Tesi di Laurea Magistrale in Ingegneria Edile

Sessione di Laurea: 24 ottobre 2017

BIM: ANALISI DEL FORMATO IFC E OTTIMIZZAZIONE DELLA GESTIONE DEI DATI AL FINE DEL CODE CHECKING STRUTTURALE

Caso studio: Palazzo della Regione Piemonte, Torino

RELATORE
Anna Osello

AUTORE
Gioele Lauro

CORRELATORI
Adriano Castagnone, S.T.A. Data s.r.l.
Matteo Del Giudice

Ai miei genitori, a Carola e a tutte le persone che mi hanno sostenuto.

Un ringraziamento particolare alla Prof.ssa Anna Osello, all'Ing. Adriano Castagnone e ai loro collaboratori.

Abstract

In un mondo delle costruzioni in continua evoluzione, il BIM (Building Information Modeling) non viene più considerato come un'assoluta novità ma, allo stesso tempo, è lontano dall'essere una metodologia affermata e diffusa nel mercato italiano. Il progresso tecnologico induce sempre più professionisti a confrontarsi con un cambiamento metodologico difficile in un settore ancora troppo statico. Con l'intento di esaminare una piattaforma nata per colmare i problemi di interoperabilità tra software BIM, vengono analizzate le potenzialità del formato IFC. In questo contesto, viene svolta un'analisi per la lettura del formato ai fini del *code checking* strutturale. Definite le regole da verificare, viene descritto il metodo utilizzato per la creazione di funzioni in grado di elaborare i dati presenti all'interno di un file e di restituire le informazioni necessarie alle successive verifiche normative. Le funzioni, scritte in linguaggio vb.NET, vengono applicate ad una porzione del modello strutturale del grattacielo della Regione Piemonte sito in Torino. Nell'analisi vengono illustrate le criticità e le accortezze necessarie per ottenere un'efficiente esportazione IFC per il *code checking* strutturale. Attraverso l'utilizzo di flow chart, viene posta particolare attenzione all'analisi del flusso delle operazioni da eseguire per ottenere nel modo più efficiente le informazioni cercate.

In a constantly developing construction world, Building Information Modeling (BIM) is no longer an absolute novelty but, at the same time, it is far from being a well-established and widespread methodology in the Italian system. Technological advancement induces more and more professionals to deal with a difficult methodological change in an even too static sector. With the aim of examining a platform created to bridge the interoperability among various BIM software, it is analyzed the potential of the IFC format. In this context, the analysis is performed for read the format and executes structural *code checking*. Defined the rules to be verified, the method for creating functions capable of processing the data within a file has been described with the aim of obtaining information on subsequent regulatory controls. Functions, written in vb.NET, are applied to a portion of the structural model of the skyscraper of the Piedmont Region in Turin. The analysis illustrates the criticalities and caution necessary to obtain efficient IFC export for structural *code checking*. Using flow charts, particular attention is paid to the flow analysis of the operations to obtain the information more efficiently.

Sommario

Lista delle abbreviazioni **7**

1. INTRODUZIONE **8**

1.1 Premessa *8*

1.2 Obiettivi *9*

1.3 Caso Studio - Grattacielo Regione Piemonte *9*

1.3.1 Inquadramento e descrizione del progetto 10

1.3.2 Il progetto 11

1.3.3 La realizzazione 13

1.4 BIM: evoluzione della progettazione *14*

1.4.1 BIM in Italia 15

1.5 Introduzione al code checking strutturale *17*

1.6 IFC per l'interoperabilità *18*

1.6.1 Storia 18

1.6.2 BuildingSMART 19

1.6.3 BuildingSMART Standards 19

1.6.4 Linguaggio di scrittura IFC 20

1.6.5 IFC Model Classes Breakdown 24

1.6.6 Architettura IFC 26

1.6.7 Schema IFC2x3, IFC4 30

1.6.8 Model View Definition 30

2. Metodologia **31**

2.1 Analisi e lettura di un file IFC *31*

2.1.1 Rappresentazione degli spazi 33

2.1.2 Rappresentazione dei componenti edilizi 34

2.1.3 Definizione dei sistemi di riferimento 37

2.1.4 Esempio IfcBeam 38

2.2 IFC applicato al caso studio *41*

2.2.1 Modellazione, obiettivi e materiale fornito 41

2.2.2 Esportazione del modello 43
2.2.3 Impostazioni d'esportazione IFC 43
2.2.4 Modifiche nella modellazione per l'esportazione e il code checking 45
2.3 Ottimizzazione della gestione dei dati per il code checking strutturale 47
2.3.1 Trasformazioni dei sistemi di riferimento 50
2.3.2 Associazione elemento-armatura 57
2.3.3 Distinzione armatura longitudinale-trasversale 64
2.3.4 Restituzione dei risultati 68
3. Risultati 73
3.1 Modellazione ed esportazione IFC 73
3.2 Output 74
3.2.1 Report 74
3.2.2 Rappresentazione grafica nel software del code checking 76
4. Conclusioni 77
4.1 Criticità, limiti e future implementazioni delle funzioni 77
4.2 Il formato IFC, sviluppi futuri 78
4.3 Possibili scenari e sviluppi della progettazione BIM 79
5. ALLEGATI 81
5.1 Specifiche sulla rappresentazione dei diagrammi di flusso 81
5.2 Flusso dell'applicazione delle funzioni 82
5.2.1 Diagramma di flusso 82
5.2.2 Codice vb.NET 83
5.3 Trasformazioni dei sistemi di riferimento 85
5.4 Associazione elemento-armatura 86
5.5 Distinzione armatura longitudinale e trasversale 87
Software utilizzati 89
Bibliografia 89
Normativa di riferimento 90
Sitografia 90

Indice delle figure

Figura 1 - Flusso delle informazioni nella comunicazione 8
Figura 2 - Riqualificazione area Avio-Oval, www.sedeunica.regione.piemonte.it 11
Figura 3 - Modello Torre Regione Piemonte, www.fuksas.it/it/Nuova-Sede-Regione-Piemonte_Turin ... 12
Figura 4 - Avanzamento dei lavori, gennaio 2014 13
Figura 5 - BIM Dictionary, attivo a ottobre 2017 17
Figura 6 - Standard BuildingSMART, www.buildingsmart-tech.org 20
Figura 7 - Simbolo Entity data type 21
Figura 8 - Simbolo Enumeration data type 21
Figura 9 - Simbolo Defined data type 22
Figura 10 - Simbolo Select data type 22
Figura 11 - Simbolo Simple data type 23
Figura 12 - Simbolo Attribute 23
Figura 13 - Simbolo Supertype/Subtype 23
Figura 14 - Esempio di rappresentazione EXPRESS-G 24
Figura 15 - Grafico dell'architettura IFC 27
Figura 16 - Architettura IFC, resource layer 28
Figura 17 - Architettura IFC, core layer 28
Figura 18 - Definizione dell'entità IfcObject tra Kernel e Core Extension 29
Figura 19 - Architettura IFC, interoperability layer 29
Figura 20 - Architettura IFC, domain layer 29
Figura 21 - Flusso metodologico 31
Figura 22 - Inheritance Graph dell'entità IfcAxis2Placement3D 33
Figura 23 - Gerarchia delle entità spaziali 33
Figura 24 - Scomposizione degli elementi dall'entità IfcProduct 34
Figura 25 - Specifiche EXPRESS dell'entità IfcProduct 34
Figura 26 - Inheritance Graph dell'entità IfcShapeRepresentation 35
Figura 27 - Attributi dell'entità IfcMaterial 36
Figura 28 - Inheritance Graph dell'entità IfcLocalPlacement 37
Figura 29 - Rappresentazione grafica dell'entità IfcLocalPlacement 38
Figura 30 - Modellazione parametrica di una trave 39
Figura 31 - Apertura del file IFC attraverso IfcQuickBrowser 40
Figura 32 - Apertura del file IFC attraverso Solibri Model Viewer 40
Figura 33 - Albero delle entità in IfcQuickBroswer 41
Figura 34 - Estratto di una tavola esecutiva degli elementi strutturali 42
Figura 35 - Revit Extensions per la modellazione dell'armatura 42
Figura 36 - Flusso delle operazioni di modellazione 43
Figura 37 - Configurazioni d'esportazione IFC 44
Figura 38 - Esempio di associazione tra categorie di Revit ed entità IFC 45
Figura 39 - Esempio di modellazione non ottimizzata per il code checking strutturale 45
Figura 40 - Modellazione parametrica della trave di raccordo per trefoli come telaio strutturale 46
Figura 41 - Sovrapposizione dei modelli per analizzare le interferenze 46
Figura 42 - Esempio dell'entità IfcRelAggregates 47
Figura 43 - Esempio in cui le armature vengono classificate come NOTDEFINED 48
Figura 44 - Flusso di calcolo dei dati mancanti 48
Figura 45 - Flusso metodologico seguito per la determinazione delle funzioni 49
Figura 46 - Serie di sistemi di riferimento 50

