secondo la convenzione di denominazione CamelCase (nessun trattino basso, prima lettera della parola in maiuscolo);
- i nomi degli attributi all'interno di un'entità seguono la convenzione di denominazione CamelCase senza prefisso;
- le definizioni dei set di proprietà che fanno parte di questo standard iniziano con il prefisso "Pset_" e continuano con le parole inglesi nella convenzione di denominazione CamelCase;
- le definizioni degli insiemi di quantità che fanno parte di questo standard iniziano con il prefisso "Qto_" e continuano con le parole inglesi nella convenzione di denominazione CamelCase.

L'architettura dello schema dati di IFC definisce quattro livelli concettuali, a ogni singolo schema è assegnato esattamente un livello concettuale.

1. **Resource layer:** il livello più basso include tutti gli schemi individuali contenenti definizioni di risorse; tali definizioni non includono un identificatore univoco globale e non devono essere utilizzate indipendentemente da una definizione dichiarata a un livello superiore;
2. **Core layer:** il livello successivo include lo schema del kernel e gli schemi di estensione del core, contenenti le definizioni di entità più generali; tutte le entità definite al livello core o superiore hanno un ID univoco globale e, facoltativamente, informazioni sul proprietario e sulla cronologia;
3. **Interoperability layer:** il livello successivo include schemi contenenti definizioni di entità specifiche per un prodotto generale, un processo o una specializzazione di risorse utilizzata in diverse discipline; tali definizioni sono in genere utilizzate per lo scambio e la condivisione tra domini di informazioni di costruzione;
4. **Domain layer:** il livello più alto include schemi contenenti definizioni di entità che rappresentano specializzazioni di prodotti, processi o risorse specifiche di una determinata disciplina; tali definizioni sono in genere utilizzate per lo scambio e la condivisione di informazioni all'interno del dominio. In questa immagine, il dominio "tunnel" è trasparente, poiché non fa ancora parte dell'attuale IFC 4.3, ma è previsto che venga incluso in un futuro aggiornamento. I domain layer di IFC4x3_ADD2 si differenziano dal precedente IFC4 per l'aggiunta di 4 livelli: Rail Domain, Road Domain, Ports and Waterways Domain, Tunnel Domain.

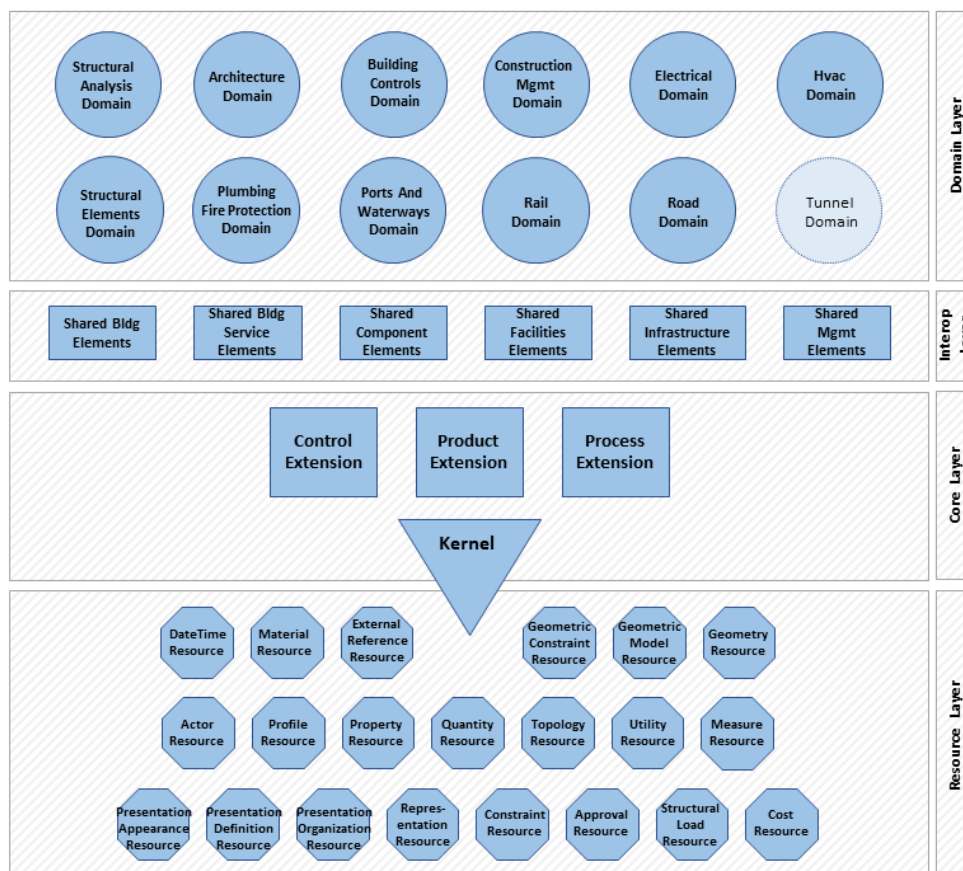


Figura 9 - Architettura dello schema dei dati di IFC4X3_ADD2. Fonte: (buildingSMART International, 2026)

Come si può notare dallo schema grafico, i diversi livelli sono organizzati secondo una struttura gerarchica in cui le entità presenti ad un livello superiore fanno riferimento a quelle contenute nei livelli inferiori. Questa gerarchia consente di correlare in maniera chiara le informazioni, favorendo una facile consultazione dei dati.

2.5 IFC Structural Domain

IFC Structural Domain è uno schema che si trova all'interno del *domain layer* di IFC, tra tutti gli schemi di dati specifici del dominio (Vedi Architettura dello schema dei dati). Questo schema definisce gli elementi utili all'analisi FEM e dunque facilita l'interoperabilità tra modelli BIM e software di analisi a elementi finiti. Nella sezione che segue si approfondirà il suo modello dati.

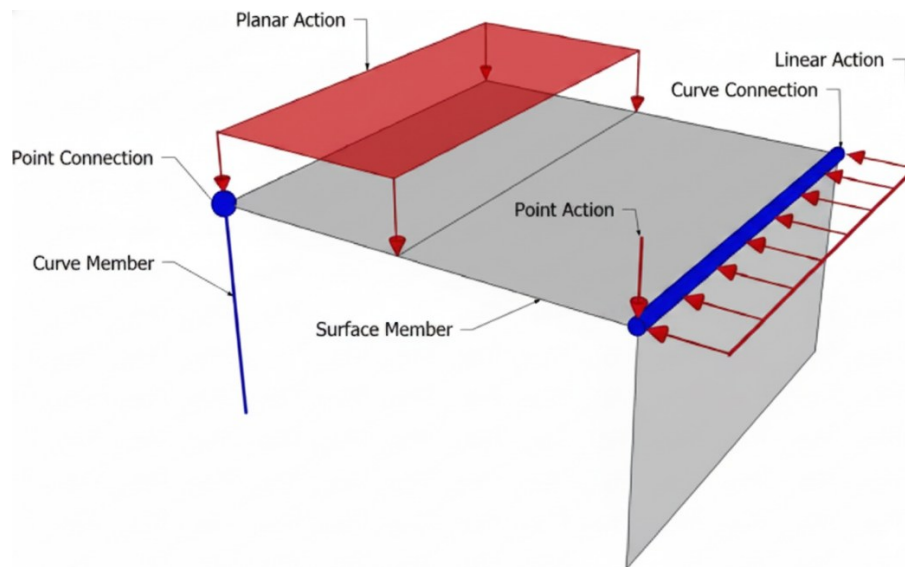


Figura 10 - Utilizzo dell'IFC Structural Domain nella modellazione 3D. Fonte: (Raimar Scherer & Fangzheng Lin, 2020)

2.5.1 Ambito d'interesse di IFC Structural Domain

Questo dominio descrive il modello di analisi strutturale. Utilizza gli elementi e la struttura spaziale del modello fisico e gli associa le caratteristiche analitiche, con lo scopo di mostrare le informazioni utili all'analisi strutturale. Gli elementi presenti al suo interno sono:

- Curve, superfici
- Connessioni tra punti, curve e superfici
- Carichi su punti, curve, superfici. Carichi termici. Combinazioni di carico.
- Risultati delle analisi: forze e spostamenti.

Non sono presenti:

- Analisi dinamiche
- Carichi di precompressione e pretensione
- Proprietà geometriche e relazioni spaziali degli elementi finiti ("finite element topology")
- Risultati dettagliati di mesh agli elementi finiti

2.5.2 Modello dati di IFC Structural Domain

Il modello dati di IFC Structural Domain è organizzato concettualmente in quattro classi: IfcStructuralItem, IfcStructuralActivity, IfcStructuralAnalysisModel, Classe di relazioni. Ciascuna classe contiene types, entities, attributes, instances, relationships, che sono strutturate gerarchicamente in tre categorie: Types, Entities e Property Sets. Di seguito un accenno sulle

categorie che descrivono il modello dati di IFC Structural Domain, per maggiori informazioni si rimanda al sito di Building Smart.

I *types* sono delle categorie gerarchiche che indicano caratteristiche comuni a molti elementi. I *types* hanno una struttura a vettore, enumerativa, contenente i *type values*.

Le *entities* sono definizioni di oggetti standardizzati nel modello di dati IFC e rappresentano un elemento, una proprietà, una relazione o un concetto del progetto edilizio. Le *entities* al loro interno contengono *attributi* specifici che le definiscono e esternamente sono collegate tra loro tramite relazioni standardizzate, che definiscono come gli oggetti interagiscono tra loro nel modello.

IfcPropertySet è l'insieme di tutte le proprietà che si possono assegnare ad un oggetto di un modello 3D in formato IFC. Le proprietà contenute in tale insieme (o *set*) possono essere di due tipi: predefinite o personalizzate/addizionali (cioè definite dall'utente) (G. Marcolin, 2017).

3 FEM PER LA STRUCTURAL ANALYSIS

3.1 FEM (*Finite Element Method*)

Il metodo degli elementi finiti (FEM, dall'inglese Finite Element Method) è una tecnica numerica atta a cercare soluzioni approssimate di problemi descritti da equazioni differenziali alle derivate parziali riducendo queste ultime a un sistema di equazioni algebriche.

Il metodo degli elementi finiti prevede la discretizzazione delle strutture attraverso la creazione di una griglia (mesh) composta da primitive (elementi finiti) di forma codificata (triangoli e quadrilateri per domini 2D, tetraedri ed esaedri per domini 3D) (J. Chaskalovic, 2008).

L'analisi FEM prevede l'utilizzo di software specifici capaci di calcolare sistemi matematici complessi: per questo tipo di analisi un oggetto viene suddiviso in un elevato numero di elementi finiti, interconnessi da punti di contatto (nodi). Attraverso operazioni matriciali che analizzano le sollecitazioni nei punti è possibile prevedere il comportamento di ogni singolo elemento in cui l'oggetto viene suddiviso. Combinando la risposta dei singoli elementi, in base ai vincoli e alle sollecitazioni che agiscono sui nodi, l'elaboratore è poi in grado di fornire la risposta dell'oggetto nella sua interezza. Questo processo permette di conoscere in maniera preventiva il comportamento meccanico dell'edificio, verificando e aggiornando la sua progettazione ingegneristica.

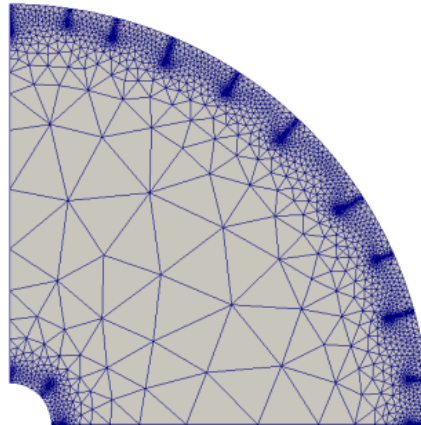


Figura 11 - Mesh frutto della discretizzazione di un oggetto. Fonte: (N. Ferro et al., 2017)

3.2 Nascita e diffusione del FEM

Il metodo degli elementi finiti trova origini nelle necessità di risoluzione strutture nell'ambito dell'ingegneria civile ed aeronautica. Nato attorno agli anni 1930-35, l'idea si sviluppa in maniera notevole nella seconda metà degli anni 1950, con il contributo fondamentale di M. J. Turner della Boeing. Egli formulò e perfezionò il Direct Stiffness Method, il primo approccio agli elementi finiti nel campo del continuo. Il lavoro di Turner trovò diffusione fuori dagli stretti ambiti dell'ingegneria aerospaziale, diffondendosi in breve in tutto il mondo.

A partire dagli anni 1970, il FEM ha trovato diffusione come strategia di modellazione numerica di sistemi fisici in un'ampia varietà di discipline ingegneristiche, per esempio elettromagnetismo, fluidodinamica, calcolo strutturale e geotecnica. Sempre in quel periodo nacquero gran parte dei codici di analisi FEM commerciali (NASTRAN, ADINA, ANSYS, ABAQUS, SAMCEF, MESHPARTS, etc) tuttora disponibili.

3.3 Software FEM

Nell'ambito della progettazione i professionisti si avvicinano all'analisi FEM attraverso software specifici che velocizzano il processo, riducono gli errori al minimo e hanno capacità di calcolo molto potenti, consentendo di confrontarsi con strutture estremamente complesse. Il funzionamento di questi programmi si può schematizzare in tre fasi: modellazione, calcolo, visualizzazione.

La modellazione è il procedimento con cui l'utente definisce la geometria, le caratteristiche meccaniche, i vincoli e i carichi che definiscono la struttura. La geometria può essere importata con un file esterno oppure può essere modellata direttamente dal software FEM. Successivamente l'ingegnere deve operare attraverso il processo di meshatura, ovvero deve suddividerla in elementi. Inoltre, l'utente deve definire i materiali che compongono la

struttura, impostandone i parametri meccanici. Infine, il fruitore del software deve attribuire i carichi e i vincoli alla struttura, posizionandoli nei nodi e sugli elementi soggetti a forze esterne.

La fase di calcolo è automatizzata: il computer, a seguito dell'input di esecuzione, è in grado di risolvere un sistema complesso di equazioni differenziali basate sulla mesh, generando un risultato per ciascuno degli elementi.

Infine, il tecnico che utilizza il software è in grado di visualizzare e interpretare i risultati ottenuti dal calcolo, che possono essere tradotti in tabelle, grafici a colori o deformazioni. Da questi dati l'ingegnere è in grado di definire come la struttura reagisca alle azioni simulate digitalmente, potendo così impostare in sicurezza i dettagli progettuali della struttura calcolata.

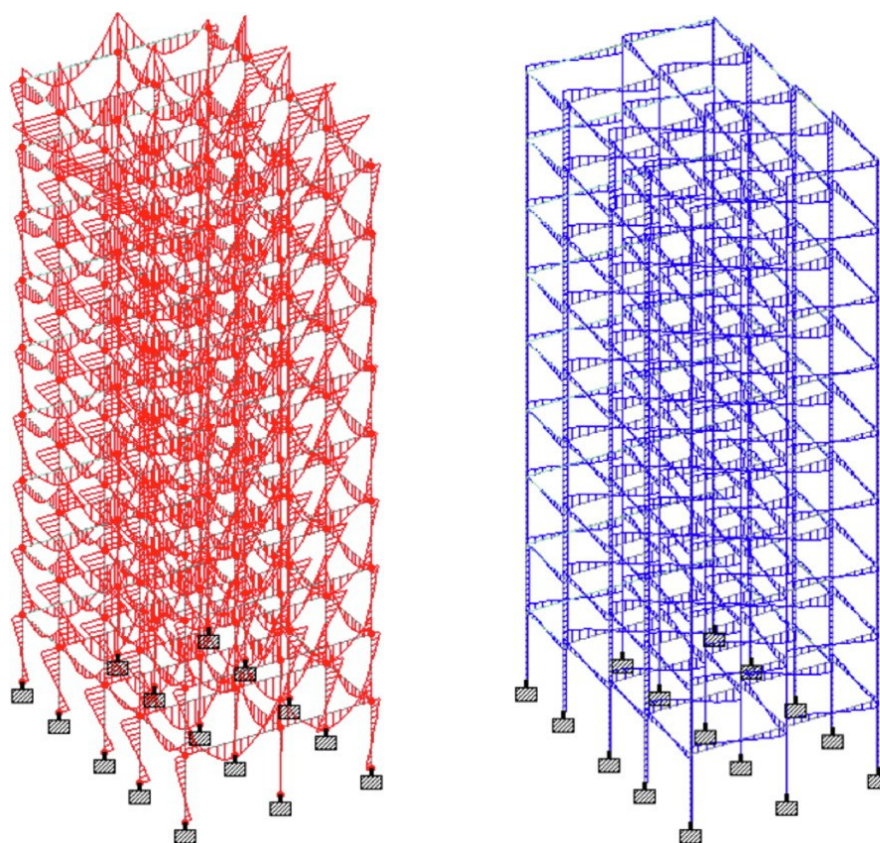


Figura 12 - Esempio di analisi FEM. Fonte: (K. Surender Kumar, 2020)

3.4 Formati per il FEM

Nell'ambito FEM esistono sia formati aperti che formati proprietari e, per questa ricerca tesi, abbiamo voluto scegliere un formato proprietario che fosse compatibile con un buon numero di software FEM: NASTRAN. Si riporta di seguito la lista dei software per analisi a elementi finiti esistenti in commercio, sottolineando che a ciascuno dei software definiti "commerciali" corrisponde un formato proprietario:

Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM IFC e NASTRAN
 Alice Pin

Tabella 1 – Elenco software FEM.

ELENCO DEI SOFTWARE A ELEMENTI FINITI	
OPENSOURCE	COMMERCIALI
CalculiX, (Per Windows)	ABAQUS [collegamento interrotto], su <i>simulia.com</i> .
CAElinux, su <i>caelinux.com</i> .	Altair RADIOSS, su <i>altair.com</i> .
Aster, su <i>code-aster.org</i> .	Altair OptiStruct, su <i>altair.com</i> .
DEAL.II, su <i>dealii.org</i> .	ADINA R&D, Inc., su <i>adina.com</i> .
DUNE (Distributed and Unified Numerics Environment)	ALGOR Incorporated
Elmer FEM solver, su <i>research.csc.fi</i> .	ANSYS, su <i>ansys.com</i> .
FEATool, su <i>featool.com</i> .	Autodesk Nastran
FEniCS Project, su <i>fenics.org</i> .	SIMULIA Abaqus, su <i>simulia.com</i> .
FELit, su <i>felt.sourceforge.net</i> .	CASTEM CAST3M, su <i>www-cast3m.cea.fr</i> .
freeFEM, su <i>freefem.org</i> .	COMSOL Multiphysics, su <i>comsol.com</i> .
GetDP, su <i>geuz.org</i> .	COSMOS CosmosWorks, su <i>cosmosm.com</i> .
Impact, su <i>impact.sourceforge.net</i> .	EDISIS, su <i>newsoft-eng.it</i> .
OOFEM, su <i>oofem.org</i> .	Esi, su <i>esi-italia.it</i> .
OpenSees (Fem per problemi sismici dell'università di Berkeley)	EuroPlexus, su <i>europlexus.jrc.ec.europa.eu</i> .
PETScFEM, su <i>cimec.org.ar</i> .	Femap, su <i>plm.sw.siemens.com</i> .
Salome, su <i>salome-platform.org</i> .	GTSTRUDL30, su <i>gtstrudl.gatech.edu</i> .
SLFFEA, su <i>slffea.sourceforge.net</i> .	Go-Mesh, su <i>we.go-mesh.it</i> .
Sundance, su <i>software.sandia.gov</i> .	JMAG, su <i>advancedmotortech.com</i> .
Z88_FEM_Software, su <i>z88.org</i> .	LISA, su <i>lisa-fet.com</i> .
	LUSAS, su <i>lusas.it</i> .
	LS-DYNA, LSTC - Livermore Software Technology Corporation
	MARC, su <i>mscsoftware.com</i> .
	Morfeo, su <i>cenaero.be</i> .
	MSC-Nastran, su <i>mscsoftware.com</i> .
	Nastran In-CAD, Autodesk
	NISA, su <i>nisasoftware.com</i> .
	PAM, su <i>esi-group.com</i> .
	PERMAS, su <i>intes.de</i> .
	Range Software, su <i>range-software.com</i> .
	RFEM, su <i>dlubal.com</i> .
	SAMCEF, su <i>lmsitaly.com</i> .
	SAP2000, su <i>csiberkeley.com</i> .
	SDC Verifier, su <i>sdcverifier.com</i> .
	STRAND7 Pty. Ltd.
	SYSTUS, su <i>esi-group.com</i> .
	SYSWELD, su <i>esi-group.com</i> .
	Frame Simulator, su <i>fluidinteractive.com</i> (Trasmittanza serramenti - Windows e Mac OS X)
	Mold Simulator, su <i>fluidinteractive.com</i> (Analisi ponti termici - Windows e Mac OS X)
	TerMus-PT, su <i>acca.it</i> (Software Calcolo Ponti Termici)

4 INTEROPERABILITA' BIM-FEM

4.1 Definizione di Interoperabilità

L'interoperabilità è la predisposizione di un prodotto tecnologico a cooperare con altri prodotti senza particolari difficoltà, con affidabilità di risultato e con ottimizzazione delle risorse. Obiettivo dell'interoperabilità è dunque facilitare l'interazione fra applicazioni software differenti, nonché lo scambio e il riutilizzo delle informazioni anche fra sistemi informativi non omogenei (Simone Aliprandi, 2014).

Il tema dell'interoperabilità ha assunto negli ultimi anni una particolare rilevanza e anche gli organi politici gli hanno riservato sempre maggiore attenzione. Attualmente, infatti, possiamo disporre di una definizione di interoperabilità decisamente più articolata e completa, nata grazie ad uno studio promosso e concluso nel 2004 da IDABC (Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens) per conto della Commissione Europea: frutto di questa ricerca è stato un documento intitolato *European Interoperability Framework for pan-European eGovernment Services* contenente una più precisa definizione del concetto di interoperabilità e la fissazione degli obiettivi più importanti da perseguire da parte degli stati dell'Unione: «Interoperability means the ability of information and communication technology (ICT) systems and of the business processes they support to exchange data and to enable the sharing of information and knowledge.» (IDABC, 2004) (Simone Aliprandi, 2014).

A livello nazionale, si segnala fra tutti il numero 38 dei "Quaderni" a cura del CNIPA intitolato "Linee guida allo sviluppo di software riusabile multiuso nella Pubblica Amministrazione", dove l'interoperabilità viene definita come segue: «La capacità di un sistema software di interagire con uno o più sistemi specificati, scambiando dati mediante un determinato insieme di funzionalità. I dati scambiati sono definiti da un formato standard accettato dai sistemi che interagiscono tra loro e la comunicazione avviene tramite un protocollo concordato.» (I Quaderni - n. 38 Linee Guida Allo Sviluppo Di Software Riusabile Multiuso Nella Pubblica Amministrazione, 2009).

In generale quando si parla di interoperabilità BIM-FEM si intende uno scambio di dati fluido e automatizzato tra modelli BIM e software di calcolo FEM e riuscire ad ottenerlo risulta fondamentale nell'ambiente BIM, dove gli oggetti non sono solo degli elementi 3d, ma sono contenitori di informazioni quali ad esempio geometria, materiali, quantità, costi.

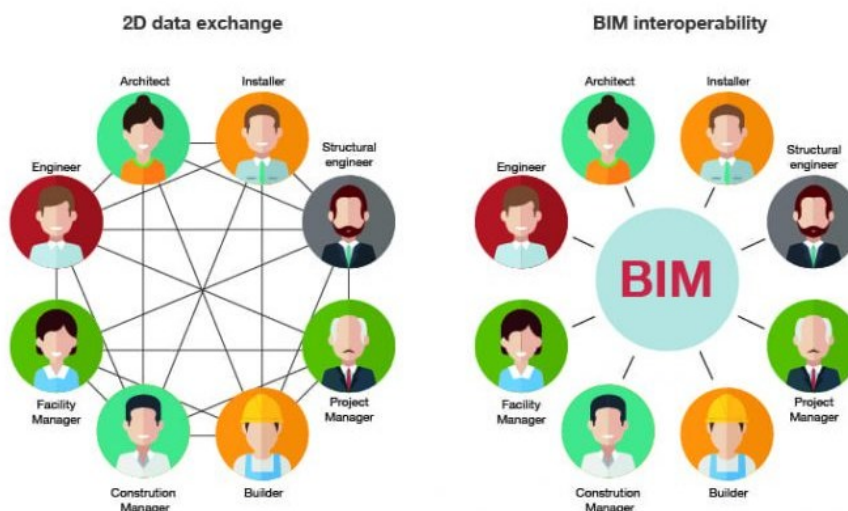


Figura 13 - Interoperabilità in edilizia. Fonte: (B. Riccardo, 2023)

4.2 Quadro Normativo

4.2.1 BIM

Lo scenario normativo per il BIM ha inizio a partire dal 2009, con la UNI 11337 *Edilizia e opere di ingegneria civile - Criteri di codificazione di opere e prodotti da costruzione, attività e risorse - Identificazione, descrizione e interoperabilità* (UNI, 2009). Tale normativa negli anni a venire subisce notevoli aggiornamenti, fino ad arrivare ad oggi ad essere strutturata in sette parti:

- Nella prima parte vengono descritte le tipologie di dato, le informazioni che possono contenere e le modalità di esplicitazione.
- Nella seconda parte, la più recente, ci si sofferma sulle relazioni tra gestione dei progetti e coordinamento delle informazioni.
- Nella terza parte sono descritte le schede di raccolta delle informazioni inerenti ai vari prodotti.
- La quarta parte illustra i LOD (Levels of Detail), classificandoli con lettere dalla A alla G a seconda del livello di definizione necessario per una specifica fase del progetto.
- Le parti quinta e sesta si indicano le regole, i ruoli e i flussi necessari alla produzione, gestione e trasmissione delle informazioni e la loro connessione e interazione nei processi di costruzione digitalizzati.
- La settima sezione evidenzia le figure professionali che operano con il BIM.

In ogni caso, attualmente, la UNI 11337 si applica insieme alla più recente e centrale UNI EN ISO 19650 *Organizzazione e digitalizzazione delle informazioni relative all'edilizia e alle opere di ingegneria civile, incluso il Building Information Modelling (BIM) - Gestione informativa*

mediante il *Building Information Modelling* (UNI, 2019). Tale normativa nasce nel 2019 e si sviluppa anch'essa in parti:

- La prima parte illustra i concetti e principi della normativa.
- La seconda parte indica le fasi di consegna dei cespiti immobili.
- La terza parte definisce le fasi gestionali dei cespiti immobili.
- La quarta fase determina le modalità relative allo scambio di informazioni.
- La quinta fase sottolinea l'importanza di un approccio orientato alla sicurezza per la gestione informativa.

Per ulteriori informazioni sull'evoluzione temporale della normativa in ambito BIM si rimanda alla figura seguente, che sintetizza in un grafico le norme pubblicate negli ultimi dieci anni dagli enti UN, CEN, ISO.

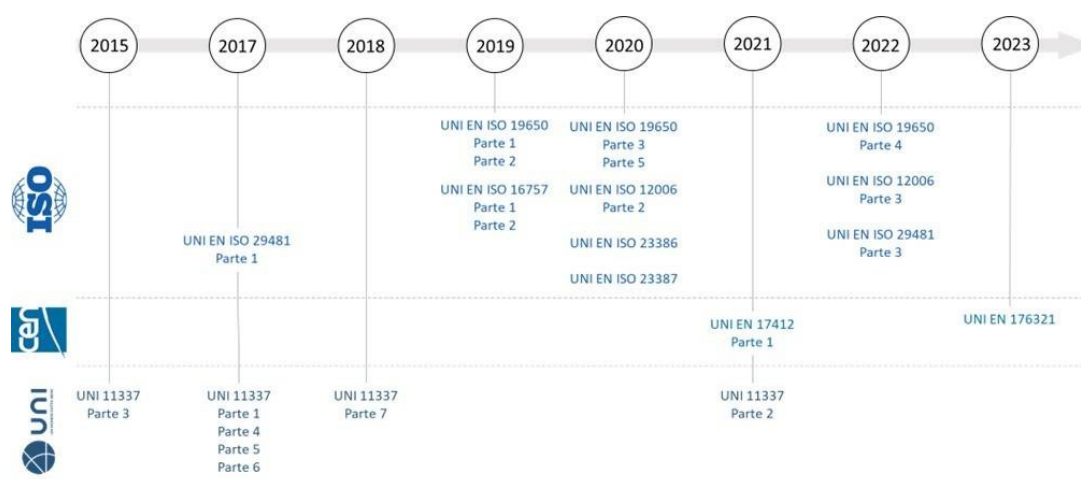


Figura 14 - Evoluzione temporale della normativa in ambito BIM. Fonte: (Maria Laura Longo, 2025)

4.2.2 NTC2018

Le Norme tecniche delle costruzioni sono la normativa vigente in Italia nell'ambito dell'ingegneria strutturale. Queste norme danno le indicazioni su come calcolare strutturalmente gli edifici affinché siano sicuri e funzionali. Inoltre, impongono il calcolo della struttura sulla base di combinazioni di carico, ovvero propongono modi con cui le azioni possono essere combinate, amplificandole a seconda della situazione.

Le azioni previste sulla struttura per effettuare un'analisi statica, combinate tra loro a seconda delle diverse condizioni di carico da verificare, sono:

- Carichi da vento (Q).
- Carichi da neve (Q).
- Peso proprio della struttura (G1).
- Carichi accidentali, ossia ad esempio il passaggio di persone (G2).

- Carichi eccezionali (P)

Nell'immagine seguente si riporta un paragrafo della normativa, in cui vengono illustrate le diverse combinazioni delle azioni.

2.5.3. COMBINAZIONI DELLE AZIONI

Ai fini delle verifiche degli stati limite, si definiscono le seguenti combinazioni delle azioni.

- Combinazione fondamentale, generalmente impiegata per gli stati limite ultimi (SLU):

$$\gamma_{G1} \cdot G_1 + \gamma_{G2} \cdot G_2 + \gamma_P \cdot P + \gamma_{Q1} \cdot Q_{k1} + \gamma_{Q2} \cdot \psi_{02} \cdot Q_{k2} + \gamma_{Q3} \cdot \psi_{03} \cdot Q_{k3} + \dots \quad [2.5.1]$$

- Combinazione caratteristica, cosiddetta rara, generalmente impiegata per gli stati limite di esercizio (SLE) irreversibili:

$$G_1 + G_2 + P + Q_{k1} + \psi_{02} \cdot Q_{k2} + \psi_{03} \cdot Q_{k3} + \dots \quad [2.5.2]$$

- Combinazione frequente, generalmente impiegata per gli stati limite di esercizio (SLE) reversibili:

$$G_1 + G_2 + P + \psi_{11} \cdot Q_{k1} + \psi_{22} \cdot Q_{k2} + \psi_{33} \cdot Q_{k3} + \dots \quad [2.5.3]$$

- Combinazione quasi permanente (SLE), generalmente impiegata per gli effetti a lungo termine:

$$G_1 + G_2 + P + \psi_{21} \cdot Q_{k1} + \psi_{22} \cdot Q_{k2} + \psi_{23} \cdot Q_{k3} + \dots \quad [2.5.4]$$

- Combinazione sismica, impiegata per gli stati limite ultimi e di esercizio connessi all'azione sismica E:

$$E + G_1 + G_2 + P + \psi_{21} \cdot Q_{k1} + \psi_{22} \cdot Q_{k2} + \dots \quad [2.5.5]$$

- Combinazione eccezionale, impiegata per gli stati limite ultimi connessi alle azioni eccezionali A:

$$G_1 + G_2 + P + A_d + \psi_{21} \cdot Q_{k1} + \psi_{22} \cdot Q_{k2} + \dots \quad [2.5.6]$$

Gli effetti dell'azione sismica saranno valutati tenendo conto delle masse associate ai seguenti carichi gravitazionali:

$$G_1 + G_2 + \sum_j \psi_{2j} Q_{kj} \quad [2.5.7]$$

Figura 15 - Combinazioni delle azioni. Fonte: (Norme Tecniche per Le Costruzioni, 2018)

4.3 Interoperabilità BIM-FEM

L'interoperabilità BIM-FEM è la capacità di trasferire dati in maniera automatizzata da un modello BIM, di tipo architettonico, a un software FEM, di analisi strutturale. Tale procedimento consente il passaggio di informazioni relative non solo alla geometria, ma anche a tutti gli altri aspetti del modello utili all'analisi strutturale: materiali, vincoli e carichi.

Grazie all'utilizzo di questo workflow, in cui non è necessario rimodellare la struttura, è possibile ridurre gli errori manuali e ottimizzare i tempi progettuali.

4.4 Professionisti interessati

La realizzazione di un edificio costituisce un'impresa complessa che richiede fasi interconnesse di progettazione, costruzione e monitoraggio continuativo. Queste attività sono affidate a un consorzio di professionisti, ciascuno con ruoli specifici, che collaborano per il successo del progetto.

In particolare, la fase di progettazione vede il coinvolgimento di un team multidisciplinare composto da figure con competenze specialistiche quali: architetto, interior designer, geometra, ingegnere strutturista, tecnici impiantisti, geologo, project manager. Risulta fondamentale che i progetti elaborati da ciascuna di queste figure siano armonizzabili, privi di interferenze e agevolmente integrabili tra loro.

Nel contesto dell'interoperabilità tra Building Information Modeling (BIM) e Finite Element Method (FEM), le professionalità coinvolte sulle quali ci si è concentrati in questo studio sono l'architetto per la modellazione BIM, e l'ingegnere strutturista per l'applicazione del FEM. Spesso, queste figure si trovano a dover interagire attraverso canali comunicativi disgiunti e con l'ausilio di strumenti e metodologie eterogenee. Tale frammentazione accresce il rischio di errori, derivanti principalmente dalle difficoltà di comunicazione e coordinamento.

L'adozione di un approccio basato sull'interoperabilità, come esplorato in questa tesi, sarebbe altamente benefico in ambito professionale poiché una progettazione pienamente integrata in ogni sua fase contribuirebbe significativamente all'ottimizzazione delle risorse e alla minimizzazione delle problematiche.

4.5 Vantaggi di una possibile soluzione

Un passaggio di dati da BIM a FEM realizzato in maniera automatizzata rende il processo progettuale molto più efficiente, eliminando la necessità di una rimodellazione della struttura. Il flusso di lavoro, in effetti, può essere concentrato esclusivamente sull'analisi della struttura, eliminando la necessità di dedicare energie ad operazioni ripetitive e di manipolazione dei dati.

Tutto questo porta senza dubbio ad una riduzione dei tempi e dei costi produttivi. Inoltre, assicurando che i dati siano trasferiti in maniera biunivoca da un modello all'altro, si evitano diversi errori derivanti da passaggi non necessari in quanto automatizzabili.

È poi inevitabile che la possibilità di accedere contemporaneamente al progetto da parte di diversi professionisti, faciliti la flessibilità delle fasi di ideazione dello stesso: viene verificata in tempo reale la compatibilità dei progetti architettonico e strutturale, evitando interferenze.

Infine, l'uso dell'OpenBIM risulterebbe vantaggioso per due ragioni fondamentali: la prima riguarda la libertà di scelta da parte dei professionisti, che grazie al formato aperto potrebbero utilizzare qualsiasi tecnologia poiché svincolati da specifici software. Inoltre, IFC, essendo basato su una struttura informativa, consente di realizzare un modello geometrico contenente dati anche sulla struttura, come ad esempio materiali e caratteristiche meccaniche, favorendo il collegamento tra modello architettonico e modello strutturale (Michele Zucco, 2022).

Capitolo Secondo

ASPETTI METODOLOGICI NELL'INTEROPERABILITÀ BIM-FEM

1 ANALISI DELLA LETTERATURA

Esistono diversi approcci con cui in letteratura si è affrontato il tema dell'interoperabilità tra un software BIM e uno FEM e secondo (F. Beirnaert & A. Lippens, 2018) si rifanno tutti fondamentalmente a tre tipi di gestione delle informazioni: metodo più comune è quello basato su file, in cui si confrontano i formati per convertire i dati. Una seconda tecnica prevede l'utilizzo di link diretti tra due software, consentendo uno scambio di informazioni in tempo reale, ma impedendo all'utente di verificare come avvenga questo passaggio. Come ultima alternativa, i dati, anziché essere trasferiti, vengono semplicemente condivisi su un database esterno che funge da piattaforma intermedia alla quale fanno riferimento i software per estrarre le informazioni.

1.1 Confronto di file

L'interoperabilità basata sul confronto dei file è influenzata innanzitutto dal tipo di formati utilizzati nel processo: proprietari o aperti. Quando si lavora con due programmi diversi dello stesso fornitore, è molto probabile che sia disponibile un formato proprietario. Questo formato permette al programma di accettare o produrre dati in modo specifico, garantendo una transizione fluida tra le diverse applicazioni. Se, invece, i software provengono da fornitori differenti, l'uso di un formato proprietario non è praticabile. In questi casi, l'adozione di formati aperti rappresenta una scelta più sicura, soprattutto quando non si sa quale software verrà impiegato nelle fasi successive. La standardizzazione di questi formati consente lo scambio di informazioni tra diverse applicazioni software. Nel settore AEC, l'IFC è l'esempio più utilizzato di formato standard aperto per la modellazione informativa (F. Beirnaert & A. Lippens, 2018).

Un ulteriore aspetto che è da valutare riguarda la modalità con cui avviene la conversione: se questa è manuale, semi-automatizzata o completamente automatizzata. Questi aspetti influenzano il controllo che l'utente ha sul processo, definendo la flessibilità e l'adattabilità della metodologia, in quanto più è automatizzato il procedimento e minore sarà la possibilità di poterlo modificare e personalizzare sulla base delle proprie esigenze. Per questo, le conversioni automatiche dovrebbero essere gestite con grande cura a causa delle limitazioni del processo, che possono causare conversioni incomplete quando l'utente non è consapevole dei limiti del metodo. Altrettanto, una conversione manuale, nella quale le informazioni vengono importate nello strumento di destinazione senza alterazioni, deve essere

scelta con consapevolezza: perché il file possa essere importato, la funzionalità di archiviazione e presentazione dei due sistemi deve corrispondere. Esiste poi la conversione semi-automatica, nella quale il modello non viene convertito completamente. L'utente può scegliere manualmente su quale parte del modello l'operazione deve essere eseguita utilizzando algoritmi e regole predefinite (F. Beirnaert & A. Lippens, 2018).

Andando ad analizzare la conversione attraverso file proprietari in letteratura, si trovano principalmente casi di trasferimento dati basato sulle informazioni geometriche delle strutture: infatti vengono valutati i file di tipo CAD, tipici di software molto diffusi come quelli Autodesk. Questo tipo di file è importabile nei software FEM, trasferendo i dati sulla geometria della struttura. Le informazioni sui materiali, sulle proprietà e sulle condizioni al contorno, invece, dovranno essere inserite manualmente dall'utente che utilizza il software di analisi strutturale.

Si è visto che il software Midas FEA risulta essere uno dei migliori per questa metodologia, perché consente di trattare elementi 3D e offre la possibilità di importare geometrie 3D avanzate da meshare. Lavori precedenti (Oreni et al., 2014) hanno riscontrato che, tra i diversi formati che possono essere importati in Midas FEA, il formato STEP è particolarmente adatto per mantenere le forme complesse durante lo scambio di informazioni. Una volta ottenuto il file STEP di partenza, il modello FEM è ricavabile utilizzando l'algoritmo di mesh automatico 3D fornito da Midas FEA. Le criticità riscontrate nella conversione in questo contesto risultano essere principalmente la semplificazione del disegno delle aperture, l'unione delle diverse parti delle pareti e l'offset dei bordi che talvolta risultano in mesh incompatibili (P. Crespi et al., 2015).

Valutando l'utilizzo di formati aperti per l'interoperabilità BIM-FEM, emerge innanzitutto la preferenza nell'utilizzo del formato aperto IFC come Standard adottato a livello internazionale, che, per via della sua maturità, è in grado di colmare il divario tra le parti interessate e le diverse fasi del progetto. L'IFC non ha ancora raggiunto il suo pieno potenziale, ma diversi scambi di informazioni sono già testimoniati in letteratura tra questo formato e software FEM come ad esempio RFEM, SAP2000, Edilus, Tekla Structures.

A livello metodologico le procedure che vengono seguite sono di due tipologie: è possibile dare priorità alla geometria del modello, convertendo il file IFC in un file CAD da importare nel software FEM e inserendo in un secondo momento manualmente i valori dei materiali, delle proprietà e delle condizioni al contorno, oppure, in alternativa, si converte IFC nel file con formato proprietario del software FEM utilizzato per l'analisi strutturale, valutando l'importazione diretta di tutte le caratteristiche analitiche del modello.

La conversione di modelli da IFC a formati CAD si avvale prevalentemente dei file STEP e STL, riconosciuti per la loro efficacia nel mantenere la complessità geometrica durante lo scambio di dati. Un approccio metodologico prevede la trasformazione iniziale di un modello

BIM basato su IFC in un "modello di connessione", realizzata tramite un algoritmo specifico. Successivamente, questo modello viene sottoposto a una fase di meshatura, che lo prepara come un modello di analisi a elementi finiti, idoneo per calcoli e simulazioni.

Nonostante la capacità del formato IFC standard di descrivere accuratamente la geometria di una struttura completa, la sua applicazione per simulazioni numeriche è limitata. La descrizione implicita della topologia strutturale e delle interrelazioni tra i componenti rende l'analisi complessa. In particolare, la definizione delle connessioni strutturali tramite i metodi di estensione generici di IFC si rivela problematica.

Per superare queste limitazioni, è essenziale rendere esplicite le relazioni topologiche e le connessioni attraverso la creazione di un modello BRep (Boundary Representation) arricchito da oggetti attributo. Il BRep descrive le entità tramite "sezioni" di superficie di contorno: ciascun oggetto è delineato da un insieme di piccole facce, ciascuna definita da linee di contorno che uniscono vertici. L'orientamento delle facce è determinato dalla loro normale direzionale. Il vantaggio principale della tecnologia BRep risiede nell'abbondanza di informazioni dettagliate su punti, bordi e piani, e sulle loro reciproche relazioni, facilitando la generazione di wireframe e diagrammi di proiezione. Ciononostante, un inconveniente significativo è l'arbitrarietà e la non univocità della forma del contorno, aspetto che può ostacolare lo sviluppo di interfacce efficienti per la conversione dei dati (Zhao Xu et al., 2019).

Inoltre, gli oggetti complessi e altamente dettagliati sono spesso suscettibili a difetti geometrici, generalmente imputabili a imprecisioni nella descrizione presente nel file IFC. Problemi comuni includono la presenza di superfici non chiuse, sovrapposizioni o orientamenti scorretti. Sebbene piccole discontinuità (entro pochi centimetri) non compromettano l'algoritmo, si cerca attivamente di correggere le forme difettose per ottimizzare i processi successivi. Questa riparazione si attua chiudendo le lacune minori tramite estensione delle facce e successivamente estraendo la superficie di involucro, assicurando un orientamento uniforme delle facce in base all'analisi geometrica di van Treeck e Rank (C. van Treeck, 2006). Dove fattibile, le superfici complanari adiacenti vengono consolidate (Fichter et al., 2022).

Nel caso in cui invece il file IFC venga convertito in un file non-Cad leggibile dal software FEM, i principali programmi che sono stati usati in letteratura sono ad esempio RFEM, SAP2000, Edilus, Tekla Structure, ANSYS APDL. In questo caso la metodologia consiste nell'esportare il file IFC dal modello BIM e importarlo direttamente nel software FEM (S. Quintero Botero & L. A. Gutiérrez Bucheli, 2018), oppure convertirlo attraverso un applicativo in un formato proprietario e importare l'output nel software.

Lo studio di Gerbino et. al (2021) ha evidenziato, attraverso un modello molto semplice, alcune delle limitazioni del protocollo IFC. Tra queste, la questione dello scambio dati, dove, in

molti casi, il modello scambiato via IFC non manteneva tutti i dati inizialmente ereditati (S. Gerbino et al., 2021).

Tale aspetto viene confermato e ribadito nella pubblicazione di Quintero et. al (2018), in cui la metodologia prevista ha verificato sia l'esportazione di IFC con Revit e con Allplan, sia il modo in cui il software SAP2000 riesce ad importare direttamente il file IFC. Anche in questo caso, per quanto riguarda l'importazione della geometria nel software SAP 2000, si è visto che il trasferimento dati ha generato problemi di connettività degli elementi laddove le intersezioni erano state modellate dal centro dell'elemento, perché questo non è corretto in tutti i casi.

Inoltre, a seguito dell'applicazione della metodologia, si è concluso che, nel caso di Revit, l'esportazione in IFC 4 Reference View è molto più accurata della IFC Design Transfer View, mentre, confrontando il modo in cui i software CAD esportano IFC, Allplan è risultato essere il più accurato dei due, generando molte meno imprecisioni (S. Quintero Botero & L. A. Gutiérrez Bucheli, 2018), (J. Jia et al., 2022).

In alternativa, un'ulteriore metodologia che viene proposta in letteratura riguarda una sequenza di step per convertire il file IFC in uno proprietario FEM, il tutto partendo da uno studio molto approfondito dello Standard IFC.

In questo contesto nasce una mappatura dei dati dai modelli IFC ai sistemi FE (Finite Element) per l'analisi strutturale che, al tempo stesso, analizza sia l'architettura dei dati dello standard IFC per individuare la struttura del modello analitico, sia, contemporaneamente, costruisce una rappresentazione topologica semantica tra strutture, elementi finiti, nodi e i parametri correlati (Raimar Scherer & Fangzheng Lin, 2020).

Le applicazioni a casi studio di questa metodologia prevedono la stesura di un codice convertitore che si basa sulla libreria IFCopshell per la lettura dei dati del file IFC, il quale genera un file output con formato relativo al software FEM adottato nella sperimentazione, come ad esempio ANSYS APDL o Gmsh.

Dallo studio della letteratura in merito all'applicazione di questo approccio emerge fondamentalmente una criticità: l'attuale standard IFC presenta ancora limitazioni nell'espressione dei concetti di elementi finiti (J. Jia et al., 2022).

Tuttavia, tale metodo risulta valido e ben funzionante per il caso studio dell'aggregato edilizio presentato da Leonardi et. al (2024): il modello è stato utilizzato per eseguire analisi lineari, dimostrando la capacità dell'algoritmo sviluppato di utilizzare i dati del modello BIM per creare un modello a elementi finiti funzionale e scalabile a geometrie complesse. In fase di modellazione è stata evidenziata la necessità di porre particolare attenzione alle connessioni tra elementi, che vengono meshati tra loro ed esportati in IFC4. Anche in questo caso studio i carichi sono stati inseriti manualmente dall'utente all'interno del software FEM (M. Leonardi et al., 2024).

1.2 Link diretti

Per quanto riguarda invece l'utilizzo di link diretti che genera uno scambio di informazioni in tempo reale tra due applicazioni, il sistema che genera i dati funge da fonte primaria di informazioni, mentre l'elaborazione computazionale avviene nel sistema di destinazione. L'utilizzo di un collegamento diretto accelera significativamente lo scambio di dati rispetto ai metodi tradizionali basati su file. Tuttavia, ciò comporta anche una specifica esigenza: entrambi gli strumenti devono essere operativi simultaneamente, sebbene non sia indispensabile che risiedano fisicamente sullo stesso hardware (F. Beirnaert & A. Lippens, 2018).

Un esempio di questo tipo di strumento riguarda Revit e Midas, con il plug-in chiamato "Midas Link per Revit Structure", che garantisce il trasferimento di tutti i dati caratterizzanti gli elementi strutturali di Revit in Midas. Tale collegamento è bidirezionale, consentendo di apportare modifiche nel software FEM e, aggiornando il modello, di restituirle al modello BIM.

Questo plug-in è efficace quando le tipologie di materiali e di sezioni appartengono a entrambi i database dei due software, riconoscendoli in automatico; per tutti gli altri, invece, viene richiesta un'associazione manuale dei parametri perché per una stessa tipologia di sezione possono essere usate simbologie diverse. Un altro aspetto che emerge tra le problematiche di questo procedimento riguarda il modo in cui Revit trasforma il modello fisico in modello analitico, il quale poi viene analizzato in Midas: laddove gli elementi fisici, come travi e pilastri, non siano allineati nel centro della loro sezione, il modello analitico che viene a crearsi è errato nelle connessioni tra elementi come mostrato nell'immagine. Questo problema deriva da un'automatizzazione della fase di modellazione analitica in Revit, che non è risolvibile se non eliminando gli elementi analitici creati in maniera errata e rimodellandoli manualmente (L. Muratore, 2018).

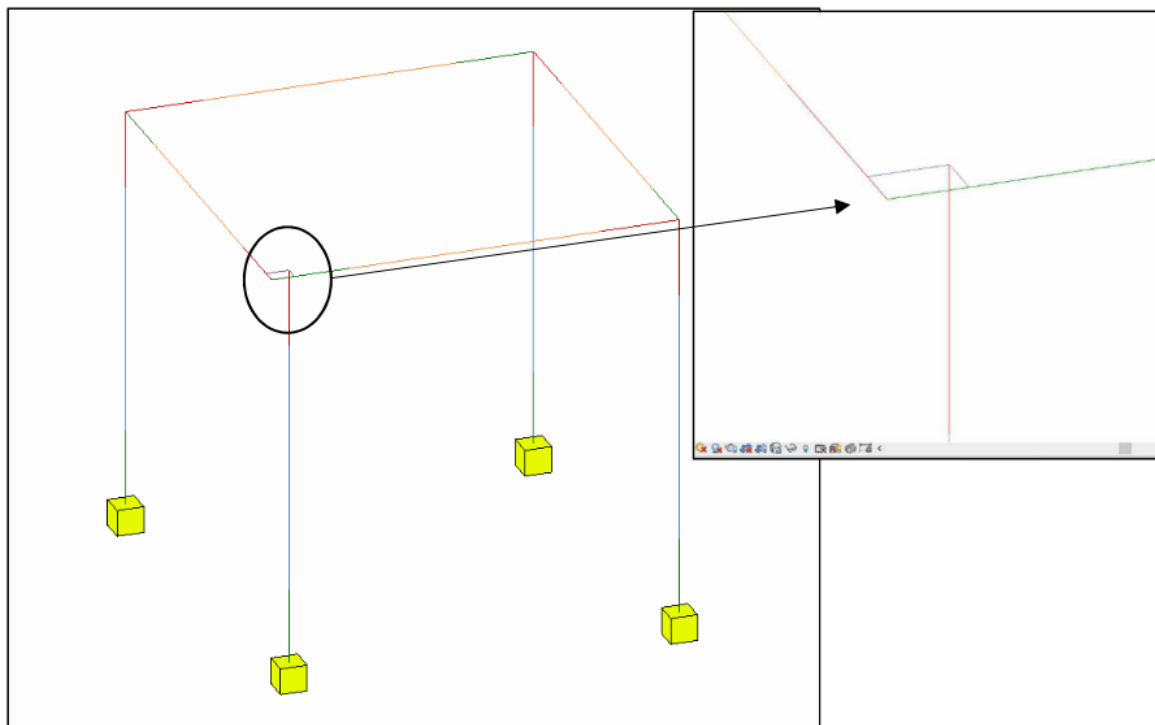


Figura 16 - Errore nelle connessioni tra elementi analitici. Fonte: (L. Muratore, 2018)

Considerando lo stato dell'arte, si può dire che Revit goda di diversi link che possono essere installati nel software BIM per favorire il collegamento con un programma di analisi FEM, come ad esempio con Robot Structural Analysis, SCIA Engineer e RFEM.

Nel complesso, si può affermare che tutti e tre i collegamenti tra Revit e i software di analisi FEM citati sono dei metodi solidi, che consentono di trasferire quasi interamente i dati da un modello BIM a uno FEM. Il flusso di lavoro è facile da seguire e il processo di progettazione diventa più efficiente per l'ingegnere. Le uniche mancanze riguardano l'esportazione dei carichi, che non avviene per nessuno dei link, il trasferimento delle barre di armatura altrettanto non funziona nel link con SCIA Engineering e la proprietà del materiale non si ritrova a seguito del collegamento con RFEM (F. Beirnaert & A. Lippens, 2018).

Anche Tekla Structures, un altro software BIM, offre la possibilità di collegarsi direttamente a programmi per l'analisi strutturale come SCIA Engineer e STAAD.Pro: la geometria e l'orientamento degli elementi risulta corretto a seguito del trasferimento, mentre, nel link Tekla-SCIA Engineer, le proprietà e i materiali di Tekla vengono ignorati dal software FEM in quanto la mappatura del link segue la denominazione di Tekla, consentendo di utilizzare solo le proprietà fornite da SCIA Engineer. Il link Tekla-STAAD.Pro, invece, risulta maggiormente deficitario non consentendo il trasferimento non solo dei carichi, ma anche del corretto orientamento degli assi, dei materiali e delle proprietà (F. Beirnaert & A. Lippens, 2018).

Un ulteriore metodo che utilizzi un link diretto è quello in cui viene realizzato un codice all'interno delle API del software BIM, che estrapoli le informazioni dal modello BIM e le riorganizzi, consentendo di visualizzarle nel software FEM.

Un esempio di questa tecnica è l'utilizzo di uno script nelle API di Revit, realizzato a seguito di uno studio approfondito del linguaggio proprietario di Ansys, APDL, ottenendo come risultato un collegamento in tempo diretto con il software di analisi strutturale ANSYS. L'API di Revit è contenuta in due file di componenti di interfaccia (RevitAPI.dll e RevitAPIUI.dll) ed è disponibile nell'ambiente RevitApiui.dll.NET per lo sviluppo. I linguaggi .NET compatibili includono C#, che può essere utilizzato per lo sviluppo dell'interfaccia. I risultati della sperimentazione dimostrano che le informazioni delle componenti strutturali del modello come i dati di posizionamento, le dimensioni geometriche, le condizioni al contorno, il regime di carico e attributi dei materiali, possono essere ottenute automaticamente in base ai metodi di estrazione dati specifici per i diversi componenti. Inoltre, all'interno del programma di interfaccia, possono essere implementate la gestione delle classi di famiglia delle componenti e la valutazione delle intersezioni. Le informazioni del modello BIM ottenute vengono esportate dal programma di interfaccia sotto forma di file in formato APDL ed eseguite in ANSYS, raggiungendo la piena automazione dell'analisi agli elementi finiti del modello (H. Zhou et al., 2021).

Un altro metodo riguarda il software Tekla Structures lato BIM e Midas lato FEM: esistendo un link diretto da BIM a FEM, esiste una tecnica sperimentale per la realizzazione di un link nella direzione opposta FEM-BIM. Tale studio prevede la stesura di uno script nel linguaggio C# realizzato all'interno delle API di Tekla Open API. In questo modo, a seguito di uno studio delle strutture dati del modello BIM e di quello FEM, si è in grado di raccogliere in modo organizzato le informazioni dal modello Midas, adattare i dati a Tekla e implementarli nel modello BIM. È stato evidenziato che la corretta associazione di ogni elemento da un software all'altro è strettamente legata alla correttezza dei modelli analitici considerati nei due programmi: nell'associare ciascun elemento del software FEM al corrispondente elemento del software BIM, il riconoscimento parte da ogni singolo nodo, poiché la loro posizione deve coincidere nei due modelli, mentre lo stesso non si può dire per gli elementi. Infatti, mentre nel software FEM gli elementi collegano sempre un nodo all'altro, ciò non avviene sempre in Tekla Structures, dove lo stesso modello analitico può derivare da modelli fisici differenti. Considerato questo aspetto, sebbene alcuni aspetti debbano ancora essere implementati, il link sviluppato da Tekla a Midas risulta essere funzionale all'interoperabilità BIM-FEM automatizzando una procedura altrimenti manuale (V. Miotto, 2022).

1.3 Database intermedi

Un metodo per trasferire i dati tra BIM e FEM è quello che fa uso di database esterni di tipo tabellare per riorganizzare le informazioni del file di partenza ed estrapolarle importandole

nell'altro software utilizzato, prevedendo oltretutto una bidirezionalità del processo. Anche in questo caso, esistono due modalità con cui le informazioni possono essere lette dal software di input, raccolte in un database e lette dal programma finale: attraverso un codice esterno oppure andando ad intervenire direttamente nelle API del software utilizzato per il modello iniziale. Un'ulteriore differenza tra le metodologie sta nella scelta del database, che può essere ad esempio SQL oppure Excel.

Nel primo caso, vi sono studi che partono da un file IFC e analizzano nel dettaglio come leggerne le informazioni per raccoglierle al meglio in un database SQL realizzando un parser ad hoc. L'intero parser può essere suddiviso in tre parti fondamentali: Tokenizer, Syntactic Parser (analizzatore sintattico) ed Entity Factory. Esso acquisisce le informazioni complete dal BIM e le trasforma in entità Java. Nella fase di analisi sintattica vengono create solo le entità, senza procedere all'assegnazione degli attributi. Grazie all'uso del multi-threading, l'assegnazione degli attributi viene completata dalla Entity Factory. Al fine di salvare le informazioni nel database per le attività successive, la ricerca definisce la struttura e le relazioni delle tabelle in conformità allo standard IFC, e utilizza il framework MyBatis per archiviare le entità analizzate in Java all'interno di un potente database MySQL (Ruoxiao Wang & Tang Yongjing, 2021).

Gli autori *Y. Zou, F. Hu, H. Yang, J. Cai, H. Pan, Q. Zhang*, nel 2024, hanno proposto una metodologia che, utilizzando le API di Revit e il linguaggio C# per lo sviluppo di applicativi personalizzati, prevede la conversione del modello solido geometrico (descritto tramite CSG-Brep) in un flusso di dati APDL. Ciò crea un'interfaccia con il software di analisi agli elementi finiti ANSYS per implementare la conversione del modello BIM-FEM nella progettazione esecutiva, ottimizzando l'algoritmo di conversione per modelli di analisi complessi che richiedono un'elevata precisione per strutture in acciaio dalle forme speciali. In questa ricerca viene validato il metodo attraverso la sperimentazione, dimostrando il funzionamento del codice con uno scambio dati che include geometria, proprietà dei materiali, proprietà della sezione trasversale degli elementi, carichi e condizioni al contorno (Y. Zou et al., 2024).

Nel secondo caso, si individua un metodo che fanno uso delle API nel software FEM Strand7 per realizzare uno scambio diretto e bidirezionale di informazioni da Strand7 e un database Excel. Altrettanto, lo stesso collegamento al database univoco è stato realizzato da software BIM come Tekla e Revit, attraverso un codice Python esterno, garantendo l'accesso alle medesime informazioni che identificano il modello in Strand7: la geometria iniziale, inclusa la posizione dei nodi, la connettività tra i nodi (eventuali svincoli traslazionali o rotazionali, o offset), gli elementi strutturali come travi o cavi, le proprietà dei materiali, il modulo elastico e le aree delle sezioni trasversali di ogni elemento, i carichi e le condizioni al contorno che descrivono i vincoli sulla struttura. Questa metodologia è stata sperimentata da M. Chernyavsky e H. Kestel (2023), dimostrando la solidità del metodo attraverso la progettazione di uno stadio dalla complessa geometria (M. Chernyavsky & H. Kestel, 2023).

2 CONFRONTO TRA MODELLI DATI

Come emerso dallo studio del modello BIM e di quello FEM presentati al capitolo primo, si osserva che questi hanno differenze strutturali nel modo in cui descrivono gli elementi e la causa sta nell'uso differente che se ne fa (T. Rácz & T. Olofsson, 2009): il modello BIM ha un'esigenza costruttiva che necessita di descrivere una struttura sotto tutti i suoi aspetti, mentre un modello analitico deve essere focalizzato sulle mere caratteristiche strutturali degli elementi, rimuovendo le informazioni sovrabbondanti e riducendo il modello ad un'astrazione matematica sintetica, semplificata il più possibile così da ottimizzare il calcolo. Dalla letteratura emergono delle criticità nelle metodologie di interoperabilità tra BIM e FEM e, con il confronto che segue, si vuole dimostrare la difficoltà della comparazione dei modelli, dovuta alle loro importanti differenze.

2.1 Rappresentazione geometrica e connessioni

La progettazione FEM-Design ha un proprio modo di gestire la geometria per supportare l'obiettivo principale: la creazione del modello di analisi. La gestione e l'approccio alla creazione della geometria del modello possono essere piuttosto diversi negli strumenti di progettazione, specialmente negli strumenti di progettazione di altre discipline, ad esempio nei sistemi architettonici dove il compito è creare un modello fisico. La differenza principale tra modelli fisici e modelli analitici è che il modello fisico è composto da oggetti 3D senza intersezioni, mentre in FEM-Design la rappresentazione 1D e 2D degli oggetti 3D dovrebbe formare un modello analitico continuo. L'intersezione dell'estensione 3D generata dagli oggetti analitici, visualizzando il parametro della sezione trasversale, nel modello di analisi non è rilevante. Il posizionamento degli elementi del modello nello spazio potrebbe presentare un'apparente incompatibilità tra modelli fisici e modelli analitici. Ad esempio, l'estensione degli elementi del modello analitico potrebbe non corrispondere al confine degli elementi del modello fisico pertinenti.