Figura 47 - Trasformazioni di coordinate tra due sistemi di riferimento 51
Figura 48 - Trasformazione di coordinate a seguito della rotazione degli assi 53
Figura 49 - Trasformazioni di coordinate per una serie di sistemi di riferimento 54
Figura 50 - Definizione dell'orientamento del sistema di riferimento locale 56
Figura 51 - Rappresentazione dell'armatura con una sola entità IfcReinforcingBar 58
Figura 52 - Analisi della relazione tra elemento e armatura 60
Figura 53 - Analisi della relazione tra elemento e armatura sul piano xy 60
Figura 54 - Analisi dell'armatura sul piano xy 62
Figura 55 - Armatura descritta da due punti esterni al volume 62
Figura 56 - Individuazione dei nuovi punti da verificare 64
Figura 57 - Analisi della classificazione dell'armatura sul piano e nello spazio 65
Figura 58 - Individuazione del piano contenente 3 punti 67
Figura 59 - Definizione del report con l'utilizzo del programma KompoZer 69
Figura 60 - Visualizzazione del codice HTML 70
Figura 61 - Estratto di codice che rappresenta una riga della tabella 71
Figura 62 - Vista in pianta del satellite del settimo piano in Revit Architecture, non in scala 73
Figura 63 - Visualizzazione del file IFC con Solibri Model Viewer 74
Figura 64 - Estratto dal report dei risultati 75
Figura 65 - Report Satelliti, attivo a ottobre 2017 75
Figura 66 - Trave la cui sezione non è stata riconosciuta 76
Figura 67 - Trave non descritta attraverso l'estrusione 76
Figura 68 - Interazione tra gli algoritmi di calcolo e il programma del code checking 77
Figura 69 - Livello di maturità del formato IFC, www.buildingsmartcanada.ca 79
Figura 70 - Grafico del rendimento nel tempo 80

Lista delle abbreviazioni

- BIM - Building Information Modeling
- CAD - Computer Aided Design
- IFC - Industry Foundation Classes
- AEC - Architecture Engineering and Construction
- FM - Facility Management
- IAI - International Alliance for Interoperability
- ISO - International Organization for Standardization
- UNI - Ente italiano di normazione
- pGI - piano per la Gestione Informativa
- NTC - Norme Tecniche per le Costruzioni
- SMC - Solibri Model Checker
- LOD - Level Of Development
- CDE - Common Data Environment
- vb .NET - Visual Basic .NET
- HTML - HyperText Markup Language

1. INTRODUZIONE

1.1 Premessa

Prima di analizzare nello specifico uno strumento che permette di connettere in maniera efficace le informazioni nel settore delle costruzioni, è necessario focalizzare l'attenzione sulla comunicazione in un sistema complesso. Con il termine "**comunicazione**" si intende il processo e i metodi di trasmissione di un'informazione tra le parti che compongono un sistema. Il mezzo che permette la comunicazione è il **messaggio** che dev'essere elaborato secondo opportune regole condivise. Gli strumenti che permettono la trasmissione di un messaggio dipendono dall'ambiente scelto per la comunicazione.

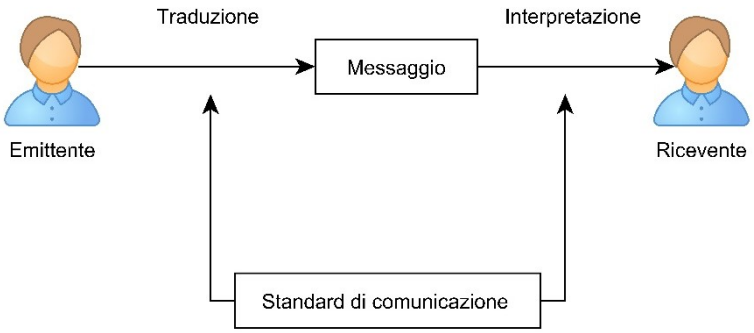


Figura 1 - Flusso delle informazioni nella comunicazione

Applicando questi concetti in un ambiente informatico, le informazioni contenute all'interno di un programma vengono tradotte in un linguaggio condiviso tramite l'applicazione rigorosa di standard definiti. Il contenuto del messaggio potrà quindi essere letto soltanto dagli applicativi in grado di reinterpretarne il contenuto attraverso le medesime regole.

Come approfondito in seguito, il metodo di comunicazione globalmente condiviso nel settore delle costruzioni si basa sull'adozione della metodologia *Building Information Modeling* (BIM) attraverso il formato *Industry Foundation Classes* (IFC).

1.2 Obiettivi

In un contesto dove la comunicazione ha un ruolo fondamentale per lo scambio di informazioni, viene svolta un'analisi metodologica introducendo l'importanza del BIM nel settore *Architecture Engineering Construction* (AEC). L'adozione di un determinato approccio metodologico permette di ottimizzare la gestione e l'interoperabilità delle informazioni di un sistema complesso.

Come vettore di comunicazione viene analizzato il formato IFC. Ormai globalmente riconosciuto dalle principali case software, il formato IFC nasce con l'intento di creare una piattaforma in grado di descrivere qualsiasi componente del settore AEC attraverso la definizione tridimensionale e informativa tipica del BIM. L'analisi viene condotta sia da un punto di vista prettamente teorico, sia attraverso esempi che ne facilitano la comprensione.

Sfruttando le potenzialità del formato, la seguente trattazione ha l'obiettivo di analizzare la sequenza logica delle operazioni che permettono di ottimizzare la gestione dei dati contenuti all'interno di un file. Un'accurata analisi ha messo in evidenza la mancanza esplicita di alcune informazioni indispensabili per eseguire i controlli da normativa (*code checking* strutturale). L'obiettivo finale è quello di fornire, attraverso l'implementazione di funzioni nel linguaggio vb.NET, un metodo capace di ricavare i dati in modo da poter verificare la corrispondenza del progetto in riferimento alle *Norme Tecniche per le Costruzioni* (NTC).

Gli algoritmi di calcolo automatico vengono applicati al modello tridimensionale parametrico del grattacielo della Regione Piemonte. In parallelo all'implementazione dei dati, viene analizzata la tecnica di modellazione per ottenere un file IFC ottimizzato per essere processato dalle funzioni per il *code checking*.

Infine, il lavoro svolto permette di valutare le potenzialità del file IFC e degli strumenti di controllo normativo. Allo stesso tempo, vengono sottolineate le criticità operative analizzando, dove possibile, le soluzioni adottate.

1.3 Caso Studio - Grattacielo Regione Piemonte

Le analisi e i ragionamenti metodologici precedentemente effettuati perderebbero di significato se non applicati ad un caso studio reale. Lo studio teorico, in parallelo al riscontro delle informazioni su un progetto reale, permette un apprendimento delle nozioni più efficace ed un processo di sviluppo che risponde alle esigenze del mercato. Come vedremo in seguito, non è immaginabile implementare algoritmi di calcolo automatico non

corrispondenti alle esigenze reali della progettazione. Nel caso specifico, la modellazione di un caso studio ha due obiettivi principali:

- definire degli "standard" di modellazione per l'esportazione IFC e il seguente *code checking* strutturale;
- testare e migliorare gli algoritmi di calcolo su un caso reale.

Per lo sviluppo di questa trattazione è stato indispensabile lavorare in parallelo sulla modellazione e sullo sviluppo delle funzioni di calcolo. Questa scelta metodologica ha permesso l'instaurarsi di un processo virtuoso di continui feedback che hanno portato correzioni e migliorie in entrambi i lavori.

Il caso studio in esame è il Grattacielo della Regione Piemonte, sito in Torino, progettato dall'architetto Massimiliano Fuksas.

1.3.1 Inquadramento e descrizione del progetto

Nel 2001 la giunta regionale presieduta da Enzo Ghigo bandì un concorso internazionale per la progettazione e realizzazione di un edificio in grado di ospitare gli uffici della Regione Piemonte. La gara venne vinta dall'architetto Massimiliano Fuksas, portatore di un progetto che prevedeva la realizzazione di una torre parallelepipeda di 100 metri d'altezza nell'area dell'ex-Materferro, tra corso Lione e corso Mediterraneo. Con il passaggio nel 2005 alla giunta presieduta da Mercedes Bresso, il progetto subì un ricollocamento nell'area ex Fiat Avio nei pressi della vecchia area industriale di Nizza Millefonti. Nel 2006 fu quindi assegnato all'architetto Fuksas l'incarico di rivedere il progetto per inserirlo nel nuovo contesto e, nell'ambito di un progetto di riqualificazione dell'intero quartiere, gli fu assegnato l'incarico di ridisegnare l'urbanistica dell'area circostante. Nel frattempo fu approvata una variante del piano regolatore che restrinse al solo centro storico della città il divieto di realizzare edifici più alti della Mole Antonelliana (alta 167,5 m). Questo ha permesso allo studio Fuksas di presentare il progetto di un grattacielo alto il doppio rispetto a quello precedentemente progettato.

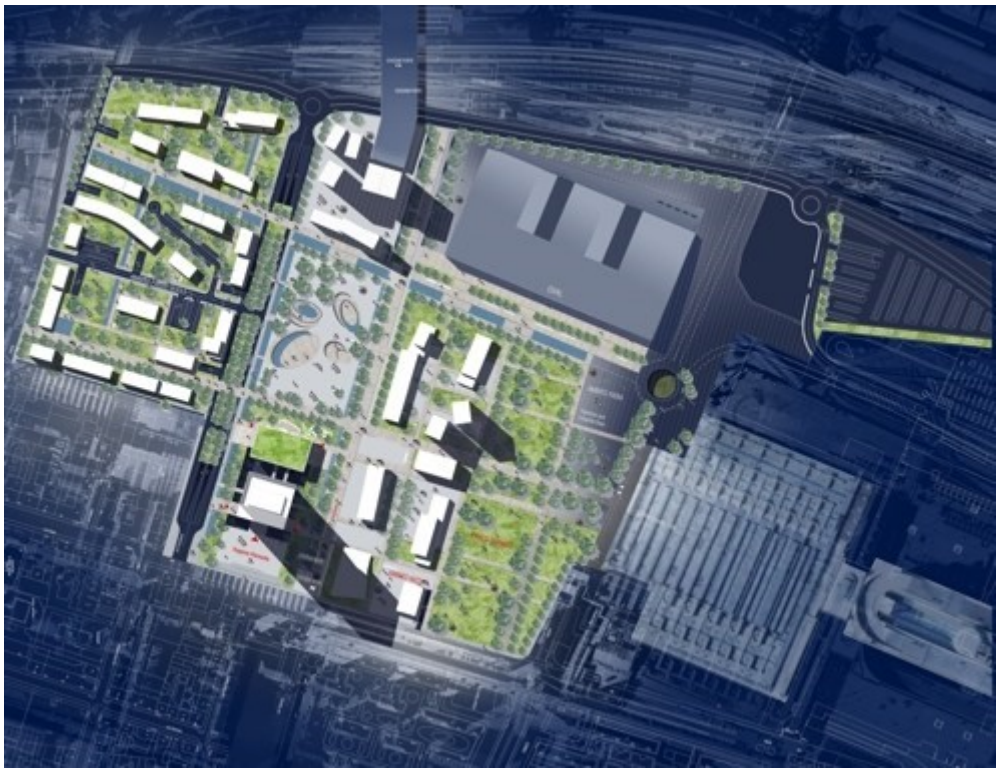


Figura 2 - Riqualificazione area Avio-Oval, www.sedeunica.regione.piemonte.it

1.3.2 Il progetto

Come anticipato, il progetto prevede la riqualificazione dell'intera area e in particolare la realizzazione dei seguenti manufatti:

- Torre degli uffici della Regione Piemonte: rappresenta l'edificio principale del progetto, composta da un edificio a pianta quadrata avente 42 livelli con interpiano di 4,27m per un'altezza totale di 205 m rispetto alla quota della piazza di ingresso;
- corte interrata: ai livelli interrati -1 e -2 sono stati progettati degli spazi a servizio della struttura principale; sono stati predisposti un centro stampa, l'archivio generale, la mensa, un centro fitness, locali per bar, uffici vari e i locali per le centrali tecnologiche per la gestione dei vari impianti;
- centro servizi: composto da un edificio indipendente di 4 livelli collegato alla torre da una passerella vetrata sospesa. Comprende un centro congressi, la biblioteca e alcuni luoghi per l'utenza esterna;
- asilo nido aziendale a servizio dei figli dei dipendenti al piano terra del centro servizi.

Concentrando l'attenzione sul progetto della Torre, lo studio Fuksas ha pensato ad un edificio a pianta quadrata di dimensioni di circa 45mx45m calcolando un affollamento previsto di 50-60 persone per ogni piano. Una delle peculiarità del progetto è la facciata posta sul lato est della costruzione: questa porzione di edificio, denominata "grande vuoto" è composta da una serie di satelliti differenti per ogni piano che si innestano nei solai. Alcuni elementi avranno il compito di raccordare i satelliti tra un piano e l'altro dando alla facciata un aspetto dinamico e innovativo. In questi spazi si troveranno delle sale riunioni e gli uffici degli assessori e dei dirigenti con relative segreterie.

Figura 3 - Modello Torre Regione Piemonte, www.fuksas.it/it/Nuova-Sede-Regione-Piemonte_Turin

Al centro della costruzione sono posti gli ascensori, i servizi igienici, i locali di servizio, le sale riunioni e le zone adibite ad ufficio. L'organizzazione degli uffici e la disposizione degli operatori è stata progettata cercando di ridurre al minimo i possibili riflessi sugli schermi e introducendo degli elementi schermanti all'interno della doppia facciata per ridurre i disturbi derivanti dall'esposizione diretta al sole.

Un altro aspetto fondamentale è l'attenzione posta alla **sostenibilità** dell'opera: in primo luogo è stato stabilito da contratto che tutti gli arredi dovranno essere il più possibile derivanti da materiali riciclati e che potranno essere riutilizzati (o riciclati) al termine del loro ciclo di vita; in secondo luogo è stata sperimentata una gestione digitale dei documenti senza dover quindi procedere alla stampa di numerosi elaborati cartacei.

1.3.3 La realizzazione

Nel 2011 iniziarono i lavori di costruzione che avrebbero portato, secondo le prime ipotesi, al trasferimento di assessori e dipendenti a partire dal 2015. Dopo una prima fase di scavo per le fondazioni, l'organizzazione cronologica del cantiere aveva previsto l'avanzamento di un piano per ogni settimana grazie all'utilizzo di particolari sistemi autorampanti.

Figura 4 - Avanzamento dei lavori, gennaio 2014

Con l'avanzamento dei lavori vi furono, però, alcuni intoppi operativi e giudiziari che oltre ad aver contribuito al crescere dei costi della costruzione, hanno portato a chiusure temporanee del cantiere e notevoli ritardi nella realizzazione dell'opera. Nell'autunno del 2015 vennero fermati tutti i lavori a causa del fallimento di una azienda appartenente al consorzio che gestiva il cantiere. Dopo aver ritrattato gli accordi con le ditte costruttrici, i lavori ripresero ufficialmente il 12 giugno 2017 con l'impegno di terminare la costruzione nei successivi 18 mesi.

Utilizzo del BIM

Nonostante la progettazione sia stata svolta secondo tecniche tradizionali, sono stati richiesti degli strumenti informatici più efficaci per il *facilities management* dell'intero edificio. A tale scopo un gruppo di ricerca del Politecnico di Torino sta lavorando alla messa a punto di un modello *As-built* con l'utilizzo del software BIM *Autodesk Revit 2017*. Il lavoro ha come obiettivo quello di creare un modello tridimensionale dell'edificio con all' interno tutte le informazioni per la gestione dei sistemi impiantistici (MEP) e architettonici.

La suddivisione del lavoro di modellazione tra i vari componenti delle squadre di progettazione avviene attraverso la definizione di regole ben precise. Per far questo

Autodesk ha implementato all'interno dei propri software degli strumenti (*WorkSet*) che permettono a più persone di lavorare su un unico modello (o su una serie di modelli federati) dove ogni utente avrà diritto di modificare solo gli elementi di propria competenza. Per rendere più efficiente e leggera la gestione è stato deciso di separare il modello centrale in diversi modelli a seconda della disciplina di modellazione. Sono quindi stati creati i modelli architettonico, strutturale, impiantistico e un modello contenente le tavole CAD. Per avere una visualizzazione di insieme e per gestire le possibili interferenze, tutti i modelli sono gestiti separatamente ma vengono collegati in un unico centrale.

1.4 BIM: evoluzione della progettazione

"The stark reality is that BIM will continue to significantly change the construction business environment and product manufactures who are not willing to adapt to support BIM process will face the reality that they will cease to remain competitive in the marketplace."

John I. Messner - Professor of Architectural Engineering Director - Penn State University, 2017

Il **Building Information Modeling** ha rappresentato una grande rivoluzione e un cambiamento di prospettiva rispetto alla progettazione tradizionale di un tempo. Con il termine BIM si intende una **metodologia** che consente di integrare in un unico modello digitale le informazioni utili in ogni fase del ciclo di vita di un edificio, permettendo ai diversi attori del processo edilizio che intervengono prima, durante e dopo il progetto, di integrare su un'unica base dati le proprie conoscenze. Il modello condivise è l'insieme di dati grafici e attributi sempre aggiornati e condivisi tra le diverse maestranze. La progettazione secondo la metodologia BIM permette il controllo attivo dell'intera opera: la progettazione, l'esecuzione e il *Facilities Management* (FM) dialogano con un unico modello aggiornato durante l'intero *Life Cycle*. Ciò permette ai professionisti di avere un valido strumento per pianificare gli interventi più efficaci.

Il grande lavoro effettuato dai produttori di software per adattarsi alla gestione del costruito con metodologie BIM, permette, non solo la digitalizzazione dell'intero processo, ma al tempo stesso, ha l'obiettivo di cambiare l'iter decisionale per ridurre gli errori di progettazione, i costi e i tempi per ottenere il miglior prodotto possibile.

Con l'avvento dei nuovi sistemi informatici la progettazione delle opere del settore AEC ha subito un notevole cambiamento non solo nelle tecniche di rappresentazione, ma anche nel modo di pensare e di pianificare un intervento.

1.4.1 BIM in Italia

Confrontando la diffusione del BIM nei paesi europei (soprattutto paesi scandinavi e Regno Unito) ed extra-europei (in particolare Stati Uniti, Canada e Singapore) con quanto avviene in Italia è possibile notare come in quest'ultima l'approccio alla progettazione BIM non sia altrettanto diffuso. Il modello di progettazione è ancora basato su **metodi tradizionali** e gli scambi di informazioni avvengono con documenti scollegati gli uni dagli altri. Possono essere diverse le motivazioni che stanno ostacolando la diffusione di questa nuova metodologia progettuale; tra queste vi possono essere:

- un ritardo legislativo nella definizione di standard condivisi e nell'obbligo dell'adozione del BIM;
- le difficoltà dovute alla non perfetta interoperabilità tra progettisti che utilizzano applicativi software differenti;
- un problema dovuto alle politiche commerciali di alcune case software che puntano ad utilizzare formati file proprietari e chiusi piuttosto che investire su formati di scambio aperti;
- una maggiore difficoltà nell'utilizzo di una metodologia BIM applicata al patrimonio costruito rispetto a alle nuove costruzioni;
- mancanza di un background culturale e degli incentivi necessari per avviare un cambiamento reale della progettazione.