La conversione del modello fisico in modello analitico è difficile per casi generici (A. Rönneblad, 2003). Tradizionalmente questo è il compito dell'ingegnere strutturale e, sebbene ci siano situazioni in cui la conversione può essere automatizzata, ciò richiederebbe molto probabilmente limitazioni e l'implementazione di linee guida per l'attività di modellazione (T. Rácz & T. Olofsson, 2009).



Figura 17 – Rappresentazione geometrica del modello architettonico, strutturale e analitico. Fonte: (T. Rácz & T. Olofsson, 2009).

Per la conversione in un modello ad elementi finiti, è fondamentale che le intersezioni geometriche tra gli elementi siano trasformate in nodi poiché nell'analisi strutturale a questi nodi vengono poi associate condizioni al contorno, permettendo al software di calcolare le reazioni e le forze trasmesse tra gli elementi, in accordo con i vincoli definiti nel modello.

Spesso, a un livello di rappresentazione basato su volumi solidi, le linee d'asse degli elementi non si intersecano in un unico punto, anche se visivamente, in 3D, i collegamenti possono apparire corretti. Questa discrepanza richiede una correzione geometrica e una ricostruzione della connettività per ottenere un modello analitico valido (L. Muratore, 2018).

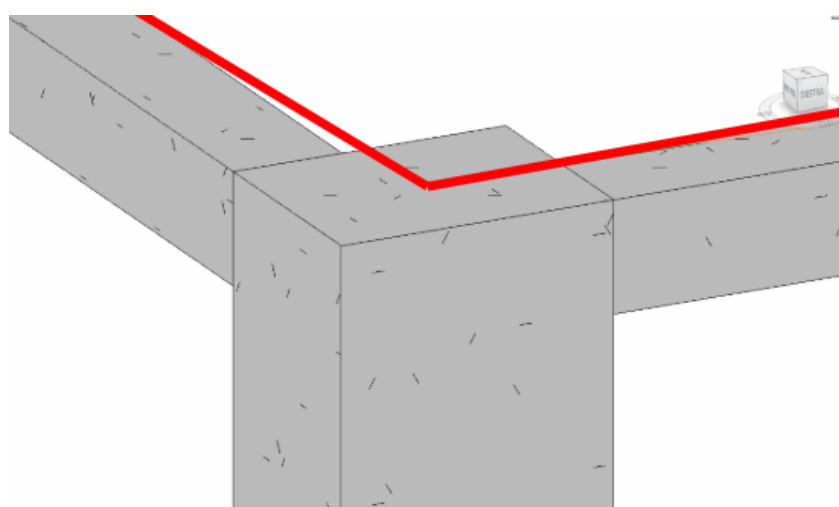


Figura 18 - Connessioni BIM e FEM. Fonte: (L. Muratore, 2018).

Per ovviare a queste problematiche, si può ricorrere a entità specifiche del "Domain Layer", in particolare quelle relative al dominio dell'analisi strutturale (IfcStructuralAnalysisDomain), che offre un livello di dettaglio più elevato per l'interazione con i software FEM.

All'interno di questo dominio, la modellazione si sposta dalla sola considerazione dei volumi solidi a quella delle linee d'asse e delle superfici di calcolo. Gli elementi si connettono direttamente ai nodi, configurando un modello analitico già pronto per l'analisi agli elementi finiti.

2.2 Informazioni contenute e struttura dei dati

Come esposto in precedenza, il BIM organizza le informazioni multidisciplinari del modello dati secondo dieci dimensioni. Il file opera come un database relazionale, strutturando i contenuti in tabelle composte da record e attributi. Di conseguenza, ogni oggetto digitale del modello è portatore di dati tecnici; si consideri, a titolo esemplificativo, l'oggetto IfcBeam, che

racchiude al tempo stesso informazioni su dimensioni, materiali, prestazioni termiche, caratteristiche meccaniche della sezione (Ruoxiao Wang & Tang Yongjing, 2021).

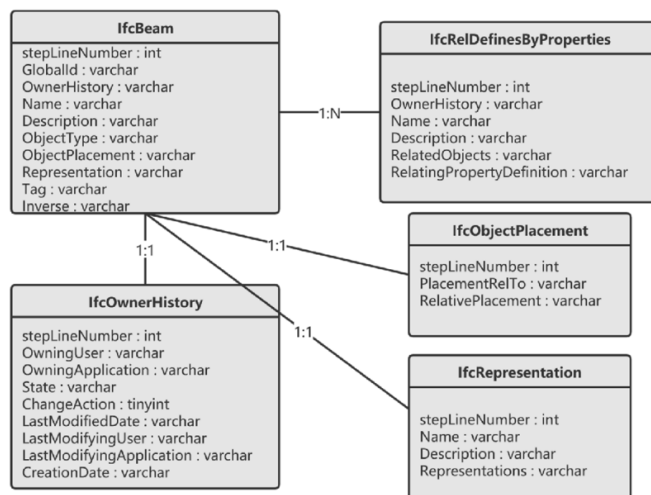


Figura 19 – Esempio database relazionale IFCBeam. Fonte: (Ruoxiao Wang & Tang Yongjing, 2021)

Nel contesto del Metodo degli Elementi Finiti, invece, le entità sono definite attraverso sequenze di vettori, i cui parametri vengono sintetizzati in valori numerici funzionali alle necessità di calcolo. Il set informativo di questo modello si restringe alla rappresentazione geometrica dell'opera, alla definizione delle condizioni al contorno e alle sollecitazioni e alle caratteristiche meccaniche dei materiali, tra cui la densità, il modulo elastico e la matrice di rigidezza.

Queste differenze creano una criticità laddove il sistema di origine o quello di destinazione conservino le informazioni relative allo stato del modello ma le descrivono in modo diverso, poiché crea difficoltà di interpretazione delle stesse da parte del sistema di destinazione (T. Rácz & T. Olofsson, 2009).

2.3 Diversi livelli di funzionalità

Ogni applicazione software offre determinate funzionalità che non sempre esistono anche negli altri programmi e in termini di interoperabilità questo aspetto può diventare una criticità quando ci si trova in una situazione in cui la funzione è disponibile nel software di modellazione, ma la conversione risulta impossibile perché la stessa non è supportata dal software di analisi. Questo tipo di incongruità è stata rilevata per modellazioni particolari, ad esempio per elementi curvi (Febe Beirnaert & Alice Lippens, 2018), oppure per il modo in cui sono definiti i materiali (T. Rácz & T. Olofsson, 2009).

Per gestire la differenza tra le funzionalità possono essere adottati diversi metodi:

- L'eliminazione dei concetti non supportati: è il metodo più semplice e comporta la perdita della funzionalità e dei relativi dati. Se il programma di destinazione non supporta un muro curvo, l'utente se ne accorgerà immediatamente. Il vero problema sorge quando la differenza tra le funzionalità non è facilmente individuabile, come nel caso di diversi livelli informativi sui materiali utilizzati. Idealmente, il software dovrebbe informare l'utente della perdita di informazioni, ma è piuttosto complesso gestire la notifica di un concetto inesistente nel sistema ricevente (F. Beirnaert & A. Lippens, 2018).
- La sostituzione della funzionalità: in questo caso si verifica comunque una perdita di dati, ma l'approccio è meno problematico rispetto all'eliminazione dei concetti non supportati. L'obiettivo è emulare la soluzione desiderata, il che rappresenta una sfida per gli sviluppatori. Riprendendo l'esempio del muro curvo, questo potrebbe essere sostituito da uno o più muri rettilinei. Considerando che la perdita di dati dovrebbe essere ridotta al minimo, è evidente che l'uso di più segmenti murari permetterà di approssimare meglio il modello originale. Dovrebbero essere utilizzati tanti muri quanti ne servono per rientrare in un margine di errore accettabile. Questa soluzione, pur permettendo un'analisi affidabile, non è comunque ideale: a causa della sostituzione, inizieranno a circolare diverse varianti dello stesso modello. Quando, in una fase successiva, tutte le informazioni verranno riunite in un unico modello di coordinamento, si verificheranno errori nel consolidamento dei dati (F. Beirnaert & A. Lippens, 2018).
- La sincronizzazione delle funzionalità: è la soluzione più difficile. In questo caso, i livelli di funzionalità di entrambi i programmi vengono sincronizzati. Normalmente, la sincronizzazione avviene in modo bidirezionale, ma ciò porterebbe a un depotenziamento (*degradation*) di un programma e a un potenziamento (*elevation*) delle funzioni nell'altro. Il depotenziamento è indesiderabile poiché comporta una perdita di operatività. Idealmente, la sincronizzazione dovrebbe avvenire in modo da generare solo un incremento del livello di funzionalità o, in altre parole, implementando nel programma solo le funzioni mancanti. L'idea alla base di questa conversione è semplice, ma l'implementazione pratica nel software è decisamente più complessa (F. Beirnaert & A. Lippens, 2018).

3 OBIETTIVO DELLA RICERCA TESI

Date le diversità strutturali esistenti tra BIM e FEM, il confronto dei modelli risulta difficoltoso. Per creare delle connessioni, tuttavia, bisogna partire dai punti di contatto esistenti tra questi modelli.

A seguito delle riflessioni proposte in letteratura, in questa tesi si è preferito procedere allo sviluppo di un applicativo per il passaggio dati da BIM a FEM dando priorità all'utilizzo del

formato aperto IFC, essendo questo ottimale in termini di flessibilità e adattabilità. Per le stesse ragioni, si è scelto di convertire IFC in un file lato FEM senza utilizzare le API, proprietarie di un software specifico e vincolanti allo stesso, e preferendo invece un codice eseguibile esternamente a entrambi i programmi. Il convertitore, sulla base delle relazioni tra elementi dei file stabilite dall'utente, deve essere in grado di generare un file FEM a partire da un file IFC di input. L'obiettivo che si pone questa tesi è realizzare tale applicativo adottando le seguenti fasi procedurali:

1. Studio dello Standard IFC di input
2. Creazione del file FEM
3. Popolamento del file FEM con le informazioni di input
4. Individuazione di un linguaggio informatico e stesura di un codice che attui la conversione IFC-FEM
5. Valutazione dei risultati ottenuti

Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM
IFC e NASTRAN
Alice Pin

Capitolo Terzo

METODOLOGIA E CASO STUDIO

La presente ricerca sperimentale intende presentare e validare una metodologia per la conversione di modelli IFC in modelli di analisi strutturale in formato FEM. La metodologia proposta si articola attraverso una serie di passaggi ben definiti, ciascuno dei quali contribuisce all'efficacia dell'intero processo.

Il punto di partenza di questa ricerca è un'approfondita analisi del file FEM a cui si vuole arrivare con la conversione. Lo studio iniziale vuole principalmente comprendere la struttura del file, decodificando la logica con cui i dati sono organizzati e rappresentati all'interno del formato specifico del software. È inoltre fondamentale identificare gli elementi chiave che caratterizzano un modello strutturale come nodi, elementi finiti (travi, pilastri), proprietà dei materiali e delle sezioni trasversali. Questo studio preliminare è essenziale per chiarire dove si deve arrivare alla fine del processo, stabilendo quali elementi del file FEM bisognerà ottenere.

Parallelamente all'analisi dei file FEM, un altro pilastro della metodologia è lo studio approfondito dello standard IFC. Comprendere la sua struttura, le sue entità e le relazioni che intercorrono tra gli elementi è fondamentale per una corretta mappatura. Ci si deve focalizzare sulle entità IFC pertinenti al dominio strutturale, come `IfcBeam` e `IfcColumn`, mappando le proprietà (attributes) e le relazioni (relationships) che queste entità possono avere, nonché il modo in cui è possibile rappresentare la geometria e le proprietà fisiche degli elementi.

Una volta compresi a fondo sia la struttura del file FEM di arrivo che quella dello standard IFC, il passaggio successivo consiste nel "popolamento" del file FEM. Questo significa assicurarsi che il modello IFC esistente contenga tutte le informazioni necessarie a descrivere gli elementi caratterizzanti il modello strutturale FEM, definendo anche come sono organizzate. Il modello IFC probabilmente non conterrà esplicitamente tutte le informazioni che servono per una rappresentazione completa del modello strutturale, per tanto, laddove vi siano carenze, sarà necessario intervenire arricchendo i dati con, ad esempio, l'aggiunta di formule matematiche per il calcolo di proprietà geometriche della sezione.

Il fulcro della metodologia è lo sviluppo di un applicativo dedicato alla conversione automatizzata: il codice realizzato deve fungere da "traduttore" tra lo standard IFC e il formato proprietario del software FEM di arrivo. Innanzitutto, per lo sviluppo dell'applicativo, è necessario scegliere il linguaggio di programmazione che si intende utilizzare e le librerie appropriate per i tipi di file. In seguito, basandosi sui principi estrapolati dai passaggi precedenti, si scriverà lo script affinché svolga, in ordine, le seguenti funzioni: leggere il file IFC

di input ed estrarne le informazioni necessarie, predisporre delle stringhe vuote per impostare il file FEM, associare gli elementi del modello IFC alle corrispondenti entità e proprietà del modello strutturale e, infine, riempire le stringhe del file FEM generando un file di output.

L'ultima fase della metodologia è la validazione del codice attraverso l'applicazione a diversi casi studio rappresentativi. Questa fase tende a dimostrare l'efficacia del processo di conversione, per questo i casi studio devono essere scelti per rappresentare strutture variegata, che siano differenti tra loro in termini di geometria, dimensioni, materiali e proprietà, così da assicurarsi la prova empirica della validità della metodologia proposta. Per effettuare le verifiche, si procede innanzitutto ottenendo i file IFC con i modelli dei casi studio, successivamente si esegue l'applicativo per generare il file FEM, e infine si valuta il modello risultante per verificarne la correttezza geometrica, la completezza delle entità e la presenza delle proprietà attese sia da un punto di vista visivo sia quantitativo, assicurandosi che le informazioni del modello di input corrispondano con quelle del file output. Nel caso vi siano incongruenze o errori è importante metterle in evidenza così da poter in futuro affinare il procedimento.

1 NASTRAN

La scelta del formato proprietario da studiare in questa tesi è ricaduta su NASTRAN, uno dei formati più diffusi del mondo e dallo schema dati piuttosto semplice, come si approfondirà in seguito. L'utilizzo del file NASTRAN, con estensione *.nas, vede la possibilità di applicarlo in software come MSC-Nastran, ANSYS, Autodesk Inventor Nastran, Femap, Strand7, Abaqus/CAE.

1.1 Modello dati Nastran

In generale, i dati di input necessari per descrivere e calcolare un modello analitico sono (MSC.Software Corporation, 2006):

- Geometria: un sistema di coordinate e degli elementi (es. elementi nodi, elementi travi, elementi colonne), ciascuno con le proprie dimensioni (es. lunghezza, sezione).
- Materiali: insieme di proprietà fisiche e meccaniche (es. elasticità, densità) attribuite a ciascuno degli elementi.
- Vincoli: connessioni tra elementi che possono limitarne i movimenti reciproci (es. cerniera, incastro, carrello).
- Carichi: sollecitazioni (es. forze, pressioni, temperature) applicate sulla struttura.
- Risultati: valori numerici ottenuti dai calcoli del software (es. deformazioni, spostamenti).

Pertanto, procedendo alla descrizione del modello dati di Nastran, si può affermare che queste caratteristiche saranno incluse nei dati descritti dal file. L'organizzazione dei dati risulta essere lo standard proprietario dell'omonimo software e verrà studiato di seguito.

Il modello dati del file Nastran si struttura in parti, alcune dalla funzione puramente esecutiva e una contenente i dati del file. Nell'immagine seguente è possibile osservare uno schema generale sulla struttura dei dati.

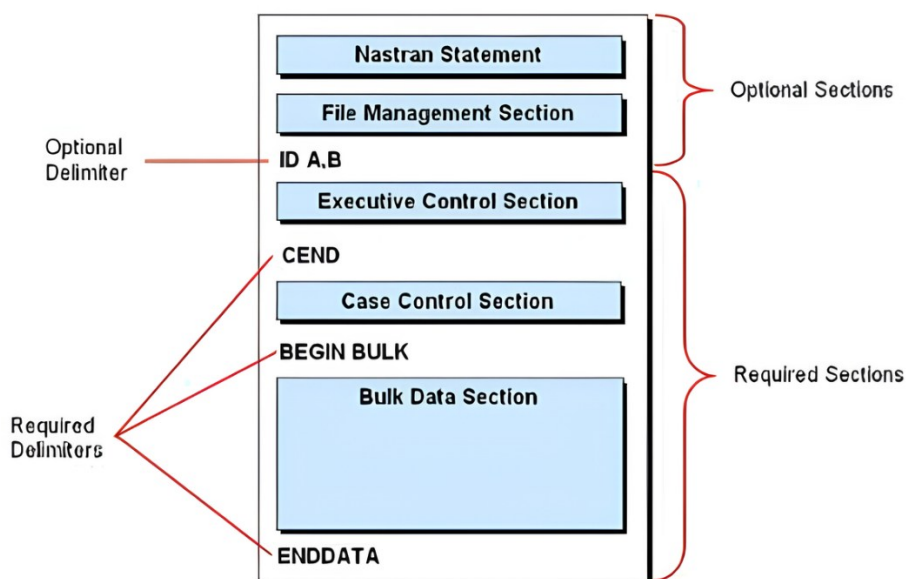


Figura 20 - Schema modello dati Nastran. Fonte: (MSC.Software Corporation, 2006)

Tra le sezioni esecutive del modello dati abbiamo *Case Control* e *Parameters*. *Case Control* è la componente esecutiva in cui si definiscono i casi di carico e si selezionano i carichi e i vincoli dai Bulk Data. I *Parameters*, invece, sono interruttori o variabili che modificano il comportamento del solutore.

Il nucleo del modello dati, però, sta nel *Bulk Data* di Nastran. In questa parte del file sono identificati tutti gli elementi e gli attributi atti a descrivere un modello analitico. Alcuni di questi sono GRID, che rappresenta i nodi, MAT, che descrive i materiali, PBAR per le travi, PLOAD identifica i carichi e SPC i vincoli. Tutte le entità che costituiscono il Bulk Data hanno una forma vettoriale (MSC.Software Corporation, 2006).

Nelle tabelle seguenti vengono riportati i principali elementi a vettore costituenti il Bulk Data di Nastran:

Tabella 2 – GRID = (GRID, ID, CP, X, Y, Z)

GRID = (GRID, ID, CP, X, Y, Z)		Descrizione del parametro
GRID	=	Nodi
ID	=	Numero di identificazione nodo

CP	=	Lasciato vuoto, sistema di coordinate di riferimento per i grids e per le X,Y,Z che seguono
X	=	Coordinata X
Y	=	Coordinata Y
Z	=	Coordinata Z

Tabella 3 - MAT1 = (MAT1, ID, E, G, NU, RHO)

MAT1 = (MAT1, ID, E, G, NU, RHO)		Descrizione del parametro
MAT1	=	Materiale
MID	=	Numero di identificazione materiale
E	=	Modulo elastico
G	=	Modulo di taglio
NU	=	Coefficiente di Poisson
RHO	=	Densità

Tabella 4 – PBEAM = (PBEAM, PID, MID, Area, I1 (zz), I2 (yy), J)

PBEAM = (PBEAM, PID, MID, Area, I1 (zz), I2 (yy), J)		Descrizione del parametro
PBEAM	=	Proprietà degli elementi trave
PID	=	Numero di identificazione della proprietà
MID	=	Numero di identificazione materiale
Area	=	Area
I1(zz)	=	Modulo d'inerzia
I2(yy)	=	Modulo d'inerzia
J	=	Modulo di rigidezza torsionale - Default = 0.0 (Real ≥ 0.0)

Tabella 5 - CBEAM = (CBEAM, EID, PID, GA, GB, X1, X2, X3)

CBEAM = (CBEAM, EID, PID, GA, GB, X1, X2, X3)		Descrizione del parametro
CBEAM	=	Elemento trave
EID	=	Numero di identificazione elemento
PID	=	Numero di identificazione proprietà fisiche
GA	=	Numero identificativo del nodo 1
GB	=	Numero identificativo del nodo 2
X1	=	Vettore direzionale da nodo 1 a 2 (in x)
X2	=	Vettore direzionale da nodo 1 a 2 (in y)
X3	=	Vettore direzionale da nodo 1 a 2 (in z)

Tabella 6 - PLOAD = (PLOAD1, SID, EID, TYPE, SCALE, X1, P1, X2, P2)

PLOAD1 = (PLOAD1, SID, EID, TYPE, SCALE, X1, P1, X2, P2)		Descrizione del parametro
PLOAD1	=	Carico di pressione statica
SID	=	Numero di identificazione del set di carico
EID	=	Numero di identificazione dell'elemento
TYPE	=	Tipo di caricamento, una delle seguenti variabili carattere: FX, FY, FZ, FXE, FYE, FZE, MX, MY, MZ, MXE, MYE e MZE

SCALE	=	Determina il fattore di scala per X1, X2. Deve essere una delle seguenti variabili carattere: LE, FR, LEPR o FRPR
X1	=	Distanza lungo l'asse dell'elemento dall'estremità A ($X2 \geq X1 \geq 0,0$)
P1	=	Fattore di carico nella posizione X1
X2	=	Distanza lungo l'asse dell'elemento dall'estremità A ($X2 \geq X1 \geq 0,0$)
P2	=	Fattore di carico nella posizione X2

Tabella 7 - SPC = (SPC1, SID, C, Gi)

SPC1 = (SPC1, SID, C, Gi)		Descrizione del parametro
SPC1	=	Insiemi di vincoli a punto singolo
SID	=	Numero di identificazione del set di vincoli a punto singolo
C	=	Numero del componente della coordinata globale. (È possibile inserire fino a sei cifre univoche nel campo, senza spazi vuoti incorporati.)
Gi	=	Numero/i di identificazione del punto della griglia

1.2 Strand7

1.2.1 Scelta di Strand7

Per la lettura del file Nastran è stato individuato Strand7 come software di riferimento: presente sul mercato italiano dal 1994, gode oggi di una consolidata popolarità nel settore dell'ingegneria strutturale, risultando uno dei software per analisi FEM più affidabili e utilizzati nel nostro territorio.

Gli ambiti in cui viene preferito Strand7 sono quelli legati al settore civile, sia per analisi statiche che per la valutazione sismica di strutture complesse. Anche a livello universitario, questo programma viene insegnato agli studenti come strumento digitale preferenziale per l'analisi FEM.

1.2.2 Il software Strand7

Strand7 è uno dei software di analisi agli elementi finiti più diffusi e utilizzati nei settori dell'edilizia e dell'ingegneria meccanica in tutto il mondo. Nato negli anni 90, questo programma permette di eseguire analisi statiche lineari e non lineari, di individuare le deformazioni di una struttura analitica, risolve problemi di dinamica individuando le risposte spettrali e armoniche e tiene conto del trasferimento di calore lineare e non lineare in stato stazionario e transitorio.

Le API di Strand7, inoltre, offrono la possibilità di interagire con il programma e il database dei modelli bypassando l'ambiente interattivo di Strand7, con lo scopo di svolgere funzioni specifiche ad hoc per il proprio progetto (Strand7, 2026).



Figura 21 - Logo di Strand7. Fonte: (Strand7, 2026)

1.2.3 File compatibili con Strand7

I formati compatibili con questo software sono:

Per l'esportazione

- Straus7 text file (*.txt)
- IGES file (*.igs, *.iges)
- STEP file (*.stp, *.step)
- AutoCAD DXF file (*.dxf)
- Stereo-Lithography file (*.stl, *.stb)
- Windows bitmap file (*.bmp)

- JPEG file (*.jpg)
- MSC/Nastran file (*.dat, *.nas)
- ANSYS text file (*.dat, *.cdb)

Per l'importazione

- IGES file (*.igs, *.iges)
- ACIS file (*.sat)
- STEP file (*.stp, *.step)
- Straus7 text file (*.txt)
- Straus6 binary file (*.dir)
- Straus6 text file (*.txt, *.asc)
- AutoCAD DXF file (*.dxf)
- Stereo-Lithography file (*.stl, *.stb)
- MSC/Nastran file (*.dat, *.nas, *.bdf)
- ANSYS text file (*.dat, *.cdb)
- STAAD-Pro file (*.std)
- SAP2000 file (*.s2k, *.s2k)

Il modello dati di tutti i formati contempla esclusivamente la geometria della struttura, ad esclusione di ANSYS e NASTRAN, che sono gli unici ad avere nel loro modello dati informazioni relative alle proprietà degli elementi, ai materiali, ai carichi e ai vincoli. Per questa ragione nella proposta del caso studio ci si è soffermati sul mappaggio di uno di questi due tipi di file: NASTRAN.

2 STUDIO DELLO STANDARD IFC

2.1 IFCCOLUMN

In questo paragrafo si approfondirà l'architettura dello standard IFC4X3_ADD2, identico a quello dell'IFC4 per quanto riguarda le entità riportate in questo studio, con cui viene descritta l'entità IfcColumn: lo schema grafico riportato di seguito vuole chiarificare le relazioni tra gli elementi dello standard e identificare i termini per le definizioni approfondite di seguito. Per visualizzare meglio lo schema si rimanda all'Appendice 1.

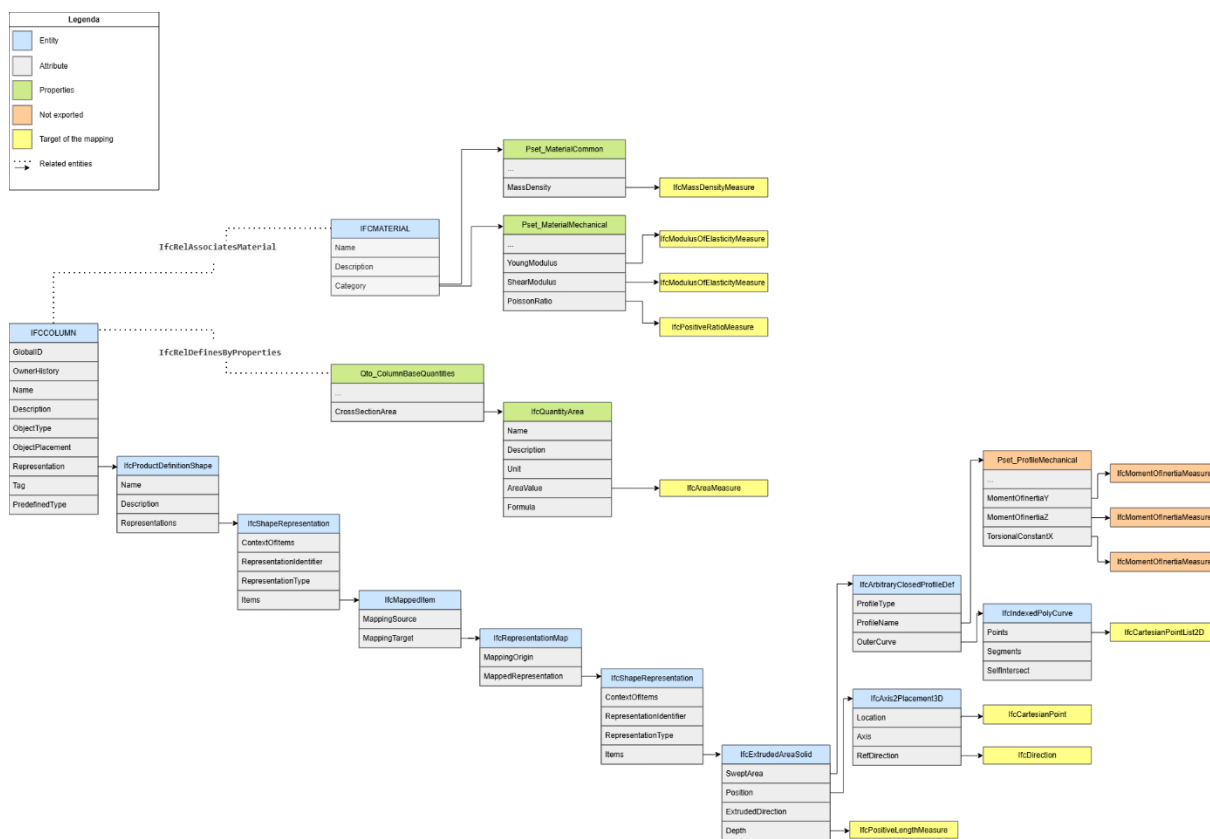


Figura 22 - Architettura IfcColumn nello Standard 4x3. Fonte: autore.

2.1.1 IFCCOLUMN

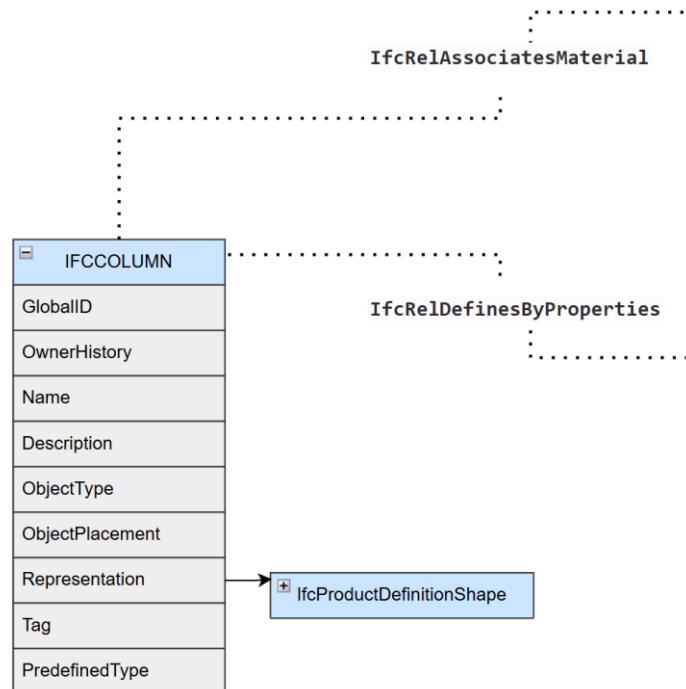


Figura 23 - Zoom architettura IfcColumn in IFC4X3_ADD2. Fonte: autore.