Con l'intento di favorire questo cambiamento, la Commissione Europea STAND-INN (*Standard Innovation*) ha finanziato un progetto che ha portato l'UNI (Ente Italiano di Normazione) a pubblicare, nel 2008, un manuale contente le *best practices* a livello europeo per mettere in luce le potenzialità degli strumenti BIM.
Qualche anno più tardi, con la Direttiva Europea 2014/24/EU vengono introdotte nuove linee guida sull'utilizzo del sistema BIM nella progettazione e realizzazione soprattutto di opere pubbliche. Queste direttive hanno lo scopo di inserire gradatamente l'obbligo di utilizzo di questa nuova metodologia come requisito fondamentale per la partecipazione ad appalti pubblici. La direttiva fissa in trenta mesi il termine per il recepimento da parte di ogni Stato membro.

1.4.1.1 Quadro normativo

In Italia, con la pubblicazione in Gazzetta Ufficiale del D.Lgs. n.50/2016 detto anche "**Nuovo Codice degli Appalti Pubblici**", è stata introdotta per la prima volta la possibilità (non l'obbligatorietà) di inserire come requisito l'utilizzo del BIM per gli appalti pubblici. Ad oggi il settore delle costruzioni sta attendendo la pubblicazione di decreti attuativi e linee guida in grado di fare chiarezza su alcune situazioni progettuali che il testo non chiarisce. Tra questi troviamo le prime bozze del cosiddetto "*Decreto BIM*": si tratta della norma che dovrà definire le modalità e i tempi di introduzione dell'obbligatorietà, da parte delle stazioni appaltanti, dell'utilizzo di strumenti elettronici specifici per la modellazione edilizia e di infrastrutture.

A completamento del quadro normativo italiano per il BIM, assume significativa importanza la norma tecnica ***UNI 11337****:2017 Gestione digitale dei processi informativi delle costruzioni*. Il testo si compone complessivamente di 9 parti, alcune delle quali sono ancora oggi in via di definizione (parti 2 e 3 in riscrittura e parti 7, 8 e 9 di nuova introduzione).

Quelle ad oggi approvate sono le seguenti:

- Parte 1: Modelli, elaborati e oggetti informativi per prodotti e processi.
- Parte 4: Evoluzione e sviluppo informativo di modelli, elaborati ed oggetti.
- Parte 5: Flussi informativi nei processi digitalizzati.
- Parte 6: Linea Guida per la redazione del capitolato informativo.

La bozza del *Decreto BIM*, pur non facendo riferimento diretto alla norma tecnica 11337, ne sposa le definizioni e i documenti necessari al sostegno del processo informativo.

In un quadro normativo complesso, una ricerca pubblicata in occasione del "*2° Forum Internazionale OICE sul BIM, Roma – 22 giugno 2017*" ha dimostrato come non vi sia ancora assoluta chiarezza né tra chi si occupa della progettazione, né tra le stazioni appaltanti. La ricerca ha analizzato 37 bandi di gara aventi come richiesta la progettazione in BIM, pubblicati in Italia dal primo gennaio 2016 al 31 maggio 2017. L'analisi ha evidenziato le seguenti criticità:

- disomogeneità nelle richieste formulate dalle stazioni appaltanti;
- utilizzo con diversi significati del termine BIM;
- assenza di standard o *best practises* consolidate a livello internazionale;
- richiesta di capitolati informativi senza ulteriori specifiche sui criteri di redazione;
- mancanza del concetto di fruibilità delle informazioni nei processi decisionali.

1.4.1.2 Esempio di approccio progettuale al BIM

Un esempio virtuoso del corretto approccio della metodologia BIM in un bando di gara può essere rappresentato del progetto del **ponte "della Navetta"** a Parma dove, nel 2014, la piena del torrente Baganza ha portato al cedimento del vecchio ponte in mattoni. Dopo l'individuazione del nuovo progetto con un concorso di idee si è proceduto alla stesura del progetto definitivo con le tradizionali modalità operative. Come base di gara per la realizzazione del progetto esecutivo è stato scelto di sviluppare un modello BIM che ha permesso, già in fase di redazione, di porre rimedio ad alcuni errori difficilmente intercettabili con altri metodi. In aggiunta ai consueti documenti progettuali, sono stati resi disponibili tre modelli (architettonico, strutturale, MEP) nel formato aperto IFC. Inoltre è stata posta molta attenzione alla stesura dei requisiti del capitolato informativo secondo quanto definito dalla UNI 11337 parte 6. In questo documento troviamo:

- la precisione richiesta secondo i diversi livelli di sviluppo (LOD) con riferimento alla classificazione prevista dalla UNI 11337 parte 4;
- le procedure di coordinamento, verifica e validazione automatica del modello secondo la UNI 11337 parte 5;
- indicazioni sulle caratteristiche dell'ambiente di dati comune (CDE, Common Data Environment) in accordo con quanto definito nella UNI 11337;

Gli stessi professionisti che hanno contribuito alla stesura della normativa tecnica sul BIM hanno sottolineato la completezza e la cura con cui il bando è stato redatto; questo progetto si candida quindi ad essere l'apripista di una rivoluzione digitale e metodologica per i lavori pubblici.

1.4.1.3 BIM Dictionary

Con l'obiettivo di fare chiarezza sull'utilizzo della corretta terminologia, è stato recentemente tradotto nella lingua italiana il **BIM Dictionary**; grazie a questa piattaforma online si sta cercando di omogenizzare il più possibile l'utilizzo della corretta terminologia standardizzata. Le definizioni dei singoli termini e dei vari acronimi cercano di ridurre il più possibile quelle incomprensioni terminologiche che, come analizzato in precedenza, pongono un freno all'utilizzo efficace e consapevole di queste tecnologie.

Figura 5 - BIM Dictionary, attivo a ottobre 2017

1.5 Introduzione al code checking strutturale

Insieme alla verifica delle interferenze, il ***code checking*** rappresenta un metodo di **controllo automatico del progetto**. Lo sviluppo delle metodologie BIM e dei formati di scambio neutri (in particolare il formato IFC) hanno permesso l'introduzione di nuovi strumenti per la validazione automatica di determinate regole. In particolare, il *code checking* può essere definito come la verifica puntuale dei requisiti normativi di tutti gli elementi di un modello. I dati letti all'interno del file IFC vengono quindi opportunamente gestiti e confrontati con i limiti dettati dai codici di riferimento. A differenza di quanto veniva fatto con la progettazione tradizionale basata su CAD, il controllo normativo avviene in maniera automatica sull'intero progetto così da permettere di intercettare le possibili criticità.

Tra i software in commercio, il più diffuso per il controllo automatico è *Solibri Model Checker* (www.solibri.com). Questo applicativo permette di caricare un modello IFC ed effettuare le verifiche scegliendo i requisiti da analizzare tra una lista di regole di diverse discipline progettuali. Inoltre, l'utente può personalizzare alcuni parametri nel caso fossero necessari limiti diversi rispetto a quelli normativi.

Nella norma **UNI 11337-5**, descrivendo i flussi informativi del processo di digitalizzazione del comparto delle costruzioni, viene sottolineata l'importanza dell'utilizzo di sistemi in grado di eseguire controlli e operazioni automatiche sui modelli.

In particolare, la norma definisce tre diversi livelli di verifica:

- LV1: definisce una verifica interna e formale con l'obiettivo di analizzare la corretta modalità di produzione, consegna e gestione delle informazioni in base a quanto specificato nel Capitolato Informativo e nel piano per la gestione informativa (pGI).
- LV2: definisce le verifiche per accertare la leggibilità, tracciabilità e coerenza delle informazioni contenute nei modelli.
- LV3: definisce la verifica da effettuarsi a carico del committente. La norma lascia la possibilità di avvalersi di una figura esterna al processo per la verifica formale e sostanziale dei modelli.

Le seguenti analisi hanno l'obiettivo di fornire uno strumento utile per aiutare il progettista ad effettuare il *code checking* strutturale in ogni livello di verifica (in modo particolare LV1 e LV3).

Con l'obiettivo di creare un prodotto per il mercato italiano, è stata effettuata un'accurata analisi delle NTC. Lo studio ha avuto l'obiettivo di individuare la lista di regole e i requisiti minimi da verificare in un modello IFC per il *code checking* strutturale.

1.6 IFC per l'interoperabilità

1.6.1 Storia

Il formato IFC rappresenta il frutto della collaborazione tra diversi operatori nel settore AEC e ha l'intento di creare una piattaforma aperta per condividere informazioni tra le diverse figure del processo edilizio. L'"*International Alliance for Interoperability*" (IAI) è l'organizzazione internazionale che raggruppa le principali aziende software e che, oltre ad aver dato vita all'IFC, ha il compito di mantenerlo sempre aggiornato per migliorare costantemente la condivisione dei dati. L'evoluzione di questa tecnologia è strettamente collegata allo sviluppo metodologico del BIM nella gestione dell'intero ciclo di vita di un sistema edilizio.