Innanzitutto, bisogna definire IfcColumn e accennare il significato dei suoi attributi, soffermandoci specificatamente su *Representation*, in quanto ci servirà per sviluppare la geometria dell'elemento colonna.

Nel sito ufficiale di BuildingSMART, un elemento IfcColumn viene descritto come un elemento strutturale o architettonico verticale che spesso è allineato con un'intersezione della griglia strutturale. Nella maggior parte dei casi rappresenta un elemento strutturale verticale, o quasi verticale, che trasmette, tramite compressione, il peso della struttura sovrastante agli altri elementi strutturali sottostanti. Può anche rappresentare tale elemento da un punto di vista architettonico, nel qual caso può sarebbe un elemento non portante.

Gli attributi che appartengono all'entità IfcColumn sono quelli illustrati nella tabella sottostante.

Tabella 8 - Attributi IfcColumn. Fonte: (buildingSMART International, 2026)

Attributo	Descrizione
GlobalID	Assegnazione di un identificatore univoco globale all'interno dell'intero mondo del software.

OwnerHistory	Assegnazione delle informazioni sulla proprietà attuale di tale oggetto, tra cui l'attore proprietario, l'applicazione, l'identificazione locale e le informazioni acquisite sulle recenti modifiche dell'oggetto.
Name	Un nome facoltativo da utilizzare da parte dei sistemi software o degli utenti partecipanti. Per alcuni sottotipi di IfcRoot potrebbe essere richiesto l'inserimento dell'attributo Name. Questo sarebbe imposto da una regola where.
Description	Descrizione facoltativa, fornita per scambiare commenti informativi.
ObjectType	Il tipo denota un tipo particolare che definisce ulteriormente l'oggetto. L'uso deve essere stabilito a livello dei sottotipi istanziabili. In particolare, contiene il tipo definito dall'utente, se l'enumerazione dell'attributo PredefinedType è impostata su USERDEFINED o quando l'entità concreta istanziata non ha un attributo PredefinedType. Quest'ultimo caso si verifica in alcune classi foglia eccezionali e quando si istanzia direttamente IfcBuiltElement.
ObjectPlacement	Questo definisce il sistema di coordinate dell'oggetto e il posizionamento del prodotto nello spazio. Il posizionamento può essere assoluto (relativo al sistema di coordinate globale), relativo (relativo al posizionamento dell'oggetto di un altro prodotto) o vincolato (ad esempio, relativo agli assi della griglia o a un elemento di posizionamento lineare). Il tipo di posizionamento è determinato dai vari sottotipi di IfcObjectPlacement. È necessario specificare un posizionamento dell'oggetto se è presente una rappresentazione.
Representation	Riferimento alle rappresentazioni del prodotto, che possono essere una rappresentazione (IfcProductRepresentation) o un caso speciale di rappresentazione di forma (IfcProductDefinitionShape). La forma della definizione del prodotto fornisce più rappresentazioni geometriche della proprietà di forma dell'oggetto all'interno dello stesso sistema di coordinate dell'oggetto, definito dal posizionamento dell'oggetto.
Tag	Identificatore del tag (o etichetta) in una specifica istanza di un prodotto, ad esempio il numero di serie o il numero di posizione. È l'identificatore a livello di occorrenza.
PredefinedType	Tipo generico predefinito per una colonna specificata in un'enumerazione. Potrebbe esserci un set di proprietà specificato specificamente per i tipi predefiniti.

In secondo luogo, dallo schema grafico presentato all'inizio di questa sezione, si osserva che da IfcColumn partono tre frecce, le quali lo collegano ai seguenti concetti:

- **IfcRelAssociatesMaterial**: è una relazione oggettivata tra una definizione di materiale e gli elementi o i tipi di elementi a cui si applica questa definizione di materiale.
- **IfcRelDefinesByProperties**: definisce le relazioni tra le definizioni dei set di proprietà e gli oggetti. Le proprietà sono aggregate in set di proprietà. I set di proprietà possono essere assegnati direttamente agli *occurrence objects* utilizzando questa relazione, oppure assegnati a un tipo di oggetto e assegnati tramite quel *type* agli *occurrence objects*.

IfcRelDefinesByProperties è una relazione N-a-N, poiché consente l'assegnazione di uno o più set di proprietà a uno o più oggetti. Tali oggetti condividono quindi la stessa definizione di proprietà.

- ***IfcProductDefinitionShape***: è un'entità che definisce tutte le informazioni rilevanti per la forma di un *IfcProduct*. Consente molteplici rappresentazioni geometriche dello stesso prodotto. Le informazioni rilevanti per la forma includono :
 - la rappresentazione della forma, compresi gli elementi di rappresentazione geometrica (per solidi 3D, annotazioni 2D, ecc.) e:
 - informazioni di presentazione associate (colore della linea, tipo di linea, proprietà di rendering della superficie)
 - assegnazione ai livelli di presentazione (livelli CAD per il controllo della visibilità)
 - gli elementi di rappresentazione topologica per i sistemi di connettività (rappresentazioni di vertici, bordi, facce) che possono includere elementi di rappresentazione geometrica (punti di vertice, curve di bordo, superfici di facce)

2.1.2 IFCMATERIAL

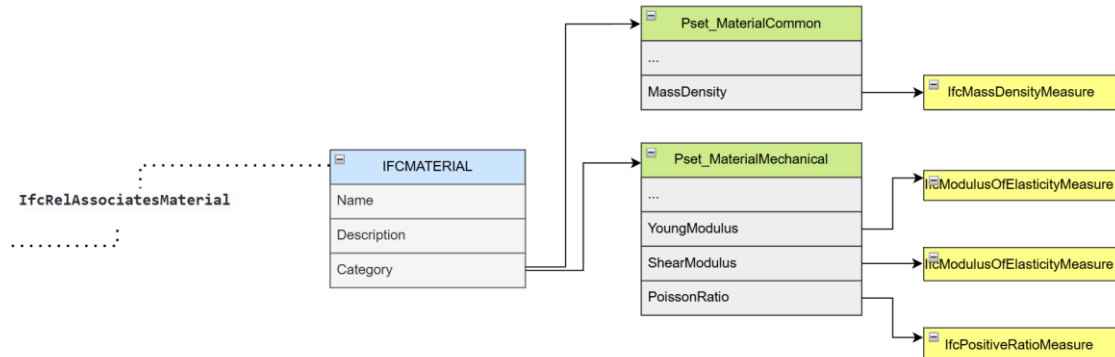


Figura 24 - Zoom architettura *IfcMaterial* in IFC4X3_ADD2. Fonte: autore.

A partire dalla relazione oggettivata *IfcRelAssociatesMaterial*, siamo in grado di identificare le caratteristiche del materiale associato all'elemento colonna del file IFC. In particolare, come si osserva dall'immagine, abbiamo l'entità *IfcMaterial* che si sviluppa in tre attributi: all'attributo *Category (HasRepresentation)*, si collegano due set di proprietà utili a identificare le caratteristiche meccaniche del materiale come Modulo di Elasticità, Modulo di Taglio e Coefficiente di Poisson, Densità, così da poterli utilizzare in fase di analisi strutturale.

2.1.3 IFCQUANTITYAREA

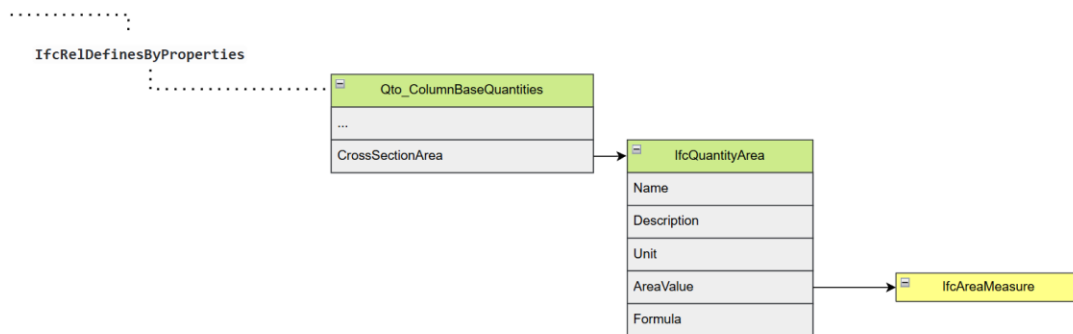


Figura 25 - Zoom architettura IfcQuantityArea in IFC4X3_ADD2. Fonte: autore.

Grazie alla relazione oggettivata *IfcRelDefinesByProperties*, siamo in grado di connettere *IfcColumn* con il set di proprietà *Qto_ColumnBaseQuantities*. Le proprietà appartenenti a questo set sono quelle nella tabella seguente. Ai nostri scopi la proprietà utile è quella inerente alla dimensione dell'area della sezione trasversale, detta in inglese *CrossSectionArea*. Il dato che ne deriva viene descritto dall'attributo *AreaValue* di *IfcQuantityArea*.

Tabella 9 - Set di proprietà *Qto_ColumnBaseQuantities*. Fonte: (buildingSMART International, 2026)

Proprietà	Data Type	Descrizione
Length	IfcQuantityLength	La lunghezza dell'oggetto. Senza considerare eventuali ritagli o altre caratteristiche di lavorazione.
CrossSectionArea	IfcQuantityArea	Area totale della sezione trasversale (o profilo) dell'oggetto.
OuterSurfaceArea	IfcQuantityArea	Area totale delle superfici dell'oggetto (senza considerare le aree delle estremità), normalmente generata come perimetro * lunghezza in caso di estrusioni.
GrossSurfaceArea	IfcQuantityArea	Area lorda totale dell'oggetto, normalmente generata come perimetro * lunghezza + 2 * area della sezione trasversale. È la somma di OuterSurfaceArea + (2 x CrossSectionArea) e deve essere specificata solo se OuterSurfaceArea e CrossSectionArea non possono essere determinate separatamente.
NetSurfaceArea	IfcQuantityArea	Area superficiale netta dell'oggetto, normalmente generata come perimetro * lunghezza + 2 * area della sezione trasversale tenendo conto delle possibili caratteristiche di lavorazione (ritagli, ecc.) o aperture e rientranze.
GrossVolume	IfcQuantityVolume	Volume lordo totale dell'oggetto. Aperture, rientranze, oggetti chiusi e sporgenze non sono presi in considerazione.

NetVolume	IfcQuantityVolume	Volume netto totale dell'oggetto, tenendo conto delle possibili caratteristiche di lavorazione (ritagli, ecc.) o aperture e rientranze.
GrossWeight	IfcQuantityWeight	Peso lordo totale dell'oggetto senza parti aggiuntive e senza tenere conto di possibili caratteristiche di lavorazione (ritagli, ecc.) o aperture e rientranze.
NetWeight	IfcQuantityWeight	Peso netto totale dell'oggetto senza parti aggiuntive, tenendo conto di possibili caratteristiche di lavorazione (ritagli, ecc.) o aperture e rientranze.

2.1.4 IFCPRODUCTDEFINITIONSHAPE

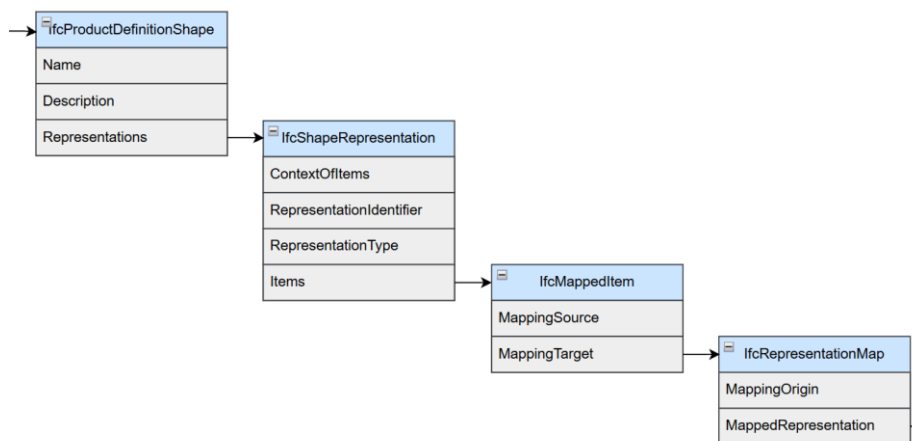


Figura 26 - Zoom architettura IfcProductDefinitionShape in IFC4X3_ADD2. Fonte: autore.

Entrando nel vivo della rappresentazione geometrica dell'elemento *IfcColumn*, il percorso da seguire nello schema architettonico di IFC4X3_ADD2 è quello rappresentato in figura: dall'attributo *Representation* di *IfcColumn* si passa all'entità *IfcProductDefinitionShape*, del tipo *IfcProductRepresentation*. L'entità *IfcProductDefinitionShape* ha tre attributi e in quello *Representations* si può trovare l'entità *IfcShapeRepresentation*, del tipo *IfcRepresentation*. Tra gli *Items* di *IfcShapeRepresentation*, *IfcMappedItem* ci porta ad *IfcRepresentationMap*, dal cui attributo *Mapped Representation* si potrà sviluppare la descrizione geometrica dell'elemento *IfcColumn* come illustrato nelle sezioni successive. Per chiarezza l'autore ha creato una tabella riassuntiva delle entità coinvolte in questo passaggio, associandone il tipo e la descrizione.

Tabella 10 - Elenco entità coinvolte nello schema *IfcColumn* IFC4X3_ADD2. Fonte: (buildingSMART International, 2026)

Entità	Tipo	Descrizione
IfcProductDefinitionShape	IfcProductRepresentation	IfcProductDefinitionShape definisce tutte le informazioni rilevanti per la forma di un IfcProduct. Consente molteplici

		<p>rappresentazioni geometriche dello stesso prodotto. Le informazioni rilevanti per la forma includono :</p> <ul style="list-style-type: none"> o la rappresentazione della forma, compresi gli elementi di rappresentazione geometrica (per solidi 3D, annotazioni 2D, ecc.) e: o informazioni di presentazione associate (colore della linea, tipo di linea, proprietà di rendering della superficie) o assegnazione ai livelli di presentazione (livelli CAD per il controllo della visibilità) o gli elementi di rappresentazione topologica per i sistemi di connettività (rappresentazioni di vertici, bordi, facce) che possono includere elementi di rappresentazione geometrica (punti di vertice, curve di bordo, superfici di facce)
IfcShapeRepresentation	IfcRepresentation	IfcShapeRepresentation rappresenta il concetto di una particolare rappresentazione geometrica di un prodotto o di un componente di prodotto all'interno di uno specifico contesto di rappresentazione geometrica.
IfcMappedItem	IfcRepresentationItem	IfcMappedItem è l'istanza inserita di una definizione sorgente (da confrontare con una definizione di blocco/cella condivisa/macro). L'istanza viene inserita applicando un operatore di trasformazione cartesiana come MappingTarget .
IfcRepresentationMap	IfcRepresentationItem	Una mappa di rappresentazione è l'identificazione di una rappresentazione e di un elemento di rappresentazione in tale rappresentazione ai fini della mappatura. L'elemento di rappresentazione definisce l'origine della mappatura. La mappa di rappresentazione viene utilizzata come origine di una mappatura da parte di un elemento mappato.

2.1.5 IFC SHAPEREPRESENTATION

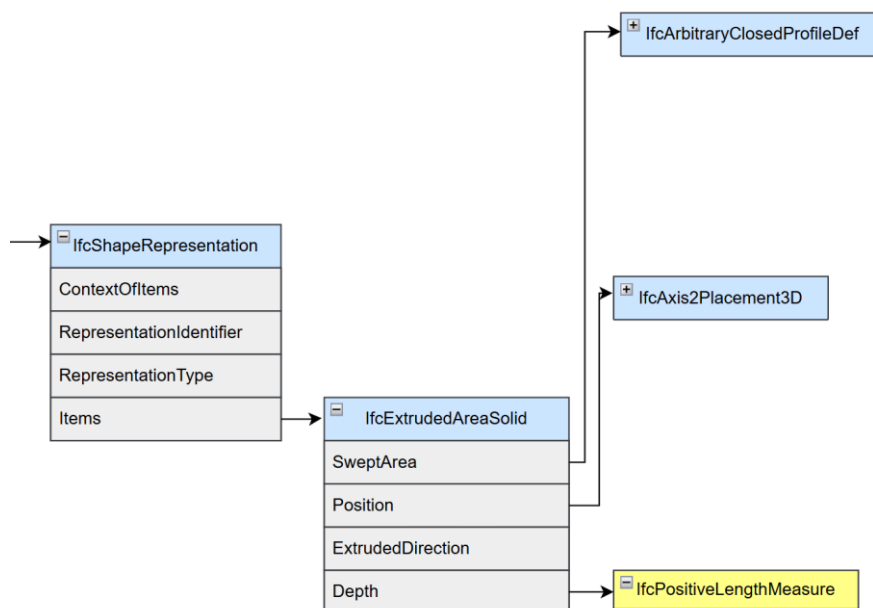


Figura 27 - Zoom architettura IFCShapeRepresentation in IFC4X3_ADD2. Fonte: autore.

Come si può vedere dall'immagine, l'articolato percorso che bisogna seguire per giungere alla descrizione della forma della colonna passa di nuovo per *IfcShapeRepresentation* a partire dall'attributo *MappedRepresentation* dell'entità *IfcRepresentationMap*. Questa volta, tuttavia, dal suo attributo *Items* si ottiene *IfcExtrudedAreaSolid*, entità fondamentale per comprendere come viene modellata e descritta in Revit e in IFC una colonna (o una trave).

IfcExtrudedAreaSolid, del tipo *IfcRepresentationItem*, viene definito estrapolando una sezione trasversale fornita da una definizione di profilo. La direzione dell'estrusione è data dall'attributo *ExtrudedDirection* e la lunghezza dell'estrusione è data dall'attributo *Depth*. Se l'area planare presenta confini interni (fori definiti), tali fori devono essere estrusi nei fori del solido.

Il solido risultante viene posizionato tramite *IfcSweptAreaSolid*. Posizione relativa al sistema di coordinate dell'oggetto. Se specificato, consente di riposizionare il solido estruso. Se non specificato, viene utilizzato di default il sistema di coordinate dell'oggetto corrente. *ExtrudedDirection* viene specificato all'interno del sistema di coordinate di posizione definito da *IfcSweptAreaSolid.Position*. La direzione di estrusione può essere qualsiasi direzione che non sia perpendicolare all'asse z del sistema di coordinate di posizione.

Si riporta di seguito un'immagine illustrativa presa dal sito BuildingSMART, in cui è possibile osservare graficamente quanto accennato in questo testo.

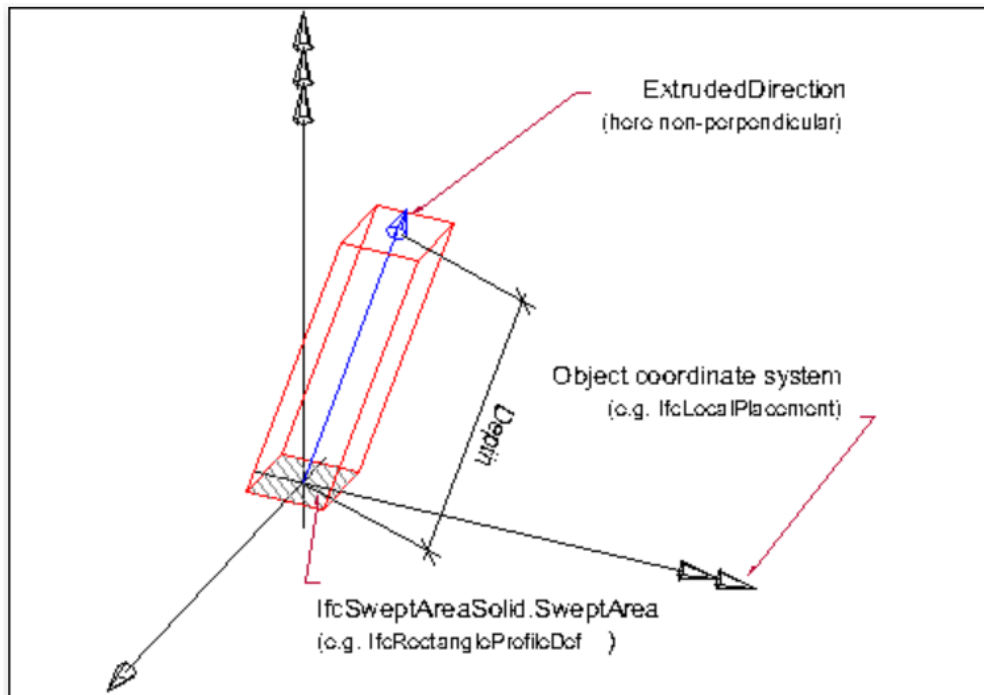


Figura 28 - IfcExtrudedAreaSolid. Fonte: (buildingSMART International, 2026)

Per comprendere al meglio *IfcExtrudedAreaSolid*, si è scelto di riportare anche una tabella con gli attributi di questa entità, approfondendone il significato e i valori associati.

Tabella 11 - Attributi IfcExtrudedAreaSolid. Fonte: (buildingSMART International, 2026)

Attributo	Descrizione
SweptArea	Superficie che definisce l'area da spazzare. Viene fornita come definizione di profilo all'interno del piano xy del sistema di coordinate di posizione.
Position	Sistema di coordinate di posizione per il solido risultante dall'operazione di sweep. Il sistema di coordinate di posizione consente il riposizionamento del solido. Se non specificato, il solido rimane nella posizione determinata dalla sezione trasversale o dalla direttrice utilizzata per l'operazione di sweep.
ExtrudedDirection	La direzione in cui deve essere spazzata la superficie, fornita da <i>SweptArea</i> . <i>ExtrudedDirection</i> non deve essere perpendicolare all'asse z locale.
Depth	La distanza lungo la quale la superficie deve essere spazzata lungo <i>ExtrudedDirection</i> .

2.1.6 IFCARBITRARYCLOSEDPROFILEDEF

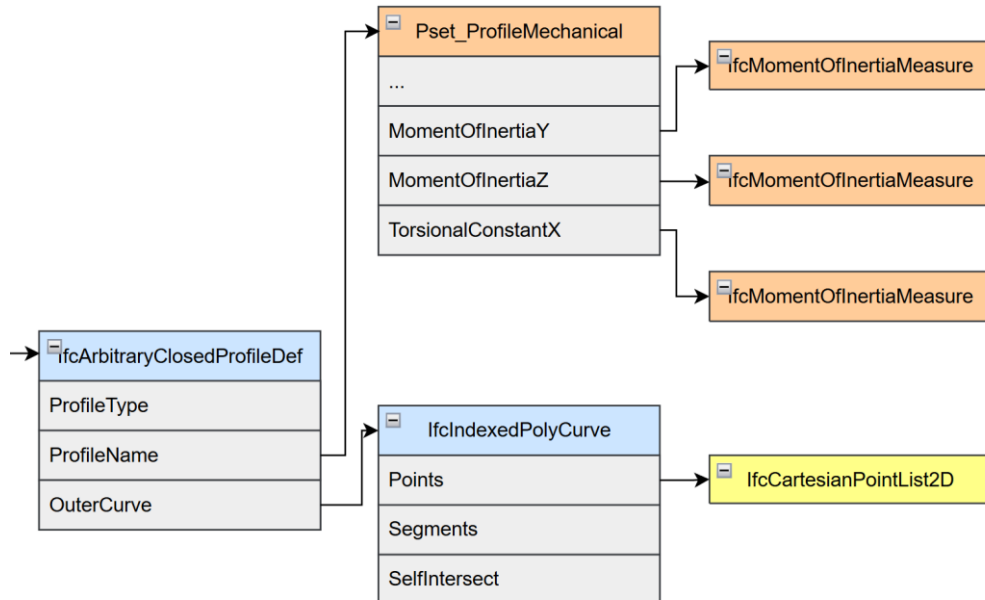


Figura 29 - Zoom architettura IfcArbitraryClosedProfileDef in IFC4X3_ADD2. Fonte: autore.

Dall'attributo *SweptArea* di *IfcExtrudedAreaSolid* si arriva all'entità *IfcArbitraryClosedProfileDef*, del tipo *IfcProfileDef*. *IfcProfileDef* è il supertipo di tutte le definizioni di profili standard e arbitrari all'interno di IFC. Viene utilizzato per definire un set standard di profili di sezione di uso comune mediante i loro parametri o mediante la loro geometria di curva esplicita. I profili arbitrari, come quello individuato in questo schema in quanto declinabile per tutte le geometrie, sono sezioni trasversali definite da un contorno esterno come curva delimitata, che può anche includere fori, definiti da contorni interni.

Tra i suoi attributi l'*OuterCurve* è quello che consente di arrivare alla *IfcIndexedPolyCurve*, del tipo *IfcCurve*. *IfcIndexedPolyCurve* è una curva delimitata con solo segmenti di arco lineare e circolare definiti da un elenco di punti cartesiani e da un elenco facoltativo di segmenti, che fornisce indici nell'elenco di punti cartesiani. Nel caso in cui l'elenco di segmenti non sia fornito, tutti i punti in *IfcCartesianPointList* sono collegati da segmenti di linea retta nell'ordine in cui appaiono in *IfcCartesianPointList*.

Nel caso in cui venga fornito l'elenco dei Segmenti, questo viene interpretato come segue:

- Segmento di tipo *IfcLineIndex*: il primo indice è il punto iniziale del segmento di polilinea, l'ultimo indice è il punto finale del segmento di polilinea. Se sono inclusi più di due indici, tutti gli indici intermedi definiscono i punti intermedi della polilinea connessi nell'ordine di comparsa degli indici;

- Segmento di tipo *IfcArcIndex*: il primo indice è il punto iniziale dell'arco circolare, il secondo indice è un punto sull'arco, il terzo indice è il punto finale dell'arco circolare. I tre punti non devono essere colineari. Nel caso in cui questa proposizione informale non venga mantenuta, il segmento di arco deve essere trattato come un segmento di polilinea.

Attraverso l'attributo *ProfileName* di *IfcArbitraryClosedProfileDef* (*HasProperties*), invece, è possibile accedere, per lo meno a livello teorico, al set di proprietà che descrivono le caratteristiche meccaniche della sezione quali Momenti di Inerzia in zz e in yy e modulo di rigidità torsionale. Si anticipa già ora che queste proprietà in fase di esportazione vengono perse, tuttavia si mette in evidenza che, conoscendo i lati della sezione, è fattibile calcolarne i valori numerici a partire da formule note.

2.1.7 IFCAXIS2PLACEMENT3D

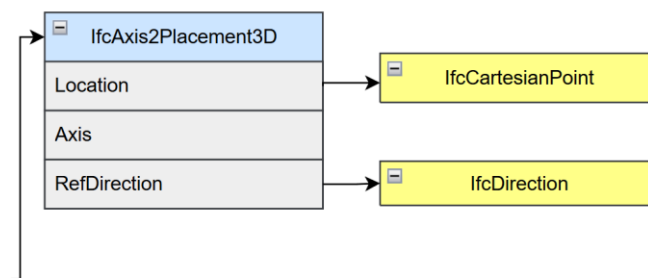


Figura 30 - Zoom architettura *IfcAxis2Placement3D* in IFC4X3_ADD2. Fonte: autore.

IfcAxis2Placement3D fornisce dati su posizione e orientamenti per posizionare gli elementi in uno spazio tridimensionale. L'attributo *Axis* definisce la direzione Z, *RefDirection* la direzione X. La direzione Y è derivata. Se i valori degli attributi per *Axis* e *RefDirection* non vengono specificati, il posizionamento predefinito è P[1] (asse x) come [1.,0.,0.], P[2] (asse y) come [0.,1.,0.] e P[3] (asse z) come [0.,0.,1.].

Gli attributi di questa entità vengono riassunti nella tabella seguente:

Tabella 12 - Attributi *IfcAxis2Placement3D*. Fonte: (buildingSMART International, 2026)

Attributo	Descrizione
Location	Posizione geometrica di un punto di riferimento, ad esempio il centro di un cerchio, dell'elemento da localizzare.
Axis	La direzione esatta dell'asse Z locale.
RefDirection	Direzione utilizzata per determinare la direzione dell'asse X locale. Se necessario, viene apportata una modifica per mantenere l'ortogonalità rispetto

alla direzione dell'asse. Se Axis e/o RefDirection vengono omessi, queste direzioni vengono ricavate dal sistema di coordinate geometriche.

2.2 IFCBEAM

Analogamente allo schema riportato per *IfcColumn*, si inserisce in questo paragrafo una descrizione dell'architettura dell'elemento *IfcBeam* descritto dallo standard IFC4X3_ADD2 come rappresentato in figura. Si osserva che gli elementi e le relazioni utili alla descrizione dell'entità in questione sono identici a quelli dell'*IfcColumn*, e per questa ragione per le definizioni degli attributi citati si rimanda al capitolo precedente. Per visualizzare meglio lo schema si rimanda all'Appendice 1.

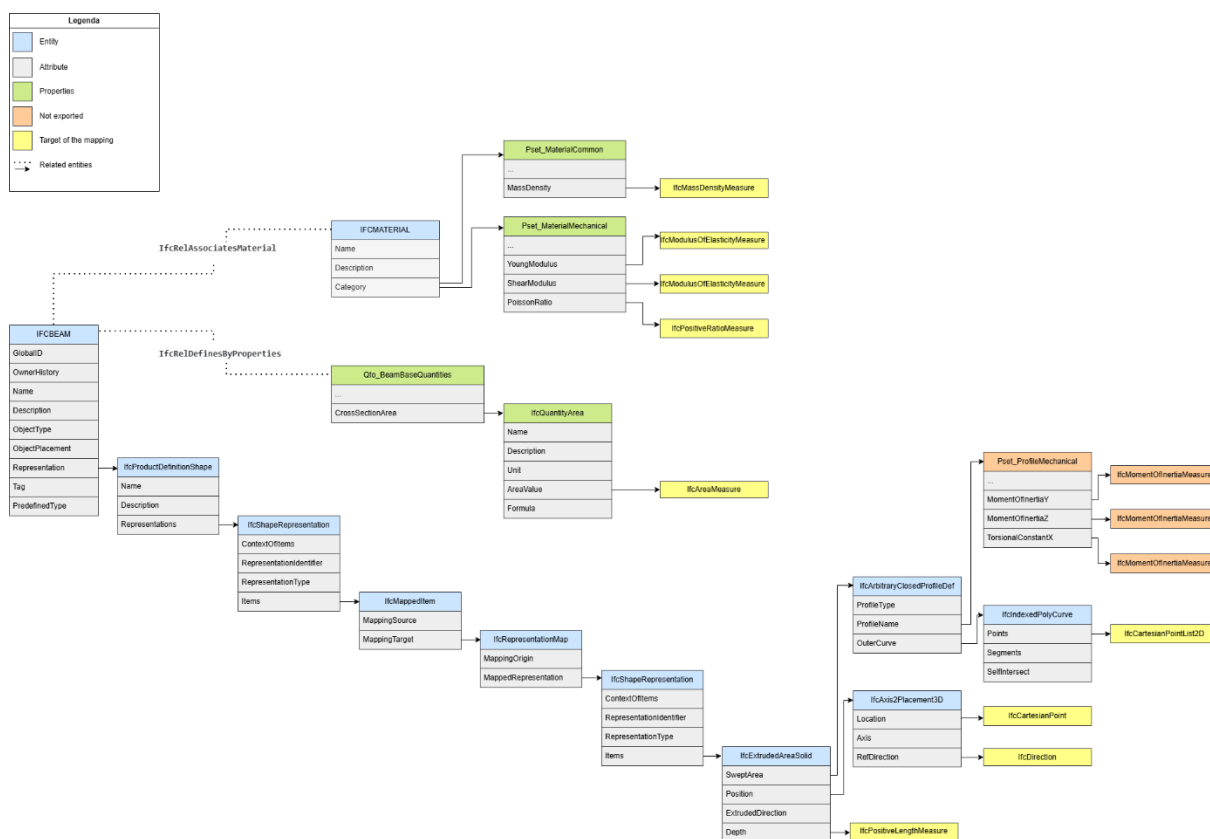


Figura 31 - Architettura *IfcBeam* nello Standard 4x3. Fonte: autore.

Le uniche differenze che si osservano rispetto agli elementi di *IfcColumn* sono relative alla nomenclatura dell'entità principale, che infatti è *IfcBeam*, e al set di proprietà *Qto_BeamBaseQuantities*.

3 POPOLAMENTO POPOLAMENTO DEL FILE NASTRAN

Entriamo ora nel vivo della mappatura: partendo dagli elementi di un file Nastran .nas individuiamo nell'architettura dello Standard IFC4X3_ADD2 i percorsi che ci consentono di riempire i vettori Nastran con le informazioni analitiche necessarie. Per completezza si riporta un esempio di file Nastran, in modo tale da averne chiara la sua struttura.

3.1 GRID – PERCORSI

I nodi di Nastran vengono identificati con il vettore GRID, come descritto al Capitolo Primo. Per ciascun parametro si ricava il percorso corrispondente in IFC così da poter riempire i valori numerici del vettore in maniera completa e sensata.

3.1.1 GRID – GRID

Il primo valore del vettore GRID è l'identificativo GRID, il quale esprime il tipo di oggetto che viene modellato nel software di analisi Fem. Tale valore deve rimanere espresso come GRID.

3.1.2 GRID – ID

Il numero di identificazione nodo è un valore progressivo che è facilmente impostabile come $ID = (n+1)$, sapendo che n corrisponde al valore numerico dell'ID del nodo precedente.

3.1.3 GRID – CP

Tendenzialmente lasciato vuoto, è il sistema di coordinate di riferimento per i grids e per le X,Y,Z che seguono. Questo valore non viene espresso e viene per tanto considerato in automatico il sistema di coordinate standard del file.

3.1.4 GRID – X,Y,Z

Per identificare le coordinate X, Y, Z del nodo in questione è necessario fare riferimento al file IFC di partenza: considerando la descrizione geometrica dell'elemento, bisogna ricondursi all'entità *IfcExtrudedAreaSolid*, seguendo il percorso presentato nell'immagine seguente.

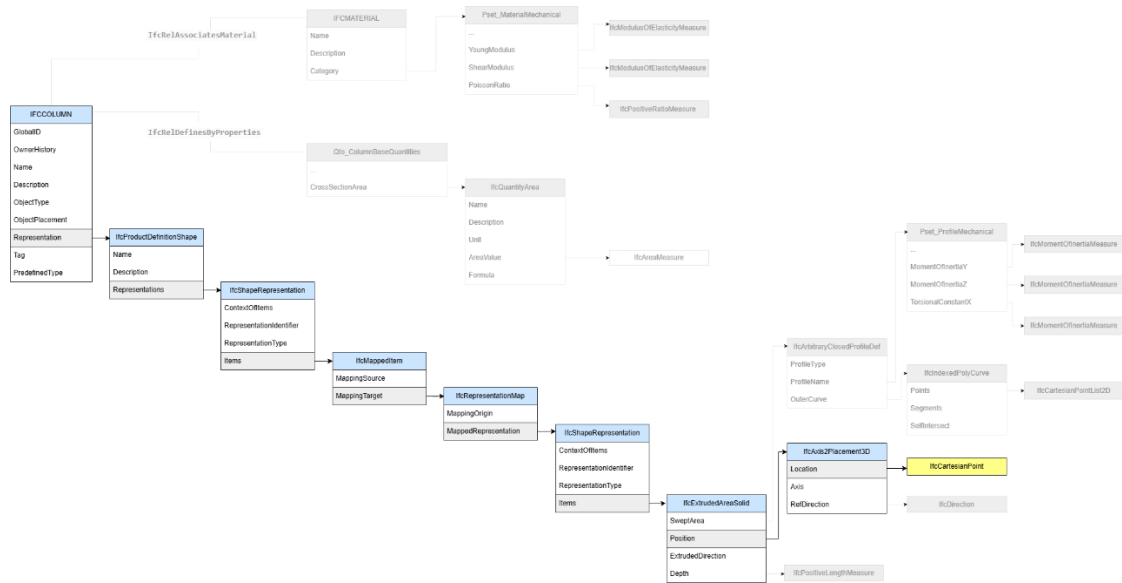


Figura 32 - Percorso GRID - X, Y, Z. Fonte: autore.

A partire da *IfcExtrudedAreaSolid*, considerando il significato degli attributi come presentato al paragrafo precedente, la posizione del nodo di origine della colonna (o trave) sarà il punto definito in IFC, per l'entità colonna, come *Location*. A tale attributo di *IfcAxis2Placement* corrisponde un vettore numerico *IfcCartesianPoint*, costituito da tre coordinate: X, Y, Z del nodo.

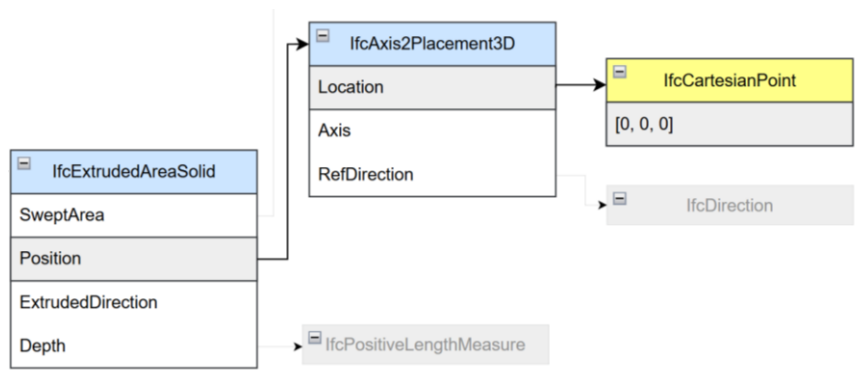


Figura 33 - GRID - IfcCartesianPoint. Fonte: autore.

3.2 MAT1 – PERCORSI

Attraverso *IfcRelAssociatesMaterial* siamo in grado di associare alla colonna il tipo di materiale con tutte le sue caratteristiche fisiche e meccaniche, per tanto andremo a prendere da qui i valori da inserire nel vettore MAT del file Nastran.

3.2.1 MAT1 – MAT1

Il primo valore del vettore è espresso come MAT1, indicando che tutti i valori seguenti identificano un materiale da associare all’entità Nastran considerata. Per tanto questo primo parametro del vettore andrà riempito sempre con “MAT1”.

3.2.2 MAT1 – MID

Nel caso in cui in un solo file siano presenti diversi materiali, si utilizzerà il numero di identificazione MID per differenziarli tra loro: MID è un valore progressivo calcolabile con la formula $MID = (n+1)$, dove n è il valore numerico pari al MID precedente.

3.2.3 MAT1 – E, G, NU, RHO

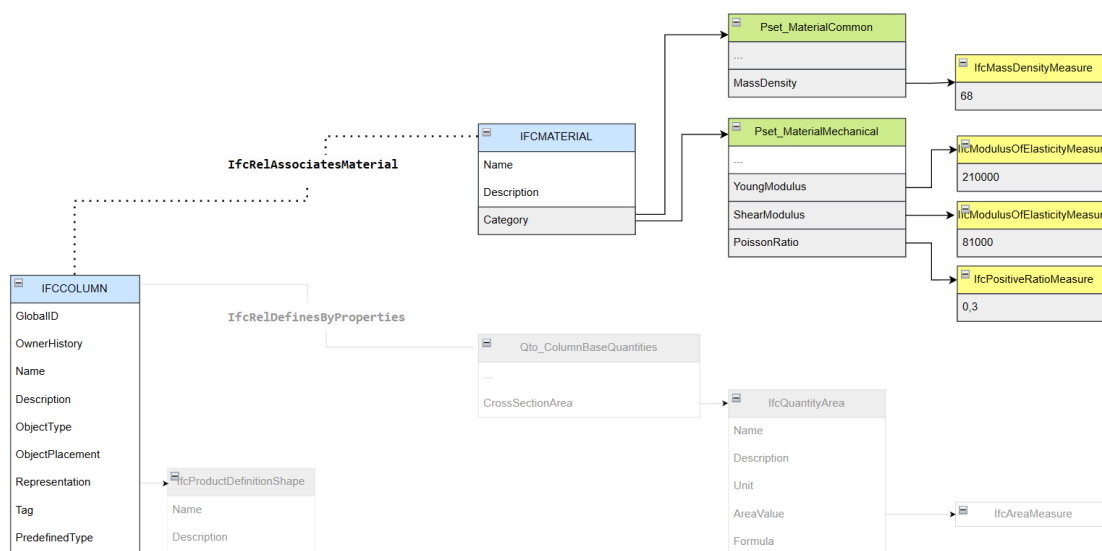


Figura 34 - MAT1 - IfcMaterial. Fonte: autore.

Per tutti i valori relativi agli aspetti fisici e meccanici del materiale, bisogna ricondursi ai set di proprietà Ifc: *Pset_MaterialCommon* per la Densità, descritta in Nastran come RHO, e *Pset_MaterialMechanical* per Modulo di Elasticità, di Taglio e Coefficiente di Poisson, descritti in Nastran rispettivamente come E, G, NU.

3.3 PBEAM – PERCORSI

3.3.1 PBEAM – PBEAM

Il primo parametro del vettore PBEAM è sempre “PBEAM” in quanto in questo modo viene identificato il tipo di oggetto che si sta descrivendo: PBEAM definisce le proprietà fisiche delle travi di Nastran, come le caratteristiche geometriche della sezione e il tipo di materiale.

3.3.2 PBEAM – PID

PID è il valore numerico identificativo della proprietà di Nastran e, per ogni PBEAM, si avrà un $PID = (n + 1)$, dove con n si intende il valore numerico associato alla PBEAM precedente.

3.3.3 PBEAM – MID

MID, il numero identificativo del materiale, viene richiamato all'interno del vettore PBEAM per associare alla trave di Nastran il materiale corretto, comprendendo le caratteristiche fisiche e meccaniche di questo. In tale maniera il software FEM sarà in grado di calcolare la resistenza della trave tenendo conto sia delle caratteristiche geometriche, sia di quelle meccaniche associate al materiale.

3.3.4 PBEAM – Area

Il valore numerico dell'area della sezione è ricavabile dal file IFC nella proprietà *CrossSectionArea*, da cui si ottiene il valore numerico di *IfcAreaMeasure*, calcolato da Revit sulla base della geometria della sezione e esportato in IFC. Il numero riportato sotto questa voce può essere facilmente utilizzato per mappare la voce "Area" all'interno del vettore PBEAM in Nastran.

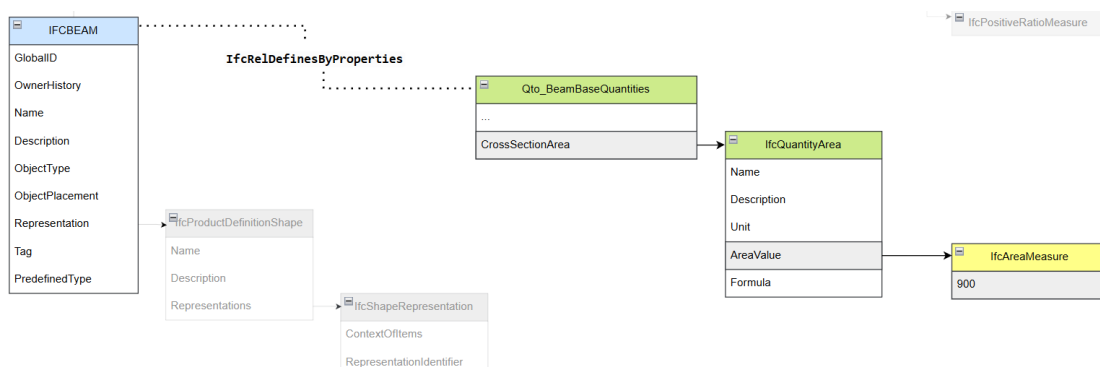


Figura 35 - PBEAM, Area - IfcAreaMeasure. Fonte: autore.

3.3.5 PBEAM – I1, I2, J

In merito ai momenti di inerzia I1 e I2 e al modulo di rigidità torsionale J, si possono trovare all'interno dell'architettura dello standard IFC4X3_ADD2 come illustrato nell'immagine seguente: seguendo il percorso rappresentato si arriva al set *Pset_ProfileMechanical*, che contiene i valori *IfcMomentOfInertiaMeasure* per ciascuno dei momenti d'inerzia e modulo di rigidità torsionale necessari all'analisi strutturale.

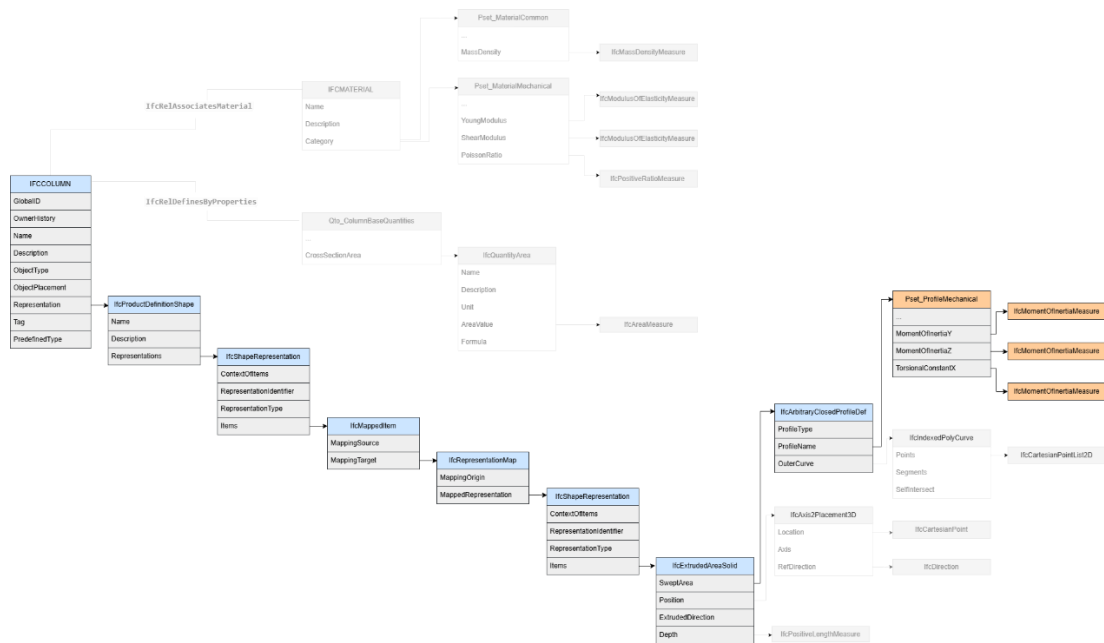


Figura 36 - PBEAM - Pset_ProfileMechanical. Fonte: autore.

In fase sperimentale, tuttavia, si osserva che il *Pset_ProfileMechanical* non viene esportato in IFC per un difetto dello Standard, che magari in futuro potrà essere colmato, ma che oggi risulta un ostacolo in ottica di conversione del file IFC in Nastran. Non essendo dunque possibile ricavare i valori direttamente dallo standard come riportati nell'immagine seguente, si procederà al calcolo degli stessi conoscendone le formule e ricavando i dati necessari in altri punti dell'architettura IFC4X3_ADD2.

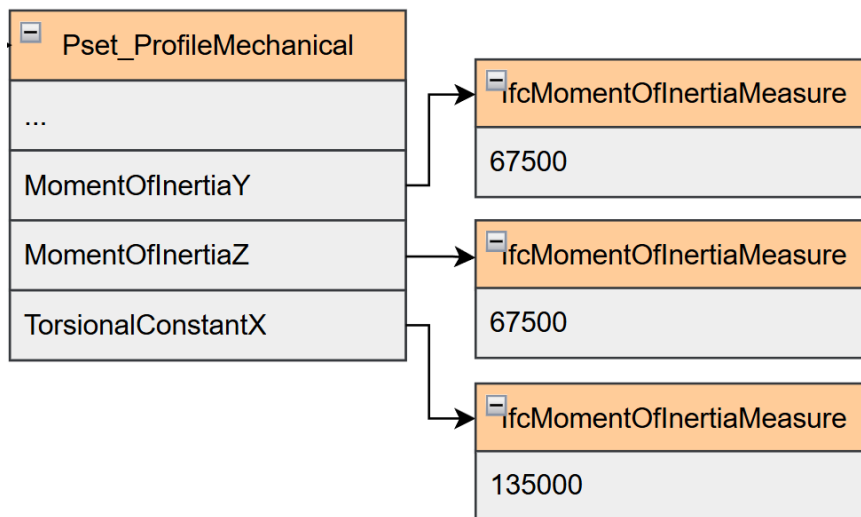
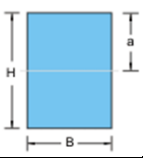
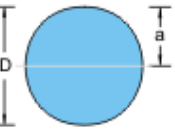


Figura 37 - Pset_ProfileMechanical to IfcMomentOfInertiaMeasure. Fonte: autore.

I Momenti di Inerzia e i moduli di rigidezza torsionale per sezioni a base rettangolare, quadrata o circolare sono calcolabili sulla base delle formule in figura, conoscendo la misura dei lati della sezione. In questa tesi ci si limiterà ai casi studio citati, sapendo che la metodologia esclude tutte le altre possibilità. Per geometrie differenti da queste sarà sufficiente calcolare il momento d'inerzia utilizzando le formule note e realizzando gli stessi passaggi eseguiti in questa sperimentazione.

Tabella 13 - - Momenti d'Inerzia e Modulo di rigidezza torsionale. Fonte: autore.

Sezione	I1 (zz)	I2 (yy)	J
	$\frac{B * H^3}{12}$	$\frac{H * B^3}{12}$	$\frac{H * B^3}{3}$
	$\frac{\pi * D^4}{64}$	$\frac{\pi * D^4}{64}$	$\frac{\pi * (\frac{D}{2})^4}{2}$

Ora bisogna capire come trovare le informazioni geometriche necessarie al calcolo: dobbiamo conoscere i lati minore (B) e maggiore (H) di una sezione rettangolare e per una circolare ci serve il diametro (D). Se analizziamo lo Standard scopriamo facilmente che a partire da *IfcExtrudedAreaSolid*, siamo in grado di trovare i punti del profilo sezionale della colonna o trave in questione. Questi punti sono dati da *IfcCartesianPointList2D*.

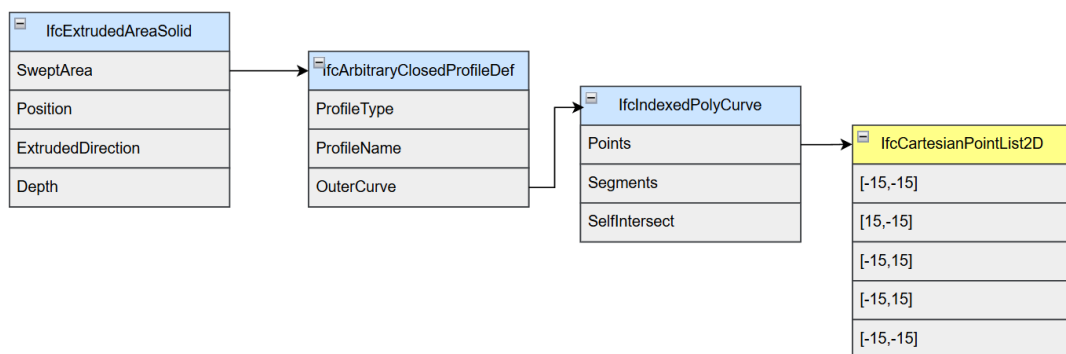


Figura 38 - PBEAM - IfcCartesianPointList2D. Fonte: autore.

Noti i punti, denominati dall'autore P_i , dove il pedice i rappresenta il numero progressivo del punto, siamo in grado di calcolare la distanza L tra questi, che equivale ai lati del rettangolo oppure al diametro della sezione. La formula utilizzata per definire la distanza tra i punti è:

$$L = \sqrt{(x_{P_2} - x_{P_1})^2 + (y_{P_2} - y_{P_1})^2}$$

3.4 CBEAM – PERCORSI

3.4.1 CBEAM – CBEAM

Il primo parametro del vettore CBEAM è sempre “CBEAM” in quanto in questo modo viene identificato il tipo di oggetto che si sta descrivendo: CBEAM definisce un elemento trave di Nastran.

3.4.2 CBEAM – EID

EID è il numero di identificazione dell’elemento trave in Nastran e, per ogni CBEAM, si avrà un $EID = (n + 1)$, dove con n si intende il valore numerico associato alla CBEAM precedente.

3.4.3 CBEAM – PID

PID, il numero identificativo delle proprietà fisiche di una trave, viene richiamato all’interno del vettore CBEAM per associare alla trave di Nastran il set di proprietà corretto, comprendendo le caratteristiche geometriche, fisiche e relative al materiale appartenenti a PBEAM.

3.4.4 CBEAM – GA, GB

GA e GB sono i nodi estremi di una trave CBEAM, che definiscono dove questa inizia e dove finisce. Il nodo GA viene identificato nel file IFC come punto *IfcCartesianPoint*, localizzato grazie all’attributo *Location* di *IfcAxis2Placement3D*. GB, invece, va calcolato aggiungendo alle coordinate di GA la distanza *Depth* di *IfcExtrudedAreaSolid*. Si riporta di seguito un’estrpolazione dell’architettura IFC4X3_ADD2.

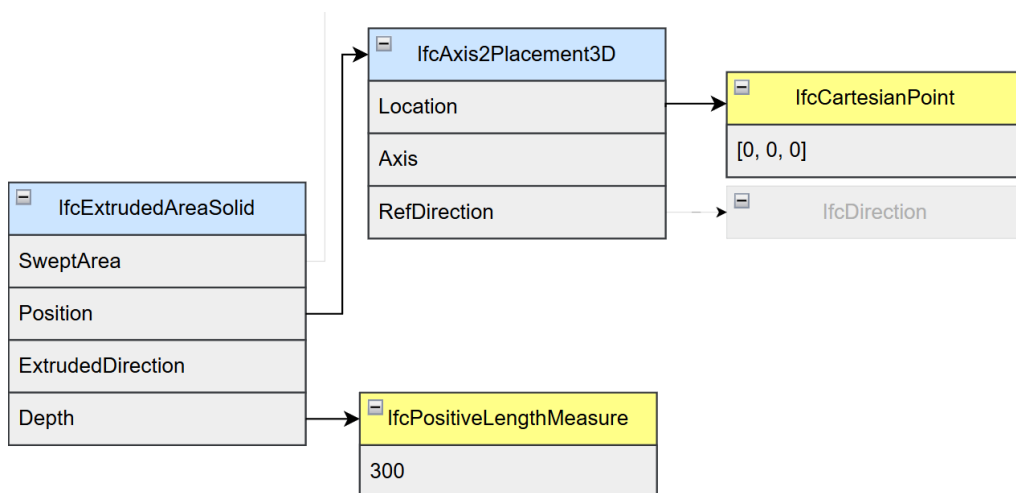


Figura 39 - CBEAM GA, GB - IfcExtrudedAreaSolid. Fonte: autore.

3.4.5 CBEAM – X1, X2, X3

X1, X2 e X3 esprimono in quale direzione deve svilupparsi la trave a partire dal nodo GA fino a quello GB. X1 rappresenta il verso in X, X2 in Y e X3 in Z. All'interno dello Standard IFC queste informazioni sono date dal vettore *IfcDirection*, ricavato dall'attributo *RefDirection* di *IfcAxis2Placement3D*. Si riporta di seguito un'estrapolazione dell'architettura IFC4X3_ADD2.

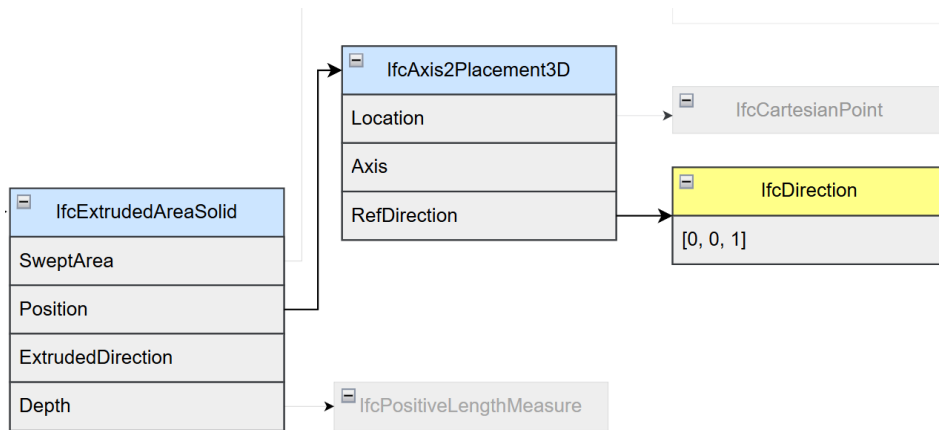


Figura 40 - CBEAM X1, X2, X3 - IfcExtrudedAreaSolid. Fonte: autore.

4 SVILUPPO DI UN APPLICATIVO

Una volta introdotti i percorsi per popolare le informazioni dei vettori del file Nastran, è necessario scrivere un codice attraverso il quale si possa verificare il funzionamento di questo popolamento. Nello script, infatti, verranno innanzitutto estratte le informazioni dal file IFC di input secondo i passaggi descritti a parole nel capitolo precedente, successivamente si creeranno delle stringhe che impostino i vettori di Nastran per un file output con estensione .nas e infine si utilizzeranno i dati di IFC per popolare le costanti del file Nastran.

4.1 Script

Le funzioni e il codice in Python sono illustrati isolando ciascun elemento dello script per descriverne le parti in maniera più comprensiva e in breve si può scomporre il codice in queste sezioni:

- **Importazione librerie:**
 - **ifcopenshell:** questa è la libreria fondamentale per interagire con i file IFC perchè fornisce le API necessarie per leggere, scrivere e manipolare dati all'interno di questi file, consentendo di accedere a tutti gli elementi del modello BIM, alle loro proprietà, relazioni e geometria.
 - **math:** questa è una libreria standard di Python che fornisce funzioni matematiche comuni e in questo caso viene utilizzata per il calcolo delle proprietà della sezione delle travi e delle colonne.
- **Inizializzazione e costanti:**
 - Vengono definite delle costanti per i tipi di card Nastran (GRID, MAT1, PBEAM, CBEAM) e i loro campi.
 - Successivamente, vengono create liste vuote (nastran_grids, nastran_mats, ecc.) per memorizzare i dati Nastran generati.
 - Infine, vengono inizializzati dei contatori e dei dizionari (unique_nodes, unique_materials, unique_profiles) per garantire l'univocità degli ID e delle definizioni.
- **Apertura del File IFC:**
 - Viene aperto il file IFC di input. Questo passaggio deve essere modificato da ciascun utente per definire il percorso in cui si trova il file di input nel proprio computer.
- **Iterazione sugli Elementi IFC:**

Il codice itera su tutti gli elementi di tipo IfcBeam e IfcColumn trovati nel file IFC.

Per ogni elemento:

- **Estrazione della geometria (nodi GRID e orientamento):**
 - Cerca la rappresentazione geometrica dell'elemento, in particolare `IfcExtrudedAreaSolid`.
 - Estrae la Location (GA) che rappresenta il punto di inizio dell'elemento, e calcola il punto finale (GB) aggiungendo la Depth (lunghezza dell'estrusione) normalizzata lungo tutti e tre gli assi.
 - Estrae `RefDirection` per definire l'orientamento (X1, X2, X3) dell'elemento.
 - Popola le card GRID per GA e GB, assicurandosi che i nodi duplicati non vengano creati (gestione dell'univocità tramite `unique_nodes`).
- **Estrazione del materiale (MAT1):**
 - Cerca le associazioni di materiale (`IfcRelAssociatesMaterial`).
 - Estrae le proprietà del materiale dai `Pset_MaterialCommon` (densità) e `Pset_MaterialMechanical` (modulo di Young, modulo di taglio, rapporto di Poisson).
 - Popola la card MAT1 con queste proprietà, gestendo l'univocità dei materiali tramite `unique_materials`.
- **Estrazione e calcolo delle proprietà della sezione (PBEAM):**
 - Cerca le proprietà della sezione (area, momenti d'inerzia e costante torsionale) tramite `IfcRelDefinesByProperties` e `IfcElementQuantity` per l'area.
 - Analizza la geometria del profilo (`IfcArbitraryClosedProfileDef`, `IfcIndexedPolyCurve`) per determinare dimensioni come base, altezza o diametro.
 - Calcola i momenti d'inerzia (I1, I2) e la costante di torsione (J) della sezione in base alla sua forma (rettangolare o circolare), utilizzando appunto le funzioni matematiche della libreria `math`.
 - Popola la card PBEAM con l'area e le inerzie calcolate, associandola al materiale estratto e gestendo l'univocità dei profili tramite `unique_profiles`.
- **Creazione dell'elemento trave (CBEAM):**

- Una volta estratti tutti i dati necessari (ID del profilo, ID dei nodi GA e GB, orientamento), popola la card CBEAM che definisce l'elemento trave tra i due nodi.
- **Generazione del file Nastran:**
 - Al termine dell'elaborazione di tutti gli elementi, lo script stampa un riepilogo delle entità Nastran generate.
 - Crea un file di output chiamato "output.nas".
 - Scrive le card GRID, MAT1, PBEAM e CBEAM nelle rispettive sezioni del file Nastran, seguendo il formato standard.
- **Interfaccia utente (opzionale):**
 - Chiede all'utente se desidera visualizzare un'anteprima dei dati Nastran generati direttamente a video.