La nascita e lo sviluppo di questa tecnologia è dovuta alla necessità di avere un formato di scambio aperto per risolvere i problemi di comunicazione tra le varie figure del processo edilizio. L'obiettivo è quello di rendere automatica l'interoperabilità tra i diversi applicativi software che, nell'attuale settore delle costruzioni, sono diventati strumenti indispensabile di affiancamento alla progettazione.

Autodesk fu la prima software house a raggruppare società di sviluppo con lo scopo di creare applicativi in grado comunicare con diversi programmi. Nel 1995 l'unione di 12 aziende americane permise di fondare una società no-profit con il nome di "Industry Alliance for Interoperability". Tre anni dopo l'organizzazione prese il nome, ancora attualmente noto, di "Alliance for Interoperability" (IAI). Ogni membro aveva il compito di mettere a disposizione la propria esperienza per definire standard condivisi ai fini di ottenere un

formato di scambio non appartenente ad una casa software ma condiviso e globalmente riconosciuto.

Il continuo aggiornamento delle caratteristiche di questo formato ha portato, a partire dal 1996, al rilascio di diverse versioni. In particolare dall'IFC1 si è passati all'IFC1.5.1, IFC2.0, IFC2x, IFC2x2, IFC2x3 fino ad arrivare all'attuale IFC4. Dalla versione IFC2x, pubblicata nel 2000, non è più stata cambiata l'intera struttura logica: l'organizzazione del formato rimane la medesima e le versioni rilasciate successivamente sono un arricchimento e miglioramento delle diverse specifiche. Questo permette di avere un formato più flessibile e soprattutto consente alle diverse software house di passare da una versione IFC ad un'altra in modo molto più agevole.

1.6.2 BuildingSMART

Nel 2008 l'*International Alliance for Interoperability* prese il nome di ***buildingSMART International*** senza però stravolgere i propri obiettivi fondanti. Il lavoro svolto in questi anni ha dato un grande contributo allo sviluppo delle tecnologie BIM in diverse parti del mondo.

BuildingSMART International è suddivisa in diverse organizzazioni territoriali con lo scopo di dare voce alle diverse esigenze e di stabilire metodi e strumenti standardizzati, globalmente condivisi. Tra i principali paesi che hanno preso parte all'organizzazione troviamo i paesi dell'Europa occidentale, i paesi scandinavi, Nord America, Australia, Giappone, Singapore, Regno Unito e Irlanda.

In Italia nel 2004 venne fondata un'organizzazione indipendente (che successivamente prese il nome di *buildingSMART Italia*) nata per dare un contributo allo sviluppo del Paese nel comparto edilizio. Nonostante il BIM non fosse ancora diffuso, sono rimasti inalterati i principi e la consapevolezza del bisogno di investire nello sviluppo di nuove tecnologie nel settore AEC.

Ad oggi, *buildingSMART Italia* si fa portavoce e promotore dello sviluppo del BIM mantenendo comunque una visione internazionale nell'adozione di strumenti e metodi standardizzati.

Ognuna delle organizzazioni territoriali *buildingSMART* contribuisce al mantenimento del formato IFC e allo studio di come standardizzare processi, flussi di lavoro e procedure BIM. È molto importante, in un mondo in continua evoluzione, avere la giusta elasticità per fornire strumenti sempre aggiornati e in grado di soddisfare le esigenze del mercato.

1.6.3 BuildingSMART Standards

L'adozione e la diffusione di standard è uno dei principali obiettivi che si è posta l'organizzazione fin dalla sua fondazione; a questo proposito le varie definizioni e metodologie sono state recepite e normate dall'*International Organization for Standardization* (ISO) nelle norme:

- IFD – ISO 12006-3, data models (Information Framework for Dictionary): specifica le informazioni del modello che possono essere inserite nel *Dictionary* indipendentemente dalla lingua utilizzata;
- IDM -ISO 29481, process definition (Information Delivery Manual): specifica la metodologia che integra i processi produttivi nell'edilizia con l'obiettivo di favorire la condivisione tra tutte le maestranze incluse nel processo;
- IFC - ISO 16739, dictionary terms (Industry Classes Foundation): definisce lo schema concettuale del formato IFC.

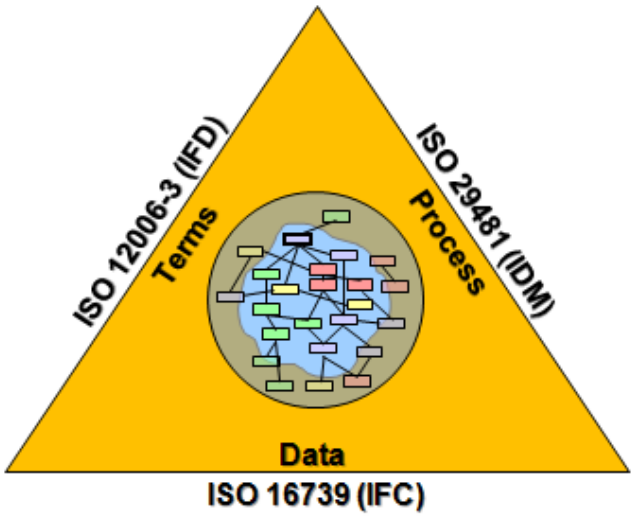


Figura 6 - Standard BuildingSMART, www.buildingsmart-tech.org

1.6.4 Linguaggio di scrittura IFC

1.6.4.1 Sintassi e struttura

Il formato IFC nasce e si sviluppa a partire dal linguaggio EXPRESS definito dalla norma "ISO 10303-11: Industrial automation system integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual". Questo linguaggio nonostante sia molto compatto, permette di gestire diversi tipi di dati e di specifiche. La struttura del file è di natura STEP (Standard for the Exchange of Product model data) definita dalla norma ISO 10303-21. In definitiva, un file IFC (avente estensione .ifc) è un file in formato ASCII scritto in linguaggio EXPRESS per rappresentare le informazioni definite dalla struttura STEP.

1.6.4.2 Standard STEP

L'utilizzo di questi standard permette di descrivere informazioni relative all'intero ciclo di vita di un prodotto (*product life cycle*) indipendentemente dall'applicativo software utilizzato. Da un punto di vista informatico, il vantaggio dell'uso della struttura STEP è rappresentato dal fatto che le informazioni vengono riconosciute da diversi protocolli e che tutti i dati possono essere condividi in un modello tridimensionale.

1.6.4.3 Linguaggio EXPRESS

EXPRESS rappresenta il linguaggio standard per la modellazione parametrica e per la descrizione di un prodotto utilizzando la struttura STEP. Sviluppato da Douglas Schenck e Peter Wilson a partire dal 1994, il linguaggio EXPRESS si sviluppa per la descrizione di prodotti industriali e grazie all'ampiamento delle librerie e delle funzioni, ora è esteso anche alla descrizione di edifici e infrastrutture. Può contare su due principali tipi di rappresentazione: la prima lessicale dove un file in formato ASCII ne descrive tutte le proprietà, l'altra è una rappresentazione grafica definita EXPRESS-G.
Il più generico elemento definito attraverso il linguaggio EXPRESS viene dichiarato con il termine "schema". Secondo la terminologia riportata anche nella normativa ISO, gli schemi di dati definiscono la struttura e i tipi di associazione esistenti fra i dati. Uno schema rappresenta un insieme di entità, attributi e istanze di tipo relazionali tra oggetti.

1.6.4.4 Rappresentazione EXPRESS-G

La rappresentazione EXPRESS-G è un metodo grafico per la definizione di uno schema dati definito da normativa ISO. Questo metodo di rappresentazione viene spesso utilizzato in supporto alla più diffusa rappresentazione lessicale del linguaggio EXPRESS; rappresentare le informazioni per via grafica può aiutare a rendere maggiormente leggibile i legami e le gerarchie tra le entità. Il linguaggio EXPRESS dispone di diversi "*Datatypes*" (tipi di dati) ai quali corrisponde una specifica rappresentazione simbolica EXPRESS-G. Al fine di comprendere la rappresentazione, è indispensabile definire e descrivere le caratteristiche dei diversi tipi di dati e le loro possibili relazioni.

1.6.4.5 Datatypes

- ***Entity data type***: è il tipo di dato più importante nel linguaggio EXPRESS, permette di definire le classi di oggetti del mondo reale e possono essere messe in relazione con altre entità attraverso un rapporto gerarchico sub-supertype oppure attraverso la definizione di attributi. Viene rappresentato semplicemente con un rettangolo.

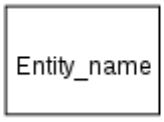


Figura 7 - Simbolo Entity data type

- ***Enumeration data type***: le enumerazioni sono una raccolta di dati di tipo stringa: per esempio l'enumerazione dei giorni della settimana sarà composta da sette stringhe indicanti ciascuna il nome di un giorno. Se opportunamente dichiarate, possono essere condivise da più schemi. Viene rappresentato con un rettangolo con linea tratteggiata avente un ulteriore segmento verticale tratteggiato a destra del nome dell'enumerazione.