5 VALIDAZIONE

5.1 Revit

5.1.1 Scelta di Revit

Il software Revit è stato scelto per modellare i casi studio per via della sua grande popolarità: risulta ad oggi il programma di modellazione BIM più utilizzato nel mondo. Revit appartiene ad Autodesk ed è stato lanciato inizialmente il 5 aprile 2000. Da allora, Autodesk ha portato con successo questo strumento al grande pubblico, posizionandolo ora come lo strumento di progettazione più diffuso nel settore AEC.

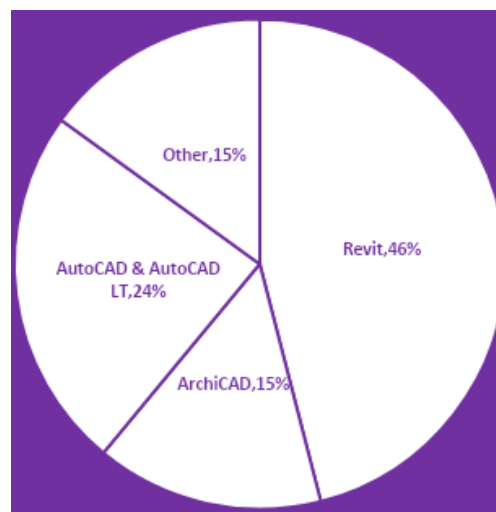


Figura 41 - Popolarità del software Revit nel mondo. Fonte: (NBS, 2019)

5.1.1 Il software Autodesk Revit

Autodesk Revit è un programma CAD e BIM per sistemi operativi Windows, creato dalla Revit Technologies Inc. e comprato nel 2002 dalla Autodesk per 133 milioni di dollari, che consente la progettazione con elementi di modellazione parametrica e di disegno (Autodesk Revit, 2026).

Le funzionalità in continua trasformazione e i set di strumenti multidisciplinari di Revit consentono a tutti gli architetti, gli ingegneri e i professionisti del settore edilizio di svolgere il proprio lavoro al meglio, sia singolarmente che in gruppo. Infatti, per migliorare la collaborazione tra professionisti, Autodesk ha sviluppato tre versioni di Revit per differenti discipline di progettazione edilizia:

- Revit Architecture, per architetti e progettisti edili
- Revit Structure, per ingegneri strutturali
- Revit MEP, per ingegneri meccanici, elettrici e termotecnici

Oggi le tre versioni sono accorpate in un unico programma, che contiene i tre moduli e un modulo di analisi.

Le diverse funzionalità proposte dal software includono la possibilità di modellare componenti parametrici, la presenza di abachi per ordinare i dati di progetto, poter sincronizzare i file con altri professionisti, definire le fasi del progetto e applicare dei filtri di fase in fase per presentare il progetto durante i vari stadi di lavoro.

5.1.2 Modello analitico di Revit

In Revit il modello fisico viene creato in viste che rappresentano la struttura fisica e può essere considerato come un set di disegni di produzione. Durante la creazione del modello fisico, viene contemporaneamente generato il modello analitico.

È possibile visualizzare, regolare e gestire automaticamente il modello analitico creato in Revit nel corso dello sviluppo del modello di edificio fisico. La visualizzazione del modello analitico varia in base al modello di progetto, al modello di vista, ai parametri di vista e agli stili degli oggetti. Il modello di progetto dell'analisi strutturale contiene due viste di pianta analitiche strutturali e una vista analitica 3D.

Un modello analitico corrisponde ad una rappresentazione 3D semplificata della descrizione tecnica completa di un modello fisico strutturale. Il modello analitico è costituito da componenti strutturali, geometria, proprietà di materiali e carichi che insieme formano un sistema di progettazione.

In merito alle condizioni al contorno, è possibile applicare carichi strutturali al modello analitico per valutare possibili deformazioni e sollecitazioni nel progetto. I vincoli esterni, invece, consentono di definire le condizioni di supporto di un elemento strutturale in base all'ambiente circostante (Autodesk Revit Guida, 2015).

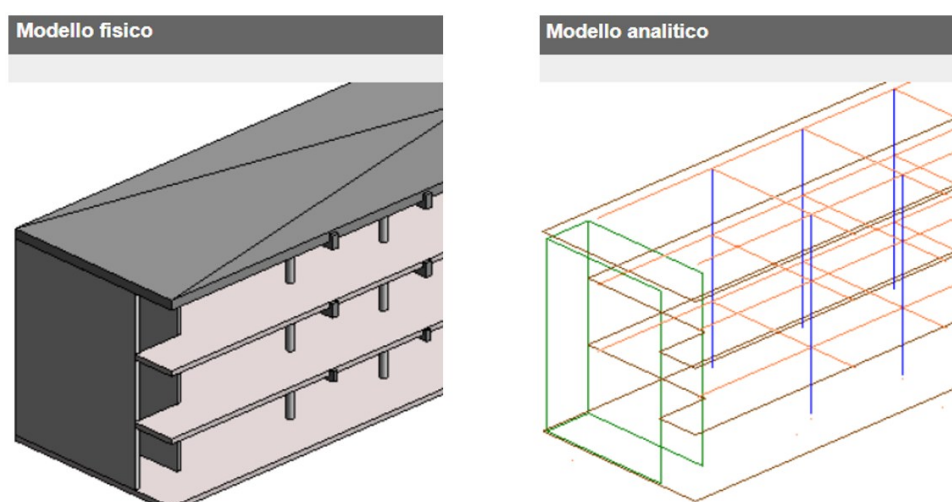


Figura 42 - Modello fisico e modello analitico di Revit. Fonte: (Autodesk Revit Guida, 2015)

Il modello analitico di una struttura include una serie di modelli analitici di elementi strutturali, uno per ogni elemento della struttura. Sono associati modelli analitici di elementi strutturali ai seguenti elementi: pilastri strutturali, elementi del telaio strutturale quali travi e controventi, pavimenti strutturali, muri strutturali ed elementi della fondazione strutturale.

Il modello analitico di un singolo elemento strutturale include:

- Parametri di istanza
- Proprietà fisiche dei materiali
- Posizione di default relativa all'elemento strutturale
- Posizione rispetto al piano di proiezione al momento dell'inserimento o della regolazione

5.1.3 Esportazione in IFC da Revit

In Revit sono disponibili funzionalità di esportazione e importazione IFC con certificazione completa, basate sullo standard per lo scambio dei dati buildingSMART® IFC. Revit esporta gli elementi di costruzione in un file IFC secondo le categorie e le sottocategorie a cui gli elementi appartengono. Ad esempio, in Revit un muro viene esportato nell'entità `IfcWallStandardCase` in quanto il muro è un elemento della categoria muro. In molti casi, l'esportazione di elementi da Revit in un file IFC risulta alquanto semplice e le impostazioni di default si rivelano appropriate. In alcuni, tuttavia, è possibile che sia necessario specificare l'entità IFC di appartenenza degli elementi di una particolare famiglia (Autodesk Revit Guida, 2017).

5.1.4 Plug-in GeometryGym per Revit

Il plug-in GeometryGym per Revit offre una serie di funzionalità che estendono le capacità di importazione ed esportazione IFC di Revit, oltre a una serie di strumenti che semplificano il collegamento dei parametri IFC con gli elementi e i progetti Revit.

Il plug-in fornisce un importatore di file IFC avanzato, complementare a quello dell'applicazione Revit integrata, che consente l'aggiornamento dei file Revit con le relative importazioni di file IFC e l'importazione di librerie di oggetti IFC.

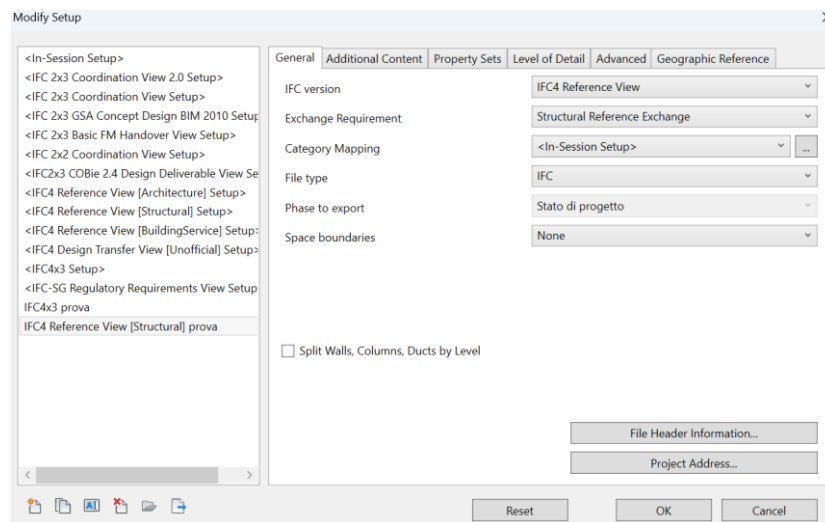
GeometryGym fornisce, inoltre, diversi programmi di installazione per il trasferimento automatico di modelli Revit in programmi di analisi strutturale. Questi strumenti possono essere forniti su richiesta, in quanto dipendono dalla versione di Revit e del programma di analisi strutturale (GeometryGym, 2026).

Per esportare un modello di analisi strutturale da Revit, secondo il sito ufficiale di GeometryGym, bisogna procedere come segue:

1. Passare a una vista 3D contenente tutti gli elementi che si desidera considerare nell'esportazione.
2. Rimuovere il parametro "Includi analitico" in Revit per tutti gli elementi di inquadratura di piccole dimensioni che non si desidera includere nell'esportazione.
3. Definire gli utilizzi dei membri come richiesto per consentire l'esecuzione di un'aggregazione appropriata dei membri nel processo di rilevamento e aggancio strutturale.
4. Fare clic su Esporta Revit nella scheda del plug-in.
5. Assegnare le opzioni di esportazione desiderate e quindi esportare.

Nei casi studio presentati di seguito l'esportazione è avvenuta seguendo le indicazioni riportate da GeometryGym nel suo sito ufficiale (GeometryGym, 2026). Le fasi che sono state eseguite sono le seguenti:

1. Modificare le impostazioni di esportazione IFC di Revit e salvarle: in questo caso è stata impostata la versione IFC4, che deve includere in fase di esportazione gli elementi visibili della vista.



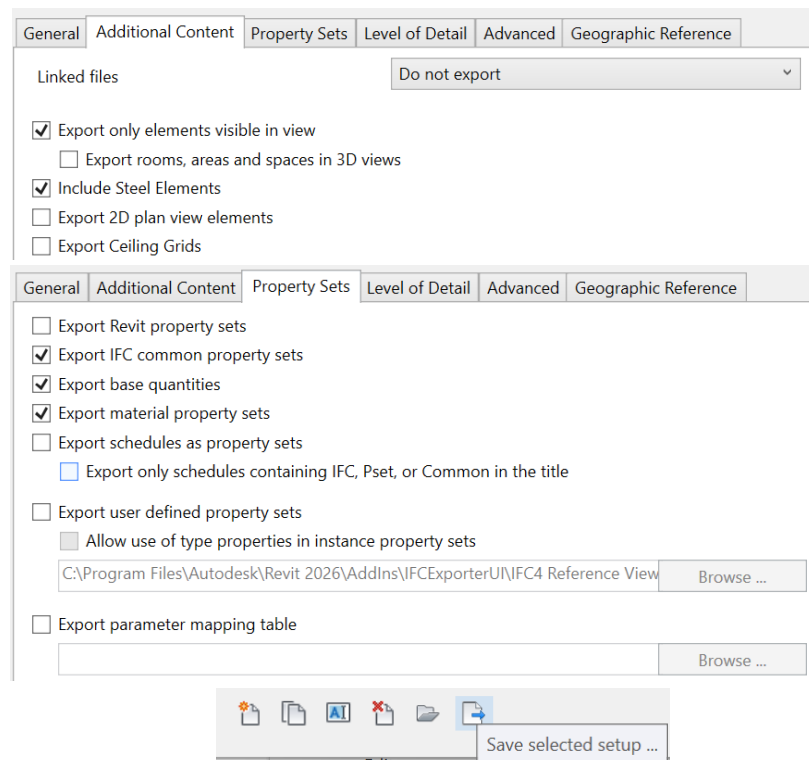
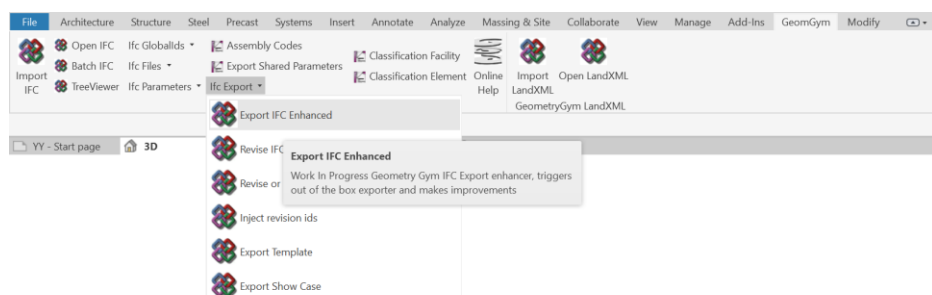


Figura 43 - Impostazioni di esportazione IFC da Revit. Fonte: autore.

- Entrare nel plug-in GeometryGym dal menu a tendina di Revit ed esportare IFC dalla sezione *Export IFC Enhanced*. Dopo aver confermato il nome del file che si vuole esportare e il suo percorso, compare in automatico una finestra *Export Options*: bisogna assicurarsi che nella configurazione sia stato selezionato il file creato precedentemente.



Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM IFC e NASTRAN
Alice Pin

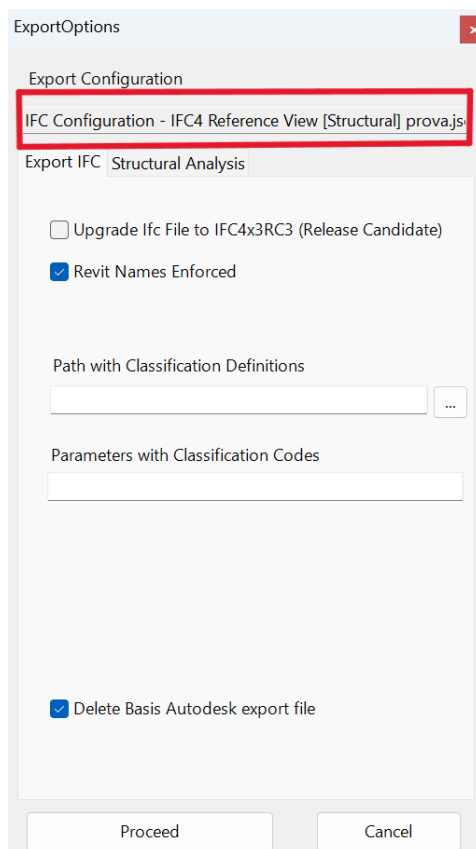


Figura 44 - Impostazione di esportazione IFC con GeometryGym. Fonte: autore.

5.2 Materiale casi studio

Il materiale attribuito alle strutture è per lo più il calcestruzzo, non armato in quanto le armature avrebbero complicato eccessivamente il modello e reso disomogeneo il materiale, allontanando la sperimentazione dai nostri fini e rendendo infattibile il trasferimento di dati (L. Muratore, 2018). Le caratteristiche fisiche e meccaniche del calcestruzzo utilizzato nella modellazione sono:

- Densità: 2406.53 kg/m³
- Modulo di Taglio: 12463.00 MPa
- Modulo Elastico di Young: 29910.00 Mpa
- Modulo di Poisson: 0.20

Per il caso studio 1 nella sua variante C, si è utilizzato anche il materiale ferro, le cui caratteristiche fisiche e meccaniche sono:

- Densità: 7110.00 kg/m³
- Modulo di Taglio: 64900.00 MPa
- Modulo Elastico di Young: 162000.00 Mpa
- Modulo di Poisson: 0.28

5.3 Caso studio 1: colonna

Il primo caso studio vede la modellazione di una colonna di Autodesk Revit, avente sezione quadrata di 30 cm x 30 cm e altezza 300 cm. La colonna fisica, in calcestruzzo, è stata successivamente trasformata in colonna analitica di Revit. Per verificare al meglio l'esportazione del caso studio è stato inserito anche un carico puntuale pari a 5 kN orientato in direzione parallela all'asse della colonna e con verso comprimente.

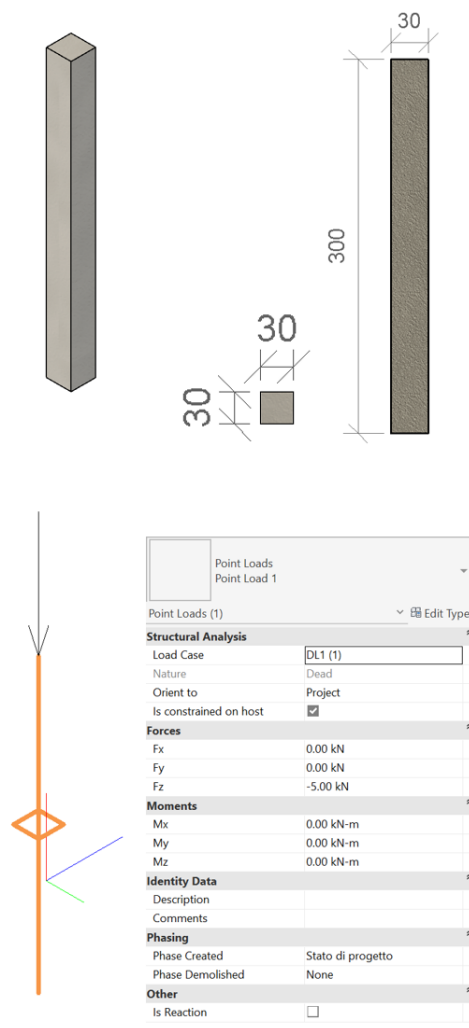


Figura 45 - Colonna a base quadrata modellata in Revit. Fonte: autore.

5.3.1 Variante A: colonna a base circolare

Con il fine di valutare come vengono esportati i dati al variare della geometria della sezione, è stato scelto di realizzare una variante della colonna sopra illustrata. Questa variante ha la medesima altezza e medesimo materiale, ma vede una base circolare con diametro di 30 cm. Non è stato inserito il carico puntuale in quanto la sua esportazione viene già verificata con il caso studio principale. Di seguito un'immagine illustrativa del modello Revit realizzato per la variante A.

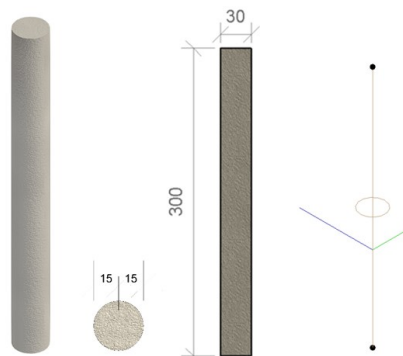


Figura 46 - Colonna a base circolare modellata in Revit. Fonte: autore.

5.3.2 Variante B: colonna a base quadrata inclinata rispetto all'asse Z

Per eseguire una sperimentazione il più completa possibile, è stata realizzata anche una variante B della colonna a base quadrata descritta inizialmente: la struttura risulta avere le stesse dimensioni, ma è stata modellata con asse inclinato rispetto all'asse z e rispetto all'asse x. Tale modifica consentirà di studiare al meglio come viene identificata la direzione di estrusione della sezione di un qualsiasi elemento IfcColumn.

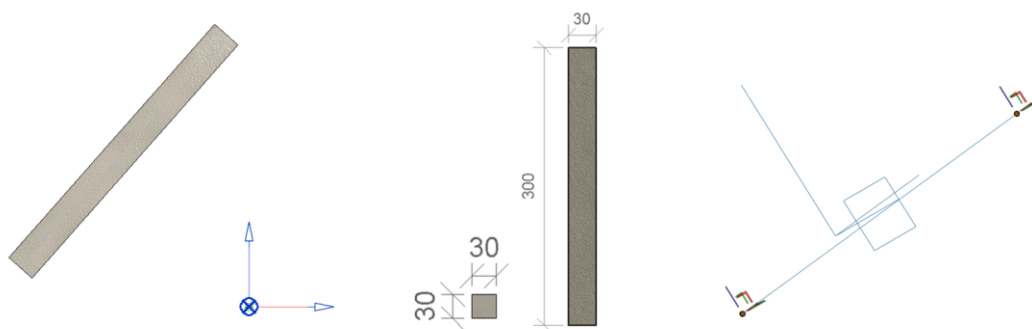


Figura 47 - Colonna a base quadrata inclinata modellata in Revit. Fonte: autore.

5.3.3 Variante C: due colonne

Con il fine di valutare come vengono esportati i dati laddove vi siano due colonne aventi materiali diversi, è stata modellata per due volte la colonna descritta al caso studio 1, con la variante che una delle due è in calcestruzzo e l'altra in ferro.

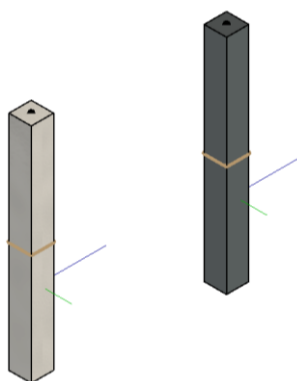


Figura 48 - Caso studio 1.C due colonne. Fonte: autore.

5.4 Caso studio 2: trave

Il secondo caso studio vede la modellazione di una trave di Autodesk Revit, avente sezione quadrata di 30 cm x 30 cm e lunghezza 300 cm. La trave fisica, in calcestruzzo, è stata successivamente trasformata in trave analitica di Revit. Per verificare al meglio l'esportazione del caso studio è stato inserito anche un carico puntuale pari a 5 kN orientato in direzione ortogonale all'asse della trave e posizionato all'estremità della stessa. La trave qui individuata ha asse di estrusione della sezione orientato in direzione x.

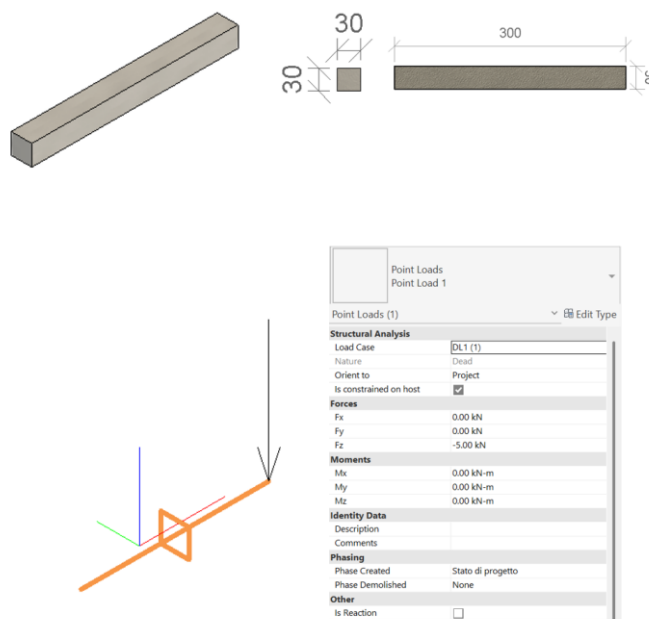


Figura 49 - Trave modellata in Revit. Fonte: autore.

5.4.1 Variante A: trave sviluppata in direzione Y

Per questo caso studio risultava interessante valutare come vengono esportati i dati in caso di estrusione della sezione quadrata in direzione y, diversa da quella del caso studio 2. Per questa ragione è stata rimodellata la medesima trave descritta sopra con la sola variante che l'elemento si sviluppa in direzione ortogonale a quella illustrata nell'immagine precedente.

5.5 Caso studio 3: telaio

Il terzo caso studio si compone di una struttura più complessa delle precedenti costituita da travi e colonne collegate tra loro a formare un telaio e alla base di ciascuna colonna vi è un plinto di fondazione a forma parallelepipedica. Gli elementi verticali e orizzontali hanno sezione quadrata di 30 cm x 30 cm e sono costituiti di calcestruzzo. Le quattro colonne hanno altezza pari a 350 cm sono posizionate su griglia quadrata con un interasse tra loro pari a 500 cm. Le travi hanno lunghezza pari a 500 cm e sono posizionate a forma quadrata con le colonne nei vertici della figura. I plinti di fondazione sono parallelepipedici con base 100 cm x 100 cm e altezza 50 cm.

La struttura fisica è stata successivamente trasformata in struttura analitica di Revit, aggiungendo in un secondo momento condizioni al contorno quali vincoli e carichi. I vincoli che sono stati scelti sono incastro alla base delle colonne e cerniere nei punti di contatto tra travi e colonne. Si nota dalle immagini seguenti che i plinti di fondazione non vedono un elemento corrispettivo nel modello analitico e sono stati semplicemente rimodellati dall'autore come

vincoli. Il carico che è stato modellato è un carico lineare, applicato su una colonna in direzione ortogonale al suo asse e con modulo 3 kN/m.

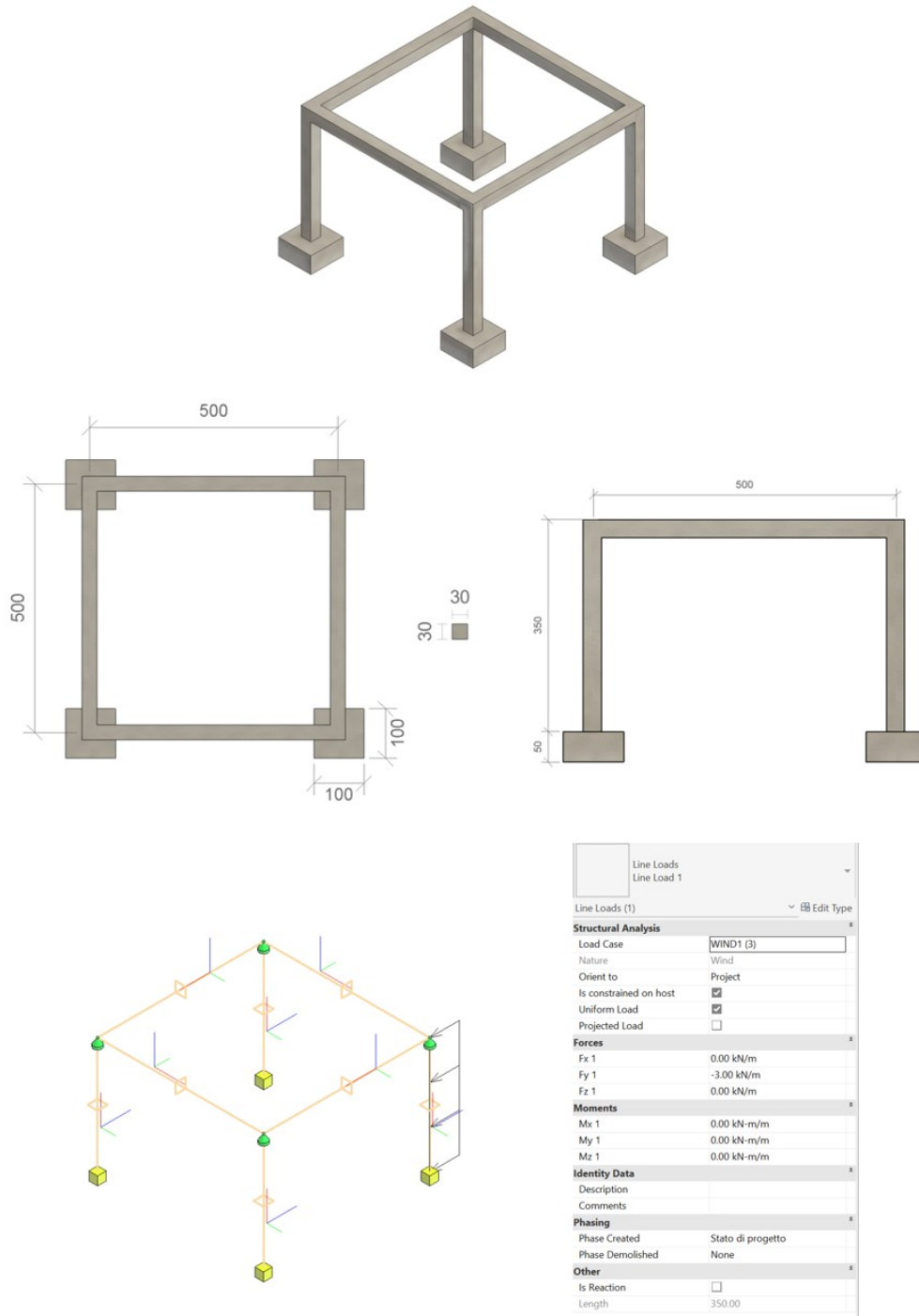


Figura 50 - Telaio modellato in Revit. Fonte: autore

5.6 File dei casi studio

Per completezza, si riporta in questo paragrafo un esempio di file IFC utilizzato come caso studio 1 colonna a base quadrata. In questo codice si può facilmente ritrovare l'architettura esplicitata al Capitolo Quarto.

```
#81= IFCCOLUMN('2EvtEz4whjwojb3Avwi',#6,'300 x 300mm',$,'Concrete Square:300 x 300mm',#188,#189,'322852',.COLUMN.);
...#6= IFCOWNERHISTORY(#1,#4,$,.ADDED,$,$,1772697946);
...#188= IFCLOCALPLACEMENT(#76,#80);
...#189= IFCPRODUCTDEFINITIONSHAPE($,$,#190,#193);
...#190= IFCSHAPEREPRESENTATION(#161,'Body','MappedRepresentation',(#191));
...#161= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,*,*,#162,$,.MODEL_VIEW,$);
...#191= IFCMAPPEDITEM(#157,#192);
...#157= IFCREPRESENTATIONMAP(#158,#160);
...#158= IFCAXIS2PLACEMENT3D(#159,$,$);
...#160= IFCSHAPEREPRESENTATION(#161,'Body','SweptSolid',(#168));
...#161= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,*,*,#162,$,.MODEL_VIEW,$);
...#168= IFCEXTRUDEDAREASOLID(#173,#176,#178,3000.0);
...#173= IFCARBITRARYCLOSEDPROFILEDEF(.AREA,'300 x 300mm',#174);
...#174= IFCINDEXEDPOLYCURVE(#175,$,.F.);
...#176= IFCAXIS2PLACEMENT3D(#177,#178,#180);
...#177= IFCARTESIANPOINT((-779.6813,796.0674,0.0));
...#178= IFCDIRECTION((0.0,0.0,1.0));
...#180= IFCDIRECTION((0.0,-1.0,0.0));
...#178= IFCDIRECTION((0.0,0.0,1.0));
...#192= IFCARTESIANTRANSFORMATIONOPERATOR3D($,$,#159,1.0,$);
...#193= IFCSHAPEREPRESENTATION(#167,'Axis','MappedRepresentation',(#194));

#298= IFCMATERIALPROPERTIES('Pset_MaterialMechanical',$,(#294,#295,#296,#297),#82);ENDSEC;END-ISO-10303-21;
...#294= IFCPROPERTYSINGLEVALUE('YoungModulus',$,IFCMODULUSOFELASTICITYMEASURE(9116568000.0),$);
...#295= IFCPROPERTYSINGLEVALUE('ShearModulus',$,IFCMODULUSOFELASTICITYMEASURE(3798722400.0),$);
...#296= IFCPROPERTYSINGLEVALUE('PoissonRatio',$,IFCPOSITIVERATIOMEASURE(0.2),$);
...#297= IFCPROPERTYSINGLEVALUE('ThermalExpansionCoefficient',$,IFCTHERMALEXPANSIONCOEFFICIENTMEASURE(0.0000099),$);
...#82= IFCMATERIAL('Concrete - Cast-in-Place Concrete',$,'Concrete');

#95= IFCELEMENTQUANTITY('2y5UA3Z_Y8XAIt1E_FEAaK',#6,'Qto_ColumnBaseQuantities',$,$,(#96,#97,#98,#99,#100));
...#6= IFCOWNERHISTORY(#1,#4,$,.ADDED,$,$,1772697946);
...#96= IFCQUANTITYLENGTH('Length',$,$,3000.0,$);
...#97= IFCQUANTITYAREA('CrossSectionArea',$,$,0.09,$);
...#98= IFCQUANTITYAREA('OuterSurfaceArea',$,$,3.6,$);
...#99= IFCQUANTITYVOLUME('GrossVolume',$,$,0.27,$);
...#100= IFCQUANTITYVOLUME('NetVolume',$,$,0.27,$);
```

Figura 51 - Codice Caso Studio 1. Fonte: autore.

Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM
IFC e NASTRAN
Alice Pin

Capitolo Quarto

RISULTATI

1 ANALISI CRITICA DEI RISULTATI

A seguito della stesura dello Script è stato testato il funzionamento dello stesso sui casi studio presentati nella sezione precedente. In questo paragrafo si dimostra che la conversione è fattibile e, anche se con qualche deficit, funziona per il trasferimento di dati da IFC a Nastran nei casi presentati. Inoltre, l'importazione del file Nastran nel software Strand7 è avvenuta correttamente, consentendo di ottenere un modello analitico utile all'analisi strutturale.

```
$ NASTRAN BULK DATA FILE generato da script Python
BEGIN BULK
$
$ SEZIONE GRID (Nodi)
GRID,1,0,-7838.982392442766,9407.065065683135,-350.0
GRID,2,0,-7838.982392442766,9407.065065683135,6.821210263296962e-13
$
$ SEZIONE MAT1 (Materiali)
MAT1,1,29909999999.999996,12462999999.999998,0.2,2406.5302626279113
$
$ SEZIONE PBEAM (Proprieta sezioni)
PBEAM,1,1,4.200000000000023,67500.00000000099,67500.00000000099,270000.00000000396
$
$ SEZIONE CBEAM (Connettivita elementi)
CBEAM,1,1,1,2,0.0,0.0,-1.0
$
ENDDATA
```

Figura 52 - Risultato output Nastran. Fonte: autore.

1.1 Caso studio 1: risultati

La colonna a base quadrata viene restituita nel software di analisi Fem come un elemento Beam in calcestruzzo armato, avente i valori di densità, modulo elastico, modulo di taglio e coefficiente di Poisson corretti. Anche la modellazione geometrica viene riportata correttamente in relazione al sistema di coordinate: la colonna si sviluppa tra due elementi GRID e viene rappresentata come una linea poiché è un elemento analitico. Da un punto di vista delle proprietà geometriche della sezione va messo in evidenza il fatto che la Shape non viene definita, come prevedibile poiché non abbiamo estrapolato da IFC i dati sul tipo di forma che ha la sezione (per l'analisi strutturale è il valore numerico dell'area sezionale a influenzare la resistenza della trave e non tanto la sua forma). Tuttavia, siamo in grado di leggere nel file i valori di area, momenti d'inerzia e modulo di rigidezza torsionale sotto la voce Properties.

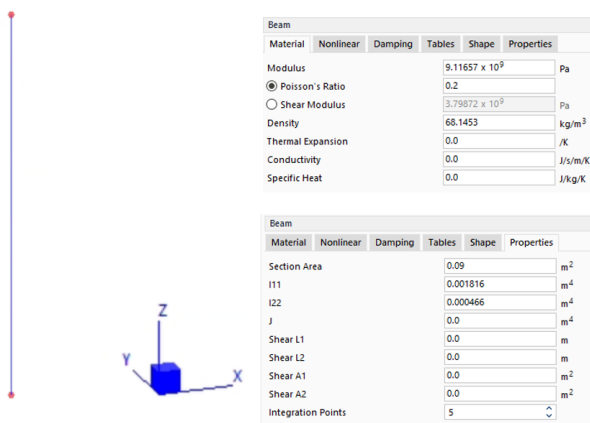


Figura 53 – Risultati Caso studio 1. Fonte: autore.

La colonna a base circolare vede il medesimo risultato in merito al materiale e alla rappresentazione geometrica, così come per le caratteristiche della sezione.

La colonna inclinata viene modellata correttamente come una Beam con GRID come estremi, identificando due nodi GA e GB spostati rispetto alla verticale e una direzione di estrusione dell'elemento orientata correttamente verso tutte le coordinate x, y, z. Anche le proprietà geometriche, sezionali e quelle del materiale del file di partenza IFC si ritrovano perfettamente negli elementi di Nastran.

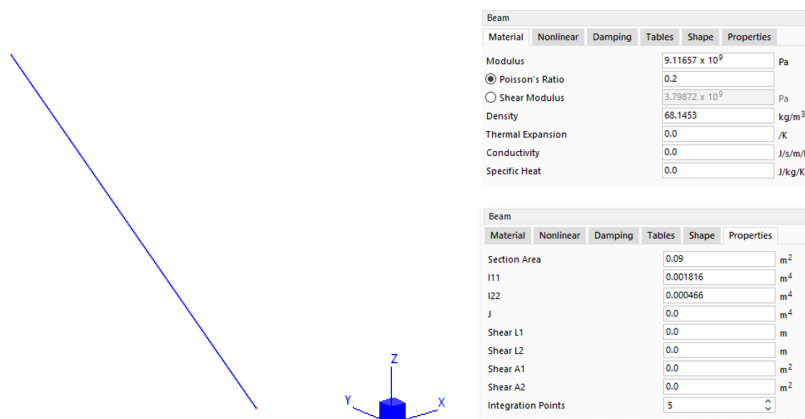


Figura 54 - Risultati Caso studio 1.B. Fonte: autore.

Nella variante con due colonne, vengono rappresentate perfettamente sia le proprietà materiche e sezionali della colonna in ferro sia quelle della colonna in calcestruzzo, a dimostrazione della validità della mappatura proposta per la conversione. Anche la posizione e la direzione di estrusione delle colonne è corretta.

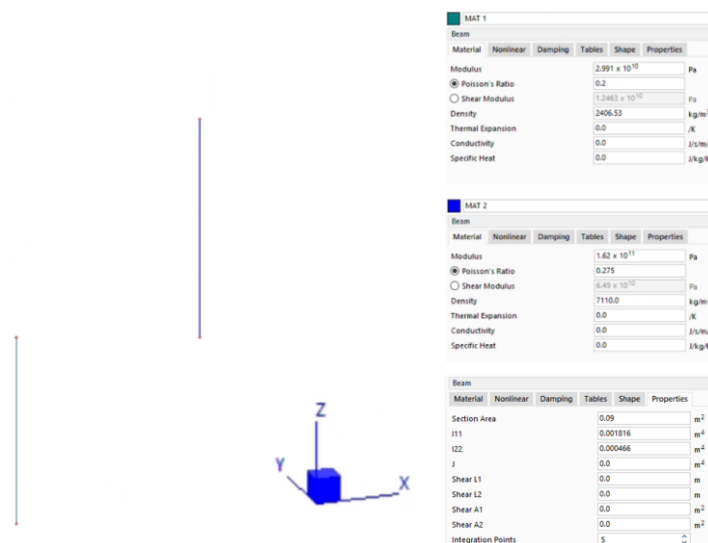


Figura 55 - Risultati Caso studio 1.C. Fonte: autore.

1.2 Caso studio 2: risultati

Le travi del caso studio 2 erano state modellate ortogonalmente all'asse Z prima in direzione X e dopo in direzione Y e avevano le medesime caratteristiche geometriche e materiche della colonna presentata inizialmente nel caso studio 1. A seguito della conversione, in Strand7 è possibile importare il file Nastran contenente le travi convertite in elementi analitici e il risultato è positivo: esse possiedono le corrette proprietà sezionali, l'associazione al materiale, i nodi di inizio e fine e vengono rappresentate nel software come linee.

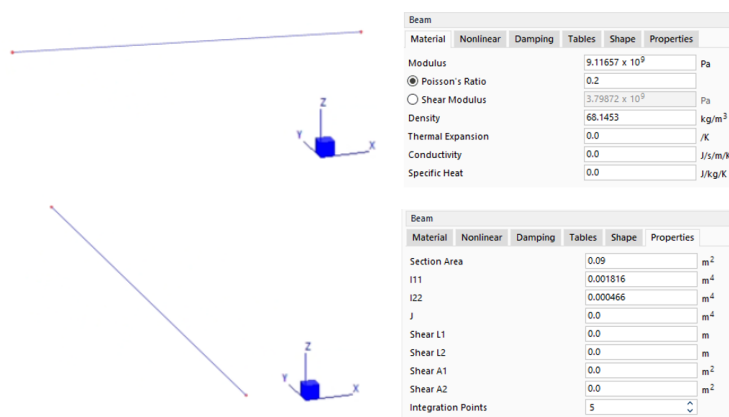


Figura 56 - Risultati Caso studio 2. Fonte: autore.

1.3 Caso studio 3: risultati

Andando a valutare i risultati sulla conversione del telaio, ci si rende subito conto che la complessità leggermente superiore del caso studio rende difficoltosa una conversione corretta. Le criticità da mettere in evidenza in questo caso sono due: innanzitutto, nonostante i valori

dei materiali e delle proprietà sezionali siano corretti, vi è un difetto nel codice poiché viene creata una proprietà PBEAM per ciascuno degli elementi (travi e colonne) presenti nel modello, quando questa dovrebbe essere univoca in quanto uguale per tutti gli elementi.

Una seconda criticità, più significativa della precedente, è relativa all'identificazione dei punti GA e GB di inizio e fine degli elementi trave e colonna: essendoci in questo caso quattro travi e quattro colonne, la conversione della geometria degli elementi non avviene correttamente per quanto riguarda i punti GA e GB di inizio e fine dei diversi elementi e vengono quindi estrusi seguendo una direzione errata. Questo aspetto riguarda il modo in cui le connessioni tra diversi elementi vengono definite e tale criticità è emersa anche dallo studio della letteratura. Questo gap non è stato risolto con la sperimentazione presentata in questa sede e ci si auspica che nelle ricerche future, approfondendo e implementando lo studio dello Standard IFC in merito alle connessioni, sia possibile superare il problema.

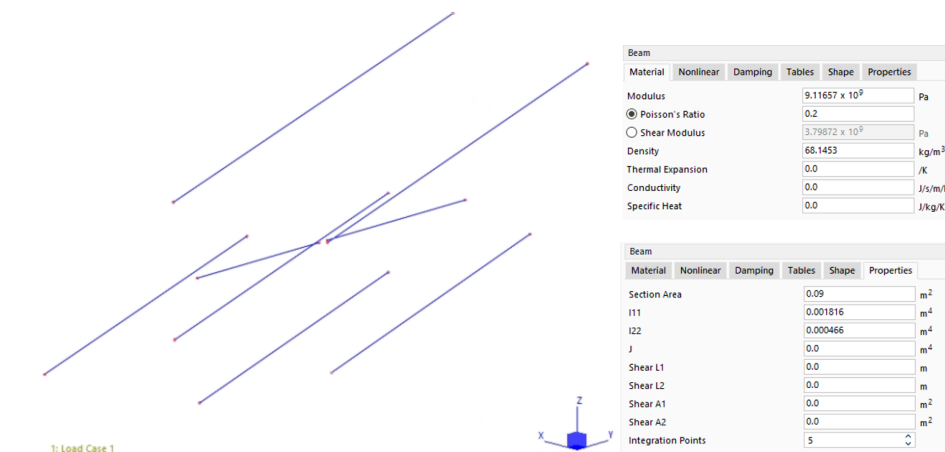


Figura 57 - Risultato 1 Caso studio 3. Fonte: autore.

2 CRITICITA' DEL METODO

Il metodo presentato in questa tesi ha alcune criticità: innanzitutto è stata messa in evidenza la difficoltà nell'esportare un modello analitico in IFC. Sebbene lo Standard, grazie al dominio *IfcStructuralDomain*, riesca a descrivere molto bene le strutture analitiche, di fatto software BIM come Revit o ArchiCAD non sono capaci, ad oggi, di esportare le informazioni relative a carichi e vincoli in un file IFC. Questo aspetto è messo in evidenza dagli stessi produttori dei software, nonché da diversi altri autori presentati nella sezione dedicata alla letteratura. Altrettanto, il Pset_ProfileMechanical non viene esportato in IFC e per tanto è stato necessario calcolare i momenti d'inerzia e quello torsionale attraverso formule matematiche, che per quanto funzionali sono limitanti laddove si voglia realizzare un applicativo che includa qualsiasi possibile geometria di sezione trasversale.

In secondo luogo, si sottolinea la semplicità dei casi studio presi ad esempio in questo capitolo: sono stati utilizzati dei modelli molto semplificati per valutare in generale il potenziale funzionamento di questa tecnica. Nella realtà, perché un metodo sia applicabile, esso va sviluppato ben oltre lo stato embrionale presentato in questa sede, ma soprattutto va testato su modelli più complessi che diano la garanzia della sua applicabilità in qualsiasi contesto e valutando ogni variabile modellabile.

Inoltre, dal caso studio del telaio emerge la necessità di un approfondimento dello studio dello standard IFC in merito al posizionamento reciproco degli elementi, con una conseguente revisione del popolamento dei nodi GRID e ristestura dello script. Tale aspetto è fondamentale per assicurare il trasferimento corretto delle informazioni geometriche dal modello di input a quello di output. Un altro punto da migliorare è la creazione di molteplici PBEAM identiche tra loro, che vanno ad appesantire inutilmente il file Nastran. Questo aspetto è facilmente risolvibile modificando lo script.

Non da meno, si evidenzia che la stesura dello script si basa su competenze informatiche elementari e va sicuramente rivista per migliorare il suo funzionamento e renderlo più ottimizzato. Il codice presentato in questa tesi è un punto di partenza dal quale partire per svilupparne uno professionale e affidabile, grazie anche all'appoggio di tecnici esperti.

Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM
IFC e NASTRAN
Alice Pin

Il documento è iniziato da uno studio approfondito dello stato dell'arte in ambito di interoperabilità BIM-FEM, valutando i metodi proposti dagli autori per arrivare a identificare una metodologia preferenziale in ottica di conversione da un file IFC a uno FEM. Per sviluppare la metodologia, è stato necessario approfondire dettagliatamente i modelli dati sia dello Standard IFC sia di Nastran, così da comprendere al meglio come gli elementi analitici di Nastran potessero essere descritti e individuati attraverso le entità informative di IFC. Successivamente si è proceduto al popolamento degli elementi Nastran con le informazioni prese da IFC e alla stesura di un codice che fungesse da convertitore.

L'applicativo è stato valutato attraverso diversi casi studio e dai risultati è emerso che il metodo proposto risulta vincente, dando già ora dei risultati positivi, sebbene parziali, a dimostrazione della validità della mappatura presentata. Questo ci fa presupporre che per il futuro abbia senso dedicare del tempo a sviluppare e migliorare tale metodo, basato su un codice convertitore. In questo senso, i punti critici sui quali bisognerebbe intervenire primariamente sono la stesura del codice, che va migliorata per garantire un suo funzionamento ottimizzato. In secondo luogo, dovendo in futuro aumentare la complessità dei modelli rispetto a quelli descritti come casi studio, è necessaria un'implementazione dello studio dello standard IFC, come dimostrato dai risultati deficitari ottenuti per il caso studio del telaio. In particolare, risulta primariamente importante soffermarsi sul modo in cui IFC descrive il posizionamento reciproco degli elementi laddove questi siano a contatto.

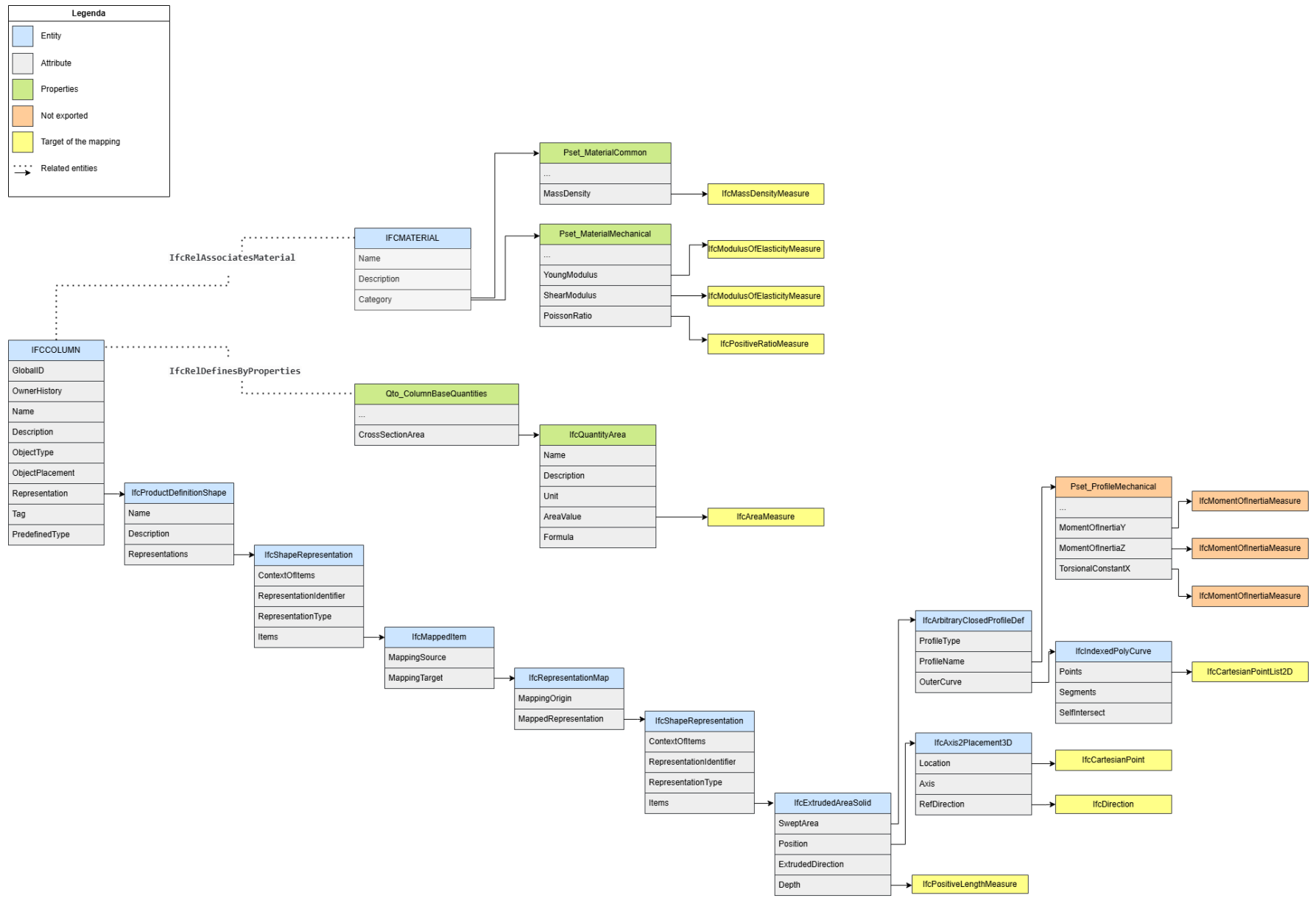
Inoltre, per il futuro ci si auspica che le energie vengano innanzitutto devolute ad accrescere notevolmente la capacità dei software BIM di esportare in formato IFC i modelli analitici, poiché è stato ampiamente dimostrato che queste risultano essere carenti sia per quanto riguarda le condizioni al contorno sia in merito ad alcuni set di proprietà. Tale obiettivo è un presupposto essenziale affinché venga facilitata e favorita l'interoperabilità tra software BIM e software per l'analisi Fem.

Sul lungo termine, inoltre, sarebbe interessante approfondire e accrescere il potenziale dei formati OpenFEM, attualmente poco utilizzati nell'interoperabilità in quanto non sufficientemente maturi. Se si riuscisse a trovare un formato OpenFEM vincente, come può essere IFC per l'OpenBIM, questo autore crede che il problema dell'interoperabilità tra i due modelli verrebbe praticamente risolto.

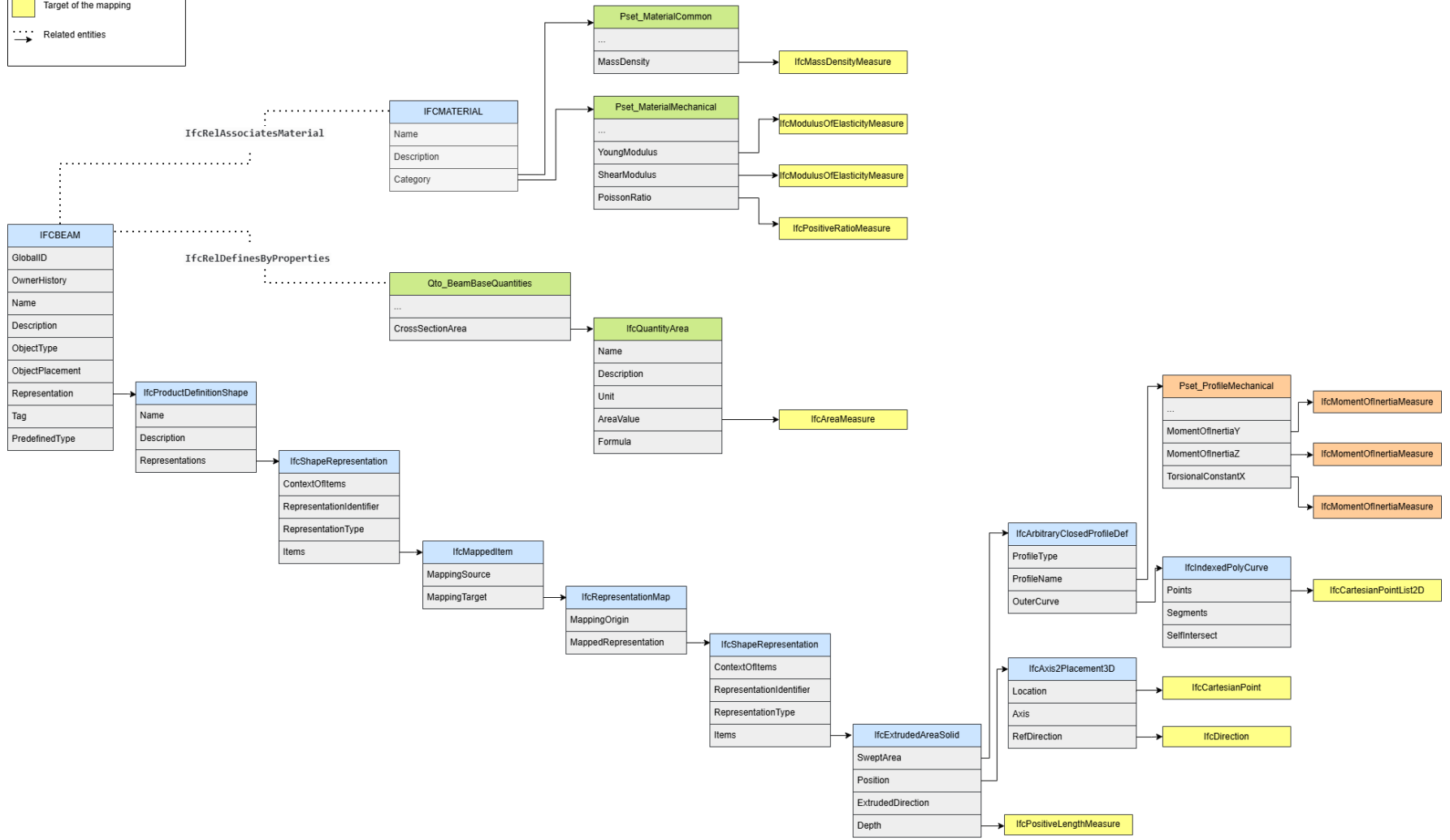
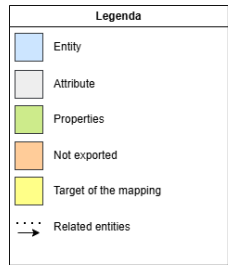
Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM
IFC e NASTRAN
Alice Pin

Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM IFC e NASTRAN

Alice Pin



Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM IFC e NASTRAN
 Alice Pin



APPENDICE 2

SCRIPT

```
# 0. Importo librerie
import ifcopenshell
import math

# 1. Creo le costanti che mi servono per riempire il file Nastran
GRID_GRID = "GRID"
GRID_ID = None
GRID_CP = 0
GRID_X = None
GRID_Y = None
GRID_Z = None

MAT1_MAT1 = "MAT1"
MAT1_MID = None
MAT1_E = None
MAT1_G = None
MAT1_NU = None
MAT1_RHO = None

PBEAM_PBEAM = "PBEAM"
PBEAM_PID = None
PBEAM_MID = None
PBEAM_Area = None
PBEAM_I1 = None
PBEAM_I2 = None
PBEAM_J = None

CBEAM_CBEAM = "CBEAM"
CBEAM_EID = None
CBEAM_PID = None
CBEAM_GA = None
CBEAM_GB = None
CBEAM_X1 = None
CBEAM_X2 = None
CBEAM_X3 = None

# 2. Definisco il percorso per leggere il file ifc di input
try:
    # NOTA: Percorso mantenuto come da originale, modificare se necessario
```

Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM IFC e NASTRAN

Alice Pin

```
ifc_file = ifcopenshell.open("C:\\Users\\alice\\Università degli Studi di Padova\\Tesi - Pin_BIMtoFEM - Pin_BIMtoFEM\\3_Sperimentazione\\2026.03.25
Script\\input.ifc")
except OSError as e:
    print(f"Errore nell'apertura del file 'input.ifc': {e}")
    exit()

# 3. Liste Nastran
nastran_grids = []
nastran_mats = []
nastran_pbeams = []
nastran_cbeams = []

# 4. Contatori
grid_id_counter = 0
mat_id_counter = 0
pbeam_id_counter = 0
cbeam_id_counter = 0

# 5. Dizionari per univocità
unique_nodes = {}
unique_materials = {}
unique_profiles = {}

# 6. Elementi IFC
elements = ifc_file.by_type("IfcBeam") + ifc_file.by_type("IfcColumn")

for element in elements:

    # Inizializzo variabili di supporto per l'elemento corrente
    current_ga_id = None
    current_gb_id = None
    current_pid = None
    depth_val = 0.0
    vec_x, vec_y, vec_z = 0.0, 0.0, 0.0

    # 7 & 13. Estrazione coordinate GA, Depth e Orientamento (X1, X2, X3)
    if element.Representation:
        if element.Representation.is_a('IfcProductDefinitionShape'):
            for representation in element.Representation.Representations:
                if representation.is_a('IfcShapeRepresentation'):
                    for item in representation.Items:
                        if item.is_a('IfcMappedItem'):
                            mapping_source = item.MappingSource
                            if mapping_source and mapping_source.is_a('IfcRepresentationMap'):
                                mapped_rep = mapping_source.MappedRepresentation
                                if mapped_rep and mapped_rep.is_a('IfcShapeRepresentation'):
                                    for shape_item in mapped_rep.Items:
                                        if shape_item.is_a('IfcExtrudedAreaSolid'):
                                            # --- Estrazione DEPTH ---
                                            depth_val = shape_item.Depth

                                        placement = shape_item.Position
                                        if placement and placement.is_a('IfcAxis2Placement3D'):
```

```

# --- Estrazione GA (Location) ---
cartesian_point = placement.Location
if cartesian_point:
    gx, gy, gz = cartesian_point.Coordinates

    # 8. Popolo GRID GA
    node_key_a = (gx, gy, gz)
    if node_key_a not in unique_nodes:
        grid_id_counter += 1
        unique_nodes[node_key_a] = grid_id_counter
        nastran_grids.append(f"{GRID_GRID},{grid_id_counter},{GRID_CP},{gx},{gy},{gz}")
    current_ga_id = unique_nodes[node_key_a]

# --- Calcolo GB (GA + Depth lungo tutte e tre le direzioni x, y, z locali) ---
# Otteniamo la direzione dell'estrusione (generalmente Z locale)
extrusion_direction = shape_item.ExtrudedDirection.DirectionRatios
ex, ey, ez = extrusion_direction

# Otteniamo le direzioni di riferimento locali X e Y
# Assumiamo che RefDirection sia l'asse X locale
ref_direction_x = placement.RefDirection.DirectionRatios if placement.RefDirection else (1.0, 0.0, 0.0)

# Calcoliamo la direzione Y locale come il prodotto vettoriale di Z locale e X locale
# Per un sistema destrorso, Y = Z x X
ref_direction_y_x = ez * ref_direction_x[1] - ey * ref_direction_x[2]
ref_direction_y_y = ex * ref_direction_x[2] - ez * ref_direction_x[0]
ref_direction_y_z = ey * ref_direction_x[0] - ex * ref_direction_x[1]
ref_direction_y = (ref_direction_y_x, ref_direction_y_y, ref_direction_y_z)

# Normalizziamo le direzioni per usarle come versori
len_ext_dir = math.sqrt(ex**2 + ey**2 + ez**2)
dir_z_norm = (ex/len_ext_dir, ey/len_ext_dir, ez/len_ext_dir) if len_ext_dir > 0 else (0.0, 0.0, 1.0)

len_ref_x = math.sqrt(ref_direction_x[0]**2 + ref_direction_x[1]**2 + ref_direction_x[2]**2)
dir_x_norm = (ref_direction_x[0]/len_ref_x, ref_direction_x[1]/len_ref_x, ref_direction_x[2]/len_ref_x) if len_ref_x >
0 else (1.0, 0.0, 0.0)

len_ref_y = math.sqrt(ref_direction_y[0]**2 + ref_direction_y[1]**2 + ref_direction_y[2]**2)
dir_y_norm = (ref_direction_y[0]/len_ref_y, ref_direction_y[1]/len_ref_y, ref_direction_y[2]/len_ref_y) if len_ref_y >
0 else (0.0, 1.0, 0.0)

# Calcoliamo le coordinate del punto GB aggiungendo 'depth_val' lungo tutte e tre le direzioni locali
gx_end = gx + depth_val * (dir_x_norm[0] + dir_y_norm[0] + dir_z_norm[0])
gy_end = gy + depth_val * (dir_x_norm[1] + dir_y_norm[1] + dir_z_norm[1])
gz_end = gz + depth_val * (dir_x_norm[2] + dir_y_norm[2] + dir_z_norm[2])

node_key_b = (gx_end, gy_end, gz_end)
if node_key_b not in unique_nodes:
    grid_id_counter += 1
    unique_nodes[node_key_b] = grid_id_counter
    nastran_grids.append(f"{GRID_GRID},{grid_id_counter},{GRID_CP},{gx_end},{gy_end},{gz_end}")