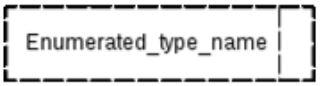


Figura 8 - Simbolo Enumeration data type

- *Defined data type*: è un datatype che deve essere ulteriormente specializzato da altri tipi di dati. A seconda della sua specializzazione può per esempio assumere le caratteristiche di stringa oppure di numero positivo. Viene rappresentato con un rettangolo a linea tratteggiata.

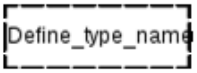


Figura 9 - Simbolo Defined data type

- *Select data type*: definisce una scelta o un'alternativa tra diverse opzioni. Viene utilizzato nella maggior parte dei casi per scegliere tra diversi entity type. Viene rappresentato con un rettangolo con linea tratteggiata con un segmento verticale tratteggiato alla sinistra del nome del tipo di dato.
- *Aggregation data type*: è un datatype che rappresenta una collezione di altri tipi di dato. In particolare un aggregation data type può essere di tipo:
    - ARRAY: rappresenta una collezione di dimensione fissata e ordinata; viene

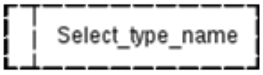


Figura 10 - Simbolo Select data type

    indicata con A[1:?];
    - BAG: rappresenta una collezione di elementi non ordinata e che permette duplicati; viene indicata con B[1:?];
    - LIST: rappresenta una collezione ordinata che non ammette elementi duplicati; viene indicata con L[1:?];
    - SET: rappresenta una collezione non ordinata e che non ammette elementi duplicati; viene indicata con S[1:?].

  Tra le parentesi quadre viene indicata la dimensione della collezione, il primo numero rappresenta la dimensione minima mentre il secondo indica la dimensione massima. Il carattere "?" indica che la collezione ha dimensione indeterminata.
- *Simple data type*: rappresentano le unità base del linguaggio EXPRESS e EXPRESS-G, infatti non è possibile dividerle in parti più piccole. Possono essere di diverso tipo a seconda della natura del dato che descrivono:
    - *string*: è il *simple data type* più utilizzato. Una stringa nel linguaggio EXPRESS può contenere qualsiasi carattere (codificato ISO 10646/Unicode) per una qualsiasi lunghezza;
    - *binary*: rappresenta un numero binario. Generalmente non viene molto utilizzato anche perché in alcune implementazioni la dimensione massima è limitata a 32 bit;
    - *boolean*: variabile che può assumere i valori tipici di una variabile booleana TRUE e FALSE;
    - *logical*: è simile ad una variabile booleana ma, a differenza di quest'ultima può assumere anche il valore UNKNOW;
    - *integer*: indica una variabile numerica di qualsiasi lunghezza anche se in molte implementazioni la sua dimensione viene ridotta a 32 bit;

- ***real***: nel linguaggio EXPRESS indica un numero reale di qualsiasi dimensione e precisione;
- ***number***: rappresenta la variabile per indicare numeri interi e reali;

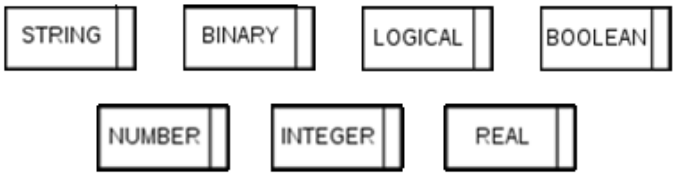


Figura 11 - Simbolo Simple data type

1.6.4.6 Attributes

Gli attributi permettono di aggiungere proprietà e relazioni alle entità.

Alcuni attributi si dicono "obbligatori" quando ad un'istanza dell'entità deve per forza essere assegnata una relazione; nella rappresentazione EXPRESS-G l'obbligatorietà dell'attributo viene rappresentata con una linea continua. Altri attributi vengono definiti "opzionali" quando l'assegnazione di una relazione non è obbligatoria; la loro rappresentazione avviene tramite una linea tratteggiata. In tutti i due i casi il nome che identifica la relazione viene scritto sopra la linea e un pallino ad una estremità ne indica la direzione principale.

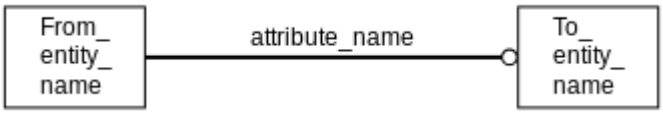


Figura 12 - Simbolo Attribute

Un'altra importante relazione tra entità può essere quella di *Supertype/Subtype*. Le entità sono, infatti, collegate tra di loro con predefinite relazioni gerarchiche.

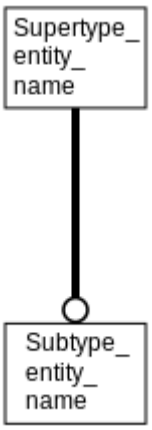


Figura 13 - Simbolo Supertype/Subtype

Nel linguaggio EXPRESS le entità definite "*subtype*" rappresentano la specializzazione delle entità "*supertype*". Le classi ad un livello gerarchico inferiore ereditano (principio informatico dell'ereditarietà) tutte le caratteristiche delle entità superiori e ne rappresentano una specializzazione grazie all'aggiunta di nuove istanze e/o attributi. Per esempio le classi *IfcWall*, *IfcBeam* e *IfcColumn* sono *subtype* della classe *IfcBuildingElement* in quanto ne ereditano le proprietà e, con l'aggiunta di nuove istanze, specializzano la classe. Questo legame viene raffigurato con una linea continua avente doppio spessore.

L'ultimo possibile tipo di attributo è quello "inverse": se una prima entità ha una relazione diretta, definita attraverso l'apposito attributo, con una seconda entità, allora quest'ultima entità potrà avere un attributo "inverse" nei confronti della prima. In altri termini la dichiarazione di un attributo inverso indica l'assegnazione di un attributo diretto all'entità a cui fa riferimento.

1.6.4.7 Confronto tra le rappresentazioni EXPRESS

Nel seguente esempio è possibile osservare la corrispondenza tra le due rappresentazioni EXPRESS, in particolare vengono descritte 3 entità aventi tra loro attributi differenti.

```
ENTITY ENTITY_1;
ATTRIBUTE_1 : INTEGER;
ATTRIBUTE_2 : STRING;
ATTRIBUTE_3 : ENTITY_2;
END_ENTITY;

ENTITY ENTITY_2;
END_ ENTITY;

ENTITY ENTITY_3;
SUBTYPE OF(ENTITY_1)
END_ ENTITY
```

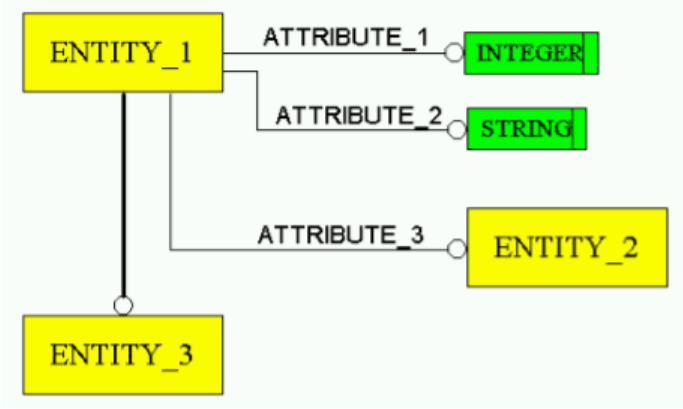


Figura 14 - Esempio di rappresentazione EXPRESS-G

1.6.5 IFC Model Classes Breakdown

Dopo aver descritto i possibili tipi di dati e le relazioni che li collegano, è importante analizzare le principali classi che caratterizzano il formato IFC. L'analisi viene fatta partendo dalla classe più astratta procedendo verso quelle più specializzate.

1.6.5.1 IfcRoot

La struttura IFC divide tutte le entità in entità "radicate" e in "non radicate". Le entità radicate dipendono da *IfcRoot* e hanno un concetto di identità cioè un insieme di attributi quali nome, descrizione e controllo di revisione. Le entità non radicate non hanno un'identità e le istanze esistono solo se si fa riferimento, direttamente o indirettamente, ad un'istanza radicata.

L'*IfcRoot*, la radice del modello, è la base grazie alla quale viene raggiunto sia il livello dell'interoperabilità che il dominio del modello IFC.

IfcRoot è suddiviso nei seguenti concetti astratti:

- *IfcObjectDefinition*: definisce i tipi di oggetti materiali nonché la generalizzazione di qualsiasi elemento trattato all'interno del modello IFC;
- *IfcRelationship*: definisce le relazioni tra gli oggetti e rappresenta la generalizzazione di tutte le relazioni tra gli oggetti;
- *IfcPropertyDefinition*: definisce le proprietà dinamicamente estensibili sugli oggetti e può essere definito come la generalizzazione di tutte le caratteristiche che possono essere assegnate agli oggetti.

L'insieme dei concetti astratti forma il 1° livello di specializzazione all'interno della gerarchia delle classi IFC.

1.6.5.2 IfcObject

È una classe astratta che rappresenta tutti gli elementi fisicamente tangibili (muri, travi, …), gli elementi fisicamente esistenti (spazi, elementi concettuali, …) e quelli concettuali (griglie, confini virtuali, …). Distingue i processi come le attività di lavoro, i controlli, le voci di costo e le risorse, la risorsa di lavoro, gli attori e le persone coinvolte nel processo di progettazione.