```

```

        current_gb_id = unique_nodes[node_key_b]

        # --- Estrazione Orientamento X1, X2, X3 (IfcDirection) ---
        # Tipicamente RefDirection rappresenta l'asse X locale del profilo
        if placement.RefDirection:
            vec_x, vec_y, vec_z = placement.RefDirection.DirectionRatios
            pass

# 9. Estrazione Materiale
current_mat_id = None
if hasattr(element, "HasAssociations"):
    for association in element.HasAssociations:
        if association.is_a('IfcRelAssociatesMaterial'):
            material_select = association.RelatingMaterial
            if material_select.is_a('IfcMaterial'):
                mat = material_select
                if hasattr(mat, "HasProperties"):
                    for mat_prop in mat.HasProperties:
                        if mat_prop.is_a('IfcMaterialProperties'):
                            if mat_prop.Name == "Pset_MaterialCommon":
                                for p in mat_prop.Properties:
                                    if p.Name == "MassDensity":
                                        MAT1_RHO = p.NominalValue.wrappedValue if p.NominalValue else None
                            elif mat_prop.Name == "Pset_MaterialMechanical":
                                for p in mat_prop.Properties:
                                    if p.Name == "YoungModulus":
                                        MAT1_E = p.NominalValue.wrappedValue if p.NominalValue else None
                                    elif p.Name == "ShearModulus":
                                        MAT1_G = p.NominalValue.wrappedValue if p.NominalValue else None
                                    elif p.Name == "PoissonRatio":
                                        MAT1_NU = p.NominalValue.wrappedValue if p.NominalValue else None

# 10. Popolo MAT1
mat_key = (MAT1_E, MAT1_G, MAT1_NU, MAT1_RHO)
if any(mat_key):
    if mat_key not in unique_materials:
        mat_id_counter += 1
        unique_materials[mat_key] = mat_id_counter
        e_val = MAT1_E if MAT1_E is not None else ""
        g_val = MAT1_G if MAT1_G is not None else ""
        nu_val = MAT1_NU if MAT1_NU is not None else ""
        rho_val = MAT1_RHO if MAT1_RHO is not None else ""
        nastran_mats.append(f"{MAT1_MAT1},{mat_id_counter},{e_val},{g_val},{nu_val},{rho_val}")
    current_mat_id = unique_materials[mat_key]

# 11. Estrazione PBEAM (Area e Inerzie)
pbeam_area = 0.0
if hasattr(element, "IsDefinedBy"):
    for definition in element.IsDefinedBy:
        if definition.is_a('IfcRelDefinesByProperties'):
            prop_set = definition.RelatingPropertyDefinition
            if prop_set.is_a('IfcElementQuantity'):

```

```

        for quantity in prop_set.Quantities:
            if quantity.is_a('IfcQuantityArea'):
                pbeam_area = quantity.AreaValue

# Geometria Profilo
base, height, diameter = 0.001, 0.001, 0.001
if element.Representation:
    for rep in element.Representation.Representations:
        for item in rep.Items:
            if item.is_a('IfcMappedItem'):
                map_rep = item.MappingSource.MappedRepresentation
                for shape_item in map_rep.Items:
                    if shape_item.is_a('IfcExtrudedAreaSolid'):
                        profile = shape_item.SweptArea
                        if profile.is_a('IfcArbitraryClosedProfileDef'):
                            polycurve = profile.OuterCurve
                            if polycurve.is_a('IfcIndexedPolyCurve'):
                                point_list = polycurve.Points
                                if point_list.is_a('IfcCartesianPointList2D'):
                                    coords = point_list.CoordList
                                    dists = []
                                    for i in range(len(coords)):
                                        p1, p2 = coords[i], coords[(i + 1) % len(coords)]
                                        dists.append(math.sqrt((p2[0]-p1[0])**2 + (p2[1]-p1[1])**2))
                                    base = min([d for d in dists if d > 1e-6] or [0.001])
                                    height = max(dists or [0.001])

                                    max_d = 0.0
                                    for i in range(len(coords)):
                                        for j in range(i + 1, len(coords)):
                                            d = math.sqrt((coords[i][0]-coords[j][0])**2 + (coords[i][1]-coords[j][1])**2)
                                            if d > max_d: max_d = d
                                    diameter = max_d if max_d > 0 else 0.001

# Calcolo I1, I2, J
if abs(height - diameter) > 1e-4:
    pbeam_i1 = (base * (height**3)) / 12
    pbeam_i2 = (height * (base**3)) / 12
    pbeam_j = (height * (base**3)) / 3
else:
    pbeam_i1 = (math.pi * (diameter**4)) / 64
    pbeam_i2 = (math.pi * (diameter**4)) / 64
    pbeam_j = (math.pi * (diameter**4)) / 32

# 12. Popolo PBEAM
if current_mat_id is not None:
    profile_key = (pbeam_area, pbeam_i1, pbeam_i2, pbeam_j, current_mat_id)
    if profile_key not in unique_profiles:
        pbeam_id_counter += 1
        unique_profiles[profile_key] = pbeam_id_counter
        nastran_pbeams.append(f"{PBEAM_PBEAM},{pbeam_id_counter},{current_mat_id},{pbeam_area},{pbeam_i1},{pbeam_i2},{pbeam_j}")
    current_pid = unique_profiles[profile_key]

```

```

# 14. Popolo CBEAM (Elemento trave)
if all(v is not None for v in [current_pid, current_ga_id, current_gb_id]):
    cbeam_id_counter += 1
    nastran_cbeams.append(f"{CBEAM_CBEAM},{cbeam_id_counter},{current_pid},{current_ga_id},{current_gb_id},{vec_x},{vec_y},{vec_z}")

# --- STAMPE DI CONTROLLO FINALI E GENERAZIONE FILE ---
print("\nCONVERSIONE COMPLETATA")
print("-----")
print(f"Nodi generati: {len(nastran_grids)}")
print(f"Materiali generati: {len(nastran_mats)}")
print(f"Proprietà (PBEAM) generate: {len(nastran_pbeams)}")
print(f"Elementi (CBEAM) generati: {len(nastran_cbeams)}")
print("-----")

# Definizione del nome del file di output
output_filename = "output.nas"

try:
    with open(output_filename, "w", encoding="utf-8") as f:
        # Intestazione opzionale per il file Nastran
        f.write("$ NASTRAN BULK DATA FILE generato da script Python\n")
        f.write("BEGIN BULK\n")

        # 1. Scrittura GRID
        f.write("$\n$ SEZIONE GRID (Nodi)\n")
        for grid_str in nastran_grids:
            f.write(grid_str + "\n")

        # 2. Scrittura MAT1
        f.write("$\n$ SEZIONE MAT1 (Materiali)\n")
        for mat_str in nastran_mats:
            f.write(mat_str + "\n")

        # 3. Scrittura PBEAM
        f.write("$\n$ SEZIONE PBEAM (Proprietà sezioni)\n")
        for pbeam_str in nastran_pbeams:
            f.write(pbeam_str + "\n")

        # 4. Scrittura CBEAM
        f.write("$\n$ SEZIONE CBEAM (Connettività elementi)\n")
        for cbeam_str in nastran_cbeams:
            f.write(cbeam_str + "\n")

        # Chiusura file Nastran
        f.write("$\n$ ENDDATA\n")

    print(f"\nFile '{output_filename}' creato con successo nella cartella dello script.")

except IOError as e:
    print(f"\nErrore critico nella creazione del file '{output_filename}': {e}")

# (Opzionale) Visualizzazione a video dei vettori generati

```

Individuazione di protocolli interoperabili per il digital twin in ambito strutturale: proposta di una metodologia basata su standard openBIM IFC e NASTRAN
Alice Pin

```
visualizza_anteprema = input("\nVuoi visualizzare l'anteprema dei dati a video? (s/n): ")
if visualizza_anteprema.lower() == 's':
    print("\n--- ANTEPRIMA DATI NASTRAN ---")
    for g in nastran_grids: print(g)
    for m in nastran_mats: print(m)
    for p in nastran_pbeams: print(p)
    for c in nastran_cbeams: print(c)
```


BIBLIOGRAFIA

- A. Rönneblad. (2003). *Product Models for Concrete Structures*. Luleå University of Technology.
- A. Osello. (2012). *Il futuro del disegno con il BIM per ingegneri e architetti*. D. Flaccovio.
- B. Riccardo. (2023). *Protocolli openBIM per la digitalizzazione del degrado e degli interventi di recupero strutturale in edifici in muratura portante*. Università degli studi di Padova.
- C. Eastman, P. Teicholz, R. Sacks, K. Liston. (2016). *Il BIM: Guida completa al Building Information Modeling* (2nd ed.). Hoepli Editore.
- C. van Treeck, E. R. (2006). Dimensional reduction of 3D building models using graph theory and its application in building energy simulation. *Engineering with Computers*, 23, 109–122.
- E. Kennett. (2007). *National Building Information Modeling Standard Version 1 - Part 1: Overview, Principles, and Methodologies*.
- E. C. Prada, L. B. López Vázquez. (2015). INFORMATION VISUALIZATION AS AN EFFECTIVE TOOL FOR SOLVING PHYSICS PROBLEMS. *Universidad Politécnica de Madrid*, 4207–4213.
- F. Beirnaert, A. Lippens. (2018). *Analysis of the interoperability from BIM to FEM*. Tampere University of Applied Sciences.
- G. Marcolin. (2017). *Structural Information Modeling: proposta di sviluppo di standard informativi*. Università degli studi di Padova.
- G. Lobato de Araújo. (2018). *Il BIM applicato all'impresa a servizio della progettazione integrale*. Politecnico di Torino.
- G. Acampa, C. M. Parisi, M. Grasso. (2018). *Prove tecniche di interoperabilità: dalla teoria alla pratica*.
- H. Zhou, X. He, X. L. Leng. (2021). Development and Application of a Revit-ANSYS Model Transformation Interface. *IOP Conference Series: Earth and Environmental Science*, 861(7). <https://doi.org/10.1088/1755-1315/861/7/072058>
- L. Han, G. Ge, H. Z., L. Yu-Shen, S. Yan, G. Ming. (2019). *MVDLite: A Light-weight Representation of Model View Definition with Fast Validation for BIM Applications*.
- I Quaderni - n. 38 Linee Guida Allo Sviluppo Di Software Riutilizzabile Multiuso Nella Pubblica Amministrazione, Centro Nazionale per l'Informatica nella Pubblica Amministrazione (2009).

- IDABC. (2004). *European Interoperability Framework*.
- Inojosa, L. da Silveira Pirillo, K. de Araújo Vilanova. (2023). BIM interoperability in the maintenance planning process for existing buildings. *Journal of Building Pathology and Rehabilitation*, 45(8.1).
- J. Chaskalovic. (2008). *Finite Elements Methods for Engineering Sciences*. Springer Berlin Heidelberg.
- J. Jia, J. Gao, W. Wang, L. Ma, J. Li, Z. Zhang. (2022). An Automatic Generation Method of Finite Element Model Based on BIM and Ontology. *Buildings*, 12(11). <https://doi.org/10.3390/buildings12111949>
- K. Surender Kumar, N. Lingeshwaran, Syed Hamim Jeelani. (2020). Analysis of residential building with STAAD. Pro & ETABS. *Materials Today: Proceedings* 33., 33, 1221–1228.
- K. Luku. (2023). *L'EFFICACIA DEL BIM NEL PROCESSO EDILIZIO: Analisi critica della curva di MacLeamy*. Politecnico di Torino.
- L. Muratore. (2018). *BIM e interoperabilità con il programma di calcolo strutturale Midas Gen*. Politecnico di Torino.
- M. Chernyavsky, H. Kestel. (2023). Sydney football stadium redevelopment. An efficient design, manufacturing and on-site assembly approach to a complex form found roof shell structure. *Proceedings of IASS Annual Symposia, IASS 2023 Melbourne Symposium*, 1–12.
- M. Leonardi, J. Granja, D. Oliveira, M. Azenha. (2024). Scalable BIM based open workflow for structural analysis of masonry building aggregates. *Computers and Structures*, 297. <https://doi.org/10.1016/j.compstruc.2024.107321>
- M. L. Longo. (2025). *La gestione delle attività manutentive attraverso i processi BIM Applicazione al caso del Liceo Darwin di Rivoli*. Politecnico di Torino.
- M. Cristani. (2002). *Introduzione all'Informatica*. CEDAM.
- M. F. Bado, D. Tonelli, F. Poli, D. Zonta, J. R. Casas. (2022). Digital Twin for Civil Engineering Systems: An Exploratory Review for Distributed Sensing Updating. *Sensors*, 3168(22.9).
- M. Zucco. (2022). *L'adozione del BIM nei piccoli studi di progettazione. Definizione ed analisi di workflow metodologici*. Politecnico di Torino.
- Ministero Della Giustizia, Pub. L. 59, Formati (2014).
- MSC.Software Corporation. (2006). *MD Nastran 2006 Quick Reference Guide*. www.mscsoftware.com
- N. Ferro, S. Micheletti, S. Perotto. (2017). *Anisotropic Mesh Adaptation for Crack Propagation Induced by a Thermal Shock*.

- NBS. (2019). *National BIM Report 2019 - The definitive industry update*.
- Norme Tecniche per Le Costruzioni, D.M. 17 Gennaio 2018 (2018).
- D. Oreni, R. Brumana, S. Della Torre, F. Banfi, L. Barazzetti, M. Previtali (2014). Survey turned into HBIM: The restoration and the work involved concerning the Basilica di Collemaggio after the earthquake (L'Aquila). *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(5), 267–273. <https://doi.org/10.5194/isprsannals-II-5-267-2014>
- P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone. (2014). Basi di dati. In *Treccani* (4th ed.). McGraw-Hill Education.
- P. Crespi, A. Franchi, P. Ronca, N. Giordano, M. Scamardo, G. Gusmeroli, G. Schiantarelli. (2015). From BIM to FEM: the analysis of an historical masonry building. *Building Information Modelling (BIM) in Design, Construction and Operations*, 1, 581–592. <https://doi.org/10.2495/bim150471>
- R. Scherer, F. Lin. (2020). *An openBIM Based Data Mapping Methodology Towards Structural Analysis Using FEM*. <http://dx.doi.org/10.46421/2706-6568.37.2020.paper030>
- R. Wang, T. Yongjing. (2021, January 26). Research on Parsing and Storage of BIM Information Based on IFC Standard. *IOP Publishing Ltd*. <https://doi.org/10.1088/1755-1315/643/1/012172>
- S. Aliprandi. (2010). *Aperti standard! : interoperabilità e formati aperti per l'innovazione tecnologica*. Ledizioni.
- S. Gerbino, L. Cieri, C. Rainieri, G. Fabbrocino. (2021). On bim interoperability via the ifc standard: An assessment from the structural engineering and design viewpoint. *Applied Sciences (Switzerland)*, 11(23). <https://doi.org/10.3390/app112311430>
- S. Quintero Botero, L. A. Gutiérrez Bucheli. (2018). *Evaluation of Interoperability in Construction Programs Using the IFC 4 Universidad de Los Andes Department of Civil and Environmental Engineering*. <https://hdl.handle.net/1992/39169>
- S. Aliprandi. (2014). *Aperti standard!* Ledizioni.
- T. Rácz, T. Olofsson. (2009). Interoperability Challenges of an Engineering Software Provider. *Managing It in Construction/Managing Construction for Tomorrow ER*, 26.
- UNI, Pub. L. UNI 11337:2009, Edilizia e opere di ingegneria civile - Criteri di codificazione di opere e prodotti da costruzione, attività e risorse - Identificazione, descrizione e interoperabilità (2009).

- UNI, Pub. L. UNI 11337:2017, Edilizia e opere di ingegneria civile - Criteri di codificazione di opere e prodotti da costruzione, attività e risorse - Identificazione, descrizione e interoperabilità (2017).
- UNI, Pub. L. UNI EN ISO 19650-1:2019, Organizzazione e digitalizzazione delle informazioni relative all'edilizia e alle opere di ingegneria civile, incluso il Building Information Modelling (BIM) - Gestione informativa mediante il Building Information Modelling - Parte 1: Concetti e principi (2019).
- V. Miotto. (2022). *BIM Interoperability from the Structural Engineering Perspective: BIM-FEM Indirect Link for Reinforcement of Concrete Elements*. Politecnico di Milano.
- Y. Zou, F. Hu, H. Yang, J. Cai, H. Pan, Q. Zhang. (2024). Detailed Design of Special-Shaped Steel Structures Based on DfMA: The BIM-FEM Model Conversion Method. *Buildings*, 18. <https://doi.org/10.3390/buildings14051320>
- Z. Xu, Z. Rao, V. J. L. Gan, Y. Ding, C. Wan, X. Liu. (2019). *Developing an Extended IFC Data Schema and Mesh Generation Framework for Finite Element Modeling*.

SITOGRAFIA

- Autodesk Revit. (2026). *Sito ufficiale*.
<https://help.autodesk.com/view/RVT/2026/ITA/?guid=GUID-C81929D7-02CB-4BF7-A637-9B98EC9EB38B> Ultimo accesso in data: 14/04/2026
- Autodesk Revit Guida. (2015). *Informazioni sul modello analitico strutturale*. Autodesk Revit.
<https://help.autodesk.com/view/RVT/2015/ITA/?guid=GUID-05CA5628-12C6-456C-B0B9-D922D22B67D0> Ultimo accesso in data: 14/04/2026
- Autodesk Revit Guida. (2017). *Esportazione in formato IFC (Industry Foundation Classes)*. Autodesk Revit.
<https://help.autodesk.com/view/RVT/2017/ITA/?guid=GUID-6EB68CEC-6C17-4B16-A509-30537F666C1F> Ultimo accesso in data: 14/04/2026
- buildingSMART International. (2026). *buildingSMART International*.
<https://www.buildingsmart.org/> Ultimo accesso in data: 14/04/2026
- GeometryGym. (2026). *Introduction Introduction to the GeometryGym Plug-In for Revit*. GeometryGym.
<https://technical.geometrygym.com/revit/introduction> Ultimo accesso in data: 14/04/2026
- Ing. Daniele Corti. (2013). *PROGETTAZIONE DI UNA BASE DI DATI*.
<https://www.ingdanielecorti.it/> Ultimo accesso in data: 14/04/2026
- MSC.Software Corporation. (2006). *MD Nastran 2006 Quick Reference Guide*.
www.mscsoftware.com Ultimo accesso in data: 14/04/2026
- Strand7. (2026). *Sito ufficiale di Strand7*. Strand7. <https://www.strand7.com/>
- Vocabolario Treccani. (2026). *Software*. Istituto della Enciclopedia Italiana Treccani.
<https://www.treccani.it/vocabolario/software/> Ultimo accesso in data: 14/04/2026

INDICE DELLE TABELLE

Tabella 1 – Elenco software FEM.....	23
Tabella 2 – GRID = (GRID, ID, CP, X, Y, Z).....	45
Tabella 3 - MAT1 = (MAT1, ID, E, G, NU, RHO)	46
Tabella 4 – PBEAM = (PBEAM, PID, MID, Area, I1 (zz), I2 (yy), J)	46
Tabella 5 - CBEAM = (CBEAM, EID, PID, GA, GB, X1, X2, X3)	46
Tabella 6 - PLOAD = (PLOAD1, SID, EID, TYPE, SCALE, X1, P1, X2, P2).....	46
Tabella 7 - SPC = (SPC1, SID, C, Gi).....	47
Tabella 8 - Attributi IfcColumn. Fonte: (buildingSMART International, 2026)	51
Tabella 9 - Set di proprietà Qto_ColumnBaseQuantities. Fonte: (buildingSMART International, 2026).....	54
Tabella 10 - Elenco entità coinvolte nello schema IfcColumn IFC4X3_ADD2. Fonte: (buildingSMART International, 2026)	55
Tabella 11 - Attributi IfcExtrudedAreaSolid. Fonte: (buildingSMART International, 2026).....	58
Tabella 12 - Attributi IfcAxis2Placement3D. Fonte: (buildingSMART International, 2026).....	60
Tabella 13 - - Momenti d'Inerzia e Modulo di rigidezza torsionale. Fonte: autore.....	67

INDICE ICONOGRAFICO

Figura 1 – Rapporto tra Standard e Formato. Fonte: (Simone Aliprandi, 2014).....	7
Figura 2 - Schema di un modello dati. Fonte: (Elena Cerro Prada & Luis B. López Vázquez, 2015)	8
Figura 3 - Livelli di modellazione dei dati. Fonte: (Ing. Daniele Corti, 2013).....	9
Figura 4 - Rappresentazione del concetto di BIM. Fonte: (Michele Zucco, 2022)	11
Figura 5 - Le dimensioni del BIM. Fonte: (Gabriel Fagundes Lobato de Araújo, 2018).....	11
Figura 6 - LOD. Fonte: (Gabriel Fagundes Lobato de Araújo, 2018)	12
Figura 7 - Schema concettuale dell'utilizzo di IFC. Fonte: (Inojosa et al., 2023).....	14
Figura 8 - Evoluzione temporale delle versioni di IFC. Fonte: (Giovanna Acampa et al., 2018)	15
Figura 10 - Architettura dello schema dei dati di IFC4X3_ADD2. Fonte: (buildingSMART International, 2026).....	18
Figura 11 - Utilizzo dell'IFC Structural Domain nella modellazione 3D. Fonte: (Raimar Scherer & Fangzheng Lin, 2020).....	19
Figura 12 - Mesh frutto della discretizzazione di un oggetto. Fonte: (N. Ferro et al., 2017)	21
Figura 13 - Esempio di analisi FEM. Fonte: (K. Surender Kumar, 2020).....	22
Figura 14 - Interoperabilità in edilizia. Fonte: (B. Riccardo, 2023).....	25
Figura 16 - Evoluzione temporale della normativa in ambito BIM. Fonte: (Maria Laura Longo, 2025).....	26
Figura 17 - Combinazioni delle azioni. Fonte: (Norme Tecniche per Le Costruzioni, 2018)	27
Figura 18 - Errore nelle connessioni tra elementi analitici. Fonte: (L. Muratore, 2018).....	34
Figura 19 – Rappresentazione geometrica del modello architettonico, strutturale e analitico. Fonte: (T. Rácz & T. Olofsson, 2009).....	37
Figura 20 - Connessioni BIM e FEM. Fonte: (L. Muratore, 2018).....	38
Figura 21 – Esempio database relazionale IFCBeam. Fonte: (Ruoxiao Wang & Tang Yongjing, 2021).....	39
Figura 22 - Schema modello dati Nastran. Fonte: (MSC.Software Corporation, 2006)	45
Figura 23 - Logo di Strand7. Fonte: (Strand7, 2026).....	48
Figura 24 - Architettura IFCColumn nello Standard 4x3. Fonte: autore.....	50

Figura 25 - Zoom architettura IfcColumn in IFC4X3_ADD2. Fonte: autore.....	51
Figura 26 - Zoom architettura IfcMaterial in IFC4X3_ADD2. Fonte: autore.....	53
Figura 27 - Zoom architettura IfcQuantityArea in IFC4X3_ADD2. Fonte: autore.....	54
Figura 28 - Zoom architettura IfcProductDefinitionShape in IFC4X3_ADD2. Fonte: autore.....	55
Figura 29 - Zoom architettura IfcShapeRepresentation in IFC4X3_ADD2. Fonte: autore.....	57
Figura 30 - IfcExtrudedAreaSolid. Fonte: (buildingSMART International, 2026).....	58
Figura 31 - Zoom architettura IfcArbitraryClosedProfileDef in IFC4X3_ADD2. Fonte: autore..	59
Figura 32 - Zoom architettura IfcAxis2Placement3D in IFC4X3_ADD2. Fonte: autore.	60
Figura 33 - Architettura IfcBeam nello Standard 4x3. Fonte: autore.....	61
Figura 34 - Percorso GRID - X, Y, Z. Fonte: autore.	63
Figura 35 - GRID - IfcCartesianPoint. Fonte: autore.....	63
Figura 36 - MAT1 - IfcMaterial. Fonte: autore.	64
Figura 37 - PBEAM, Area - IfcAreaMeasure. Fonte: autore.	65
Figura 38 - PBEAM - Pset_ProfileMechanical. Fonte: autore.....	66
Figura 39 - Pset_ProfileMechanical to IfcMomentOfInertiaMeasure. Fonte: autore.....	66
Figura 41 - PBEAM - IfcCartesianPointList2D. Fonte: autore.	67
Figura 42 - CBEAM GA, GB - IfcExtrudedAreaSolid. Fonte: autore.	68
Figura 43 - CBEAM X1, X2, X3 - IfcExtrudedAreaSolid. Fonte: autore.....	69
Figura 44 - Popolarità del software Revit nel mondo. Fonte: (NBS, 2019)	73
Figura 45 - Modello fisico e modello analitico di Revit. Fonte: (Autodesk Revit Guida, 2015) .	74
Figura 46 - Impostazioni di esportazione IFC da Revit. Fonte: autore.	77
Figura 47 - Impostazione di esportazione IFC con GeometryGym. Fonte: autore.....	78
Figura 48 - Colonna a base quadrata modellata in Revit. Fonte: autore.	80
Figura 49 - Colonna a base circolare modellata in Revit. Fonte: autore.	81
Figura 50 - Colonna a base quadrata inclinata modellata in Revit. Fonte: autore.....	81
Figura 51 - Caso studio 1.C due colonne. Fonte: autore.....	82
Figura 52 - Trave modellata in Revit. Fonte: autore.	83
Figura 53 - Telaio modellato in Revit. Fonte: autore	84
Figura 54 - Codice Caso Studio 1. Fonte: autore.....	85
Figura 55 - Risultato output Nastran. Fonte: autore.....	87

Figura 56 – Risultati Caso studio 1. Fonte: autore.....	88
Figura 57 - Risultati Caso studio 1.B. Fonte: autore.....	88
Figura 58 - Risultati Caso studio 1.C. Fonte: autore.....	89
Figura 59 - Risultati Caso studio 2. Fonte: autore.....	89
Figura 60 - Risultato 1 Caso studio 3. Fonte: autore.....	90

RINGRAZIAMENTI

Desidero prima di tutto esprimere la mia gratitudine verso il professor Carlo Zanchetta e l'Ing. Giorgia Marcellino, che hanno permesso la realizzazione di questa tesi. Vi ringrazio per la vostra disponibilità e per il vostro supporto. Vi ringrazio, inoltre, per avermi guidata nello sviluppo della ricerca, aiutandomi a trovare la giusta via per concludere in bellezza questi anni universitari.

Ringrazio tutti i docenti che ho incontrato in questi cinque anni nelle aule accademiche, per avermi trasmesso molte nozioni e competenze, ma soprattutto per avermi fatta crescere come persona e come professionista. Ho grande stima nei vostri confronti e vi sarò sempre grata per aver svolto con passione il vostro ruolo, instillandomi la curiosità necessaria a superare con impegno ogni difficoltà che ho dovuto fronteggiare nel percorso di studi.

Ringrazio la mia famiglia, che mi ha sostenuta contribuendo ciascuno a modo suo. In particolare, ringrazio mia mamma e mio papà per avermi cresciuta mostrandomi che lavorare significa impegno e sacrificio, ma anche tanta gioia e soddisfazione laddove si scelga col cuore. Inoltre, vi ringrazio per avermi dato l'opportunità di dedicarmi interamente agli studi secondo le mie attitudini, sostenendomi sempre. Ringrazio mia sorella, i nonni Wally e Walter, i nonni Luciana e Giorgio, gli zii Giorgia e Ale, per aver contribuito con consigli, domande, suggerimenti, proposte, che mi hanno permesso di arrivare a questo punto della mia vita sicura di me e motivata ad affrontare l'inizio di un nuovo capitolo con le idee chiare.

Sono grata agli amici e ai colleghi che ho incontrato a Trento, a Sheffield e a Padova, che mi hanno permesso di vivere intensamente ogni esperienza attraversando insieme difficoltà e delusioni, compensate da risate, divertimento e soddisfazioni. Ringrazio inoltre le persone che mi conoscono da prima dell'università e che mi sono sempre rimaste vicine. Sono particolarmente felice di aver condiviso con Elisa, Giulia, Elisabetta ed Elena le emozioni di questi anni, sapendo di trovare in loro una spalla su cui poter contare.

Infine ringrazio Riccardo, per aver creduto in me, per avermi spronata, per avermi capita fino in fondo e avermi dato tanto conforto quando ne avevo bisogno. Ti ringrazio infinitamente per aver affrontato questo percorso tosto stando sempre al mio fianco, portando tanta pazienza nel sopportare i miei scleri, ma anche festeggiando con allegria le piccole conquiste di questo periodo. È stato stupendo crescere insieme passetto dopo passetto e non vedo l'ora di proseguire la strada che ci aspetta, restando sempre in squadra con te.