Un oggetto ottiene le informazioni di contesto dalle relazioni di cui è parte, dalle informazioni sulle proprietà e, se disponibili, dalle informazioni relative al tipo di oggetto specifico sottostante. Un oggetto può avere un tipo di descrittore informale assegnato, che denota un particolare tipo di ulteriore specifica dell'oggetto.

IfcObjectDefinition è suddiviso in "*presenze*" di oggetti e "*tipi*" di oggetti. *IfcObject* descrive la presenza dell'oggetto come l'installazione del prodotto con quel determinato numero di serie collocato fisicamente nello spazio edificio. *IfcTypeObject* cattura le definizioni di tipo, ad esempio il tipo di prodotto con quel numero di modello e quella determinata forma.

1.6.5.3 IfcElement

Definisce univocamente un oggetto fisico esistente appartenente al settore AEC. È possibile associare ad ogni elemento una lista di proprietà (come i materiali) e le relazioni tra gli elementi.

1.6.5.4 IfcBuildingElement

Rappresenta il *supertipo astratto* di tutti gli elementi edilizi come fondazioni, pavimenti, pilastri, etc. A questa classe fanno riferimento diverse relazioni inverse che permettono di definire relazioni spaziali, strutturali e materiali.

1.6.6 Architettura IFC

L'architettura dello standard IFC è stata sviluppata secondo i seguenti obiettivi:

- ottenere una struttura modulare;
- ottenere un framework per la condivisione delle informazioni tra le diverse discipline nel settore AEC/FM;
- ottenere un formato facilmente implementabile;
- permettere agli sviluppatori di riutilizzare le componenti software già sviluppate;
- mantenere la compatibilità tra le diverse versioni.

L'Object Model Architecture del formato IFC è sviluppato secondo un preciso "schema di modello". È possibile dividere la struttura in 4 livelli principali:

- *Domain layer*;
- *Interoperability layer*;
- *Core layer*;
- *Resource layer*.

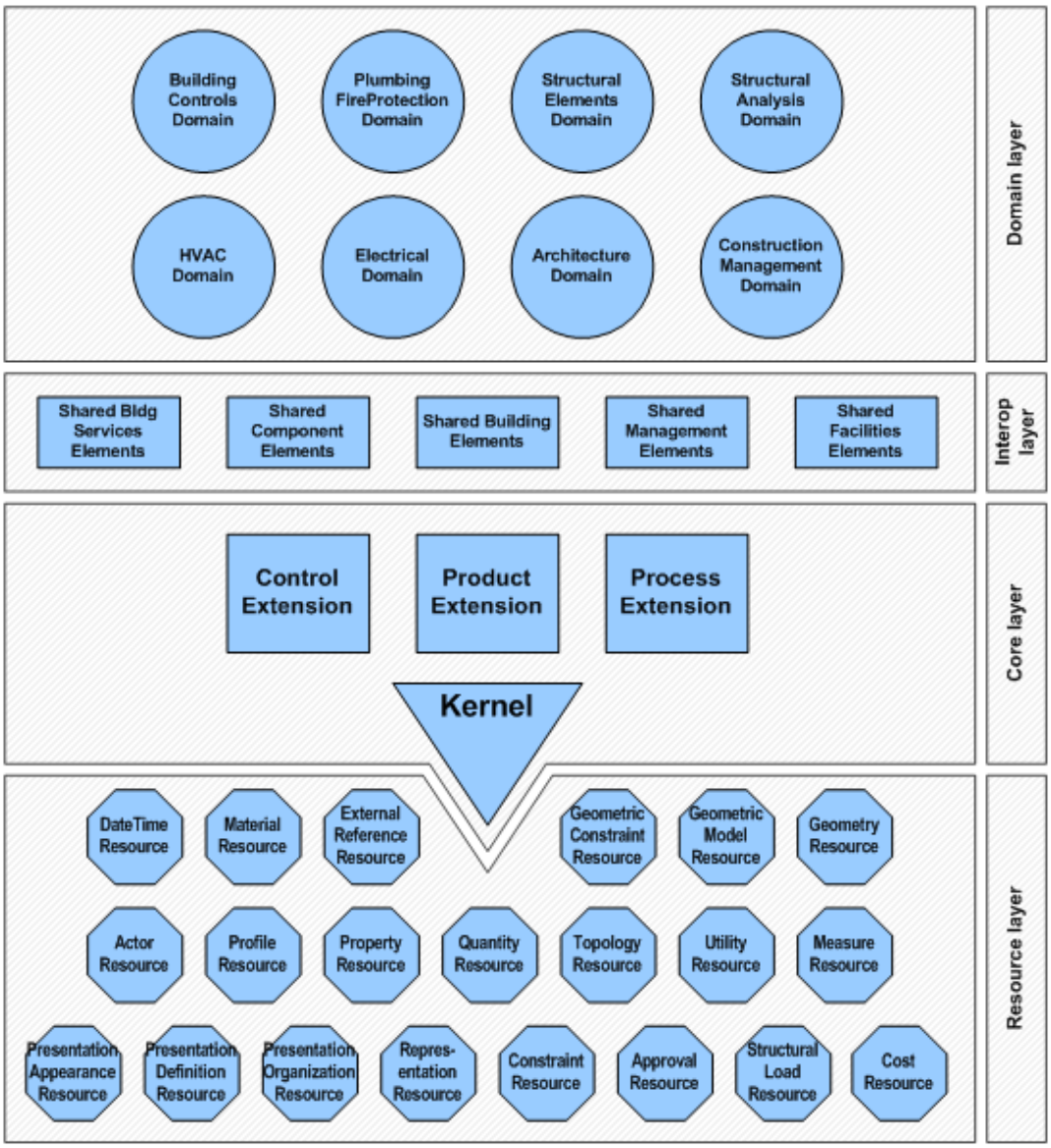


Figura 15 - Grafico dell'architettura IFC

I vari livelli sono legati gerarchicamente l'uno con l'altro in modo che le entità appartenenti ad un livello possano referenziare tutte le entità appartenenti ai livelli inferiori (principio di gravità).

1.6.6.1 Resource layer

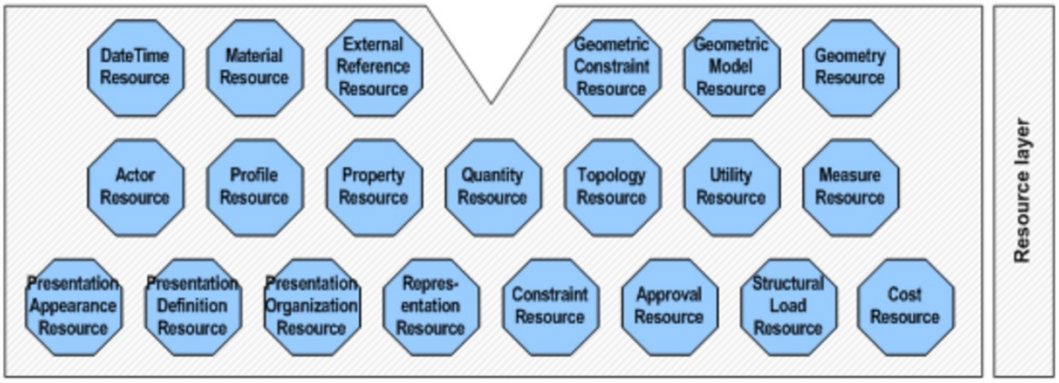


Figura 16 - Architettura IFC, resource layer

Rappresenta il livello gerarchicamente meno specializzato dello schema IFC. Le entità appartenenti al *Resource layer* non possono esistere in modo indipendente ma devono essere richiamate da una o più entità appartenenti allo stesso livello o a livelli superiori. Questi elementi non hanno le proprietà di identità (non viene definito un *GlobalID*) e non possono definire relazioni tra elementi gerarchicamente superiori. Per una gestione ottimale dei file IFC viene raccomandato di utilizzare, dove possibile, le stesse istanze per definire diversi elementi; ad esempio due linee (*IfcLine*) aventi lo stesso punto cartesiano di origine (*IfcCartesianPoint*), nella loro rappresentazione, dovranno far riferimento alla medesima istanza che ne rappresenta il punto. Questo meccanismo, nonostante aumenti la complessità nella lettura, permette di minimizzare il numero di informazioni riducendo le dimensioni del file.

1.6.6.2 Core layer

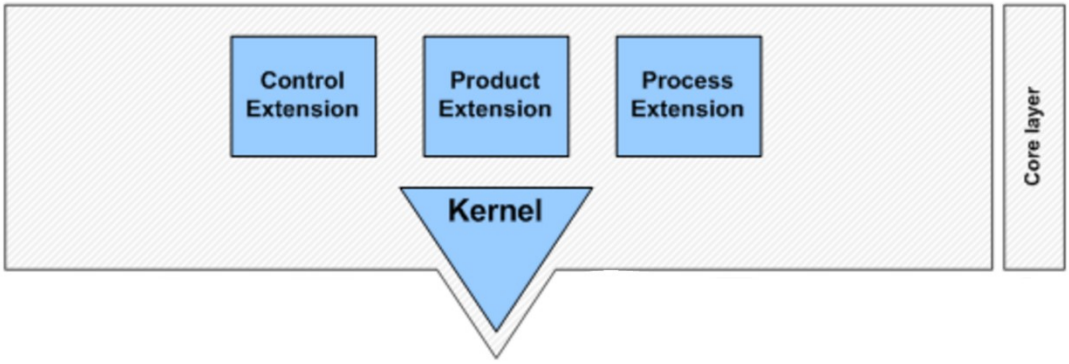


Figura 17 - Architettura IFC, core layer

Vengono definite entità astratte utilizzate e specificate in altri livelli. Questo livello può essere suddiviso in:

- *Kernel*: rappresenta il sottolivello che definisce le entità più astratte del *Core layer* (ad esempio *IfcRoot*);

- *Extension*: specializza le entità definite nel Kernel, in particolare gestisce le entità facenti parte del settore AEC/FM.

Nella seguente immagine viene rappresentato la definizione di un generico *IfcObject* (entità appartenente al *Kernel*) e la specializzazione nel substrato *Core Extension*.

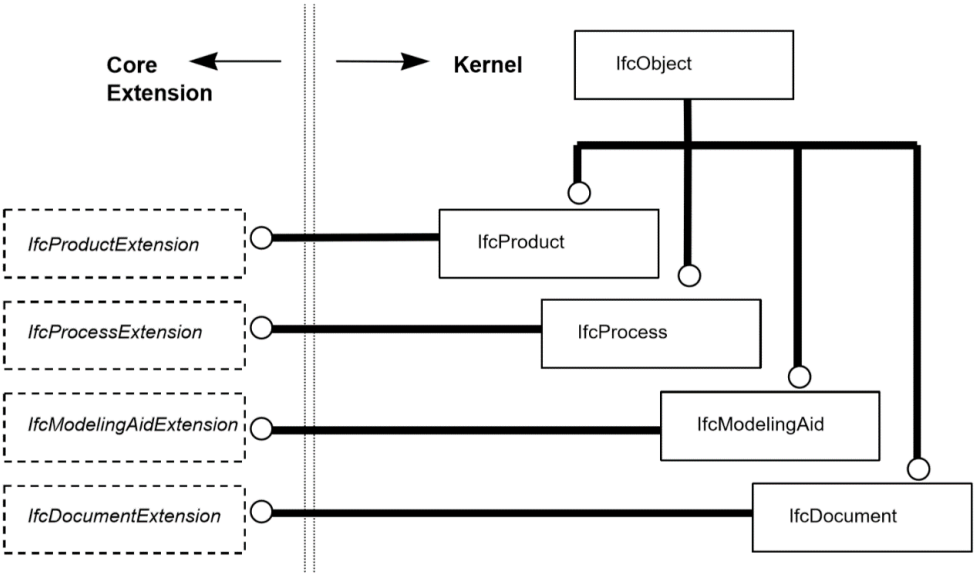


Figura 18 - Definizione dell'entità IfcObject tra Kernel e Core Extension

1.6.6.3 Interoperability layer

Figura 19 - Architettura IFC, interoperability layer

Rappresenta il livello dove vengono definite entità tipiche del settore AEC; gli elementi vengono comunemente usati e condivisi tra gli applicativi software (interoperabilità verticale). All'interno dello schema "*Shared Building Elements*" vengono definite entità come trave, pilastro, muro, porta e altri elementi edilizi. Gli elementi specificati in questo livello possono essere referenziati e specializzati dalle entità del domain layer.

1.6.6.4 Domain Layer

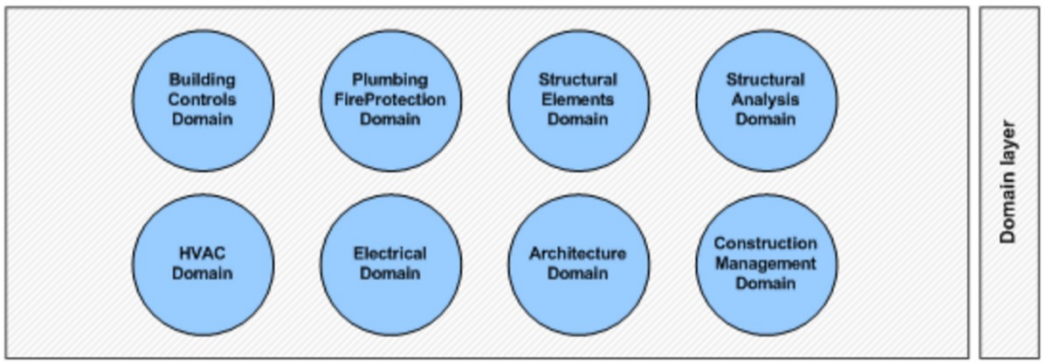


Figura 20 - Architettura IFC, domain layer

Rappresenta il livello più alto e più specializzato dello schema IFC. Ciascun elemento contiene l'elenco di entità e le enumerazioni specifiche del proprio dominio.

1.6.7 Schema IFC2x3, IFC4

Il formato IFC per adeguarsi alle nuove e sempre più specializzate esigenze del mondo delle costruzioni è in costante evoluzione. Tra le diverse versioni rilasciate, solo due sono al momento utilizzate:

- IFC2x3 Technical Corrigendum 1: è lo schema IFC più diffuso e utilizzato dai diversi *BIM tools;* nonostante alcune note limitazioni, lo schema permette un'esportazione stabile e coerente con il modello virtuale rappresentato;
- IFC4 Addendum 2: è stato pensato per superare le limitazioni e le critiche delle versioni precedenti. Sono state implementate alcune nuove proprietà per la rappresentazione geometrica e per la gestione delle proprietà parametriche.

1.6.8 Model View Definition

Con l'obiettivo di migliorare l'interoperabilità tra i diversi applicativi software sono state sviluppate definizioni di viste (MVD, *Model View Definition*) costituite da sottoinsiemi degli schemi IFC. A seconda dell'ambito AEC, vengono definiti degli standard della struttura del file in modo da ottenere uno strumento condivisibile e il più versatile possibile. La corrispondenza delle strutture di esportazione dei vari software al modello di vista viene certificata dalla stessa *buildingSMART* tramite una procedura che può verificare la compatibilità nell'ambito architettonico, strutturale o impiantistico.

Le definizioni di vista attualmente utilizzate sono:

- IFC2x3 Coordination View V2.0: è il modello di vista globalmente riconosciuto per lo schema IFC2x3, definisce i requisiti di scambio in ambito architettonico, strutturale e impiantistico;
- IFC4 Reference View 1.0: l'obiettivo principale è quello di definire un modello di vista standardizzato del formato IFC4 adatto a gestire il flusso di lavoro BIM dove l'interoperabilità è principalmente monodirezionale;
- IFC4 Design Transfer View 1.0: ha come obiettivo quello di ottenere un modello di vista che sia in grado di gestire le geometrie e le connessioni tra gli elementi. Alcuni applicativi permettono di inserire, cancellare, spostare o modificare elementi edilizi o spaziali. L'utilizzo di questo modello di vista può rivelarsi utile nel caso in cui diversi progettisti debbano intervenire sulla geometria degli stessi elementi.

2. Metodologia

Viene in seguito analizzato il flusso metodologico delle operazioni che ha permesso la definizione delle funzioni di calcolo per l'ottimizzazione della gestione dei dati e la successiva applicazione al file IFC del caso studio.

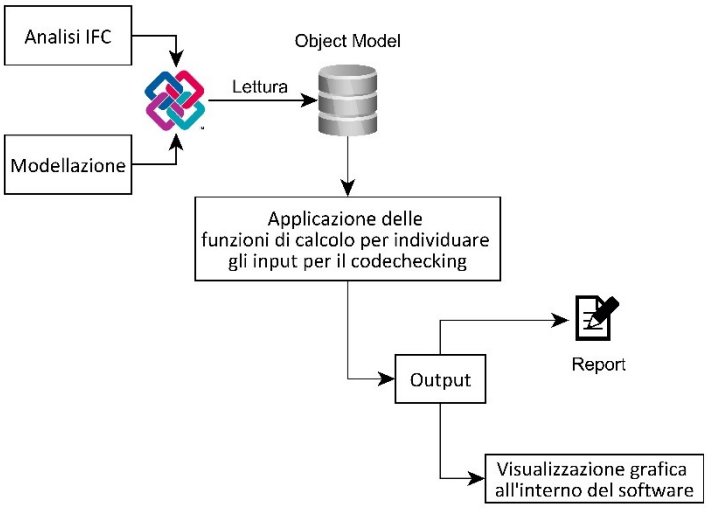


Figura 21 - Flusso metodologico

2.1 Analisi e lettura di un file IFC

Come già definito in precedenza, l'IFC si presenta come un file ASCII avente estensione ".ifc". Questo formato permette a qualsiasi lettore di testo di poter aprire il file e leggerne il contenuto in linguaggio EXPRESS.

Il file una volta aperto si presenta come un documento di testo suddiviso per righe; in testa ad ogni riga è presente il carattere "#" seguito da un numero: questo sistema permette di definire un *STEP ID* (tag) che identifica univocamente ogni riga presente nel documento.

Come riportato nel seguente esempio, ogni file ha un'intestazione (*header*) nella quale vengono specificati i dati di esportazione. Per rendere più comprensibile la lettura sono stati inseriti dei commenti utilizzando la sintassi definita dal linguaggio EXPRESS: /*testo del commento*/.

```
HEADER;
FILE_DESCRIPTION(('ViewDefinition [CoordinationView_V2.0]'),'2;1');
FILE_NAME('TRP_modello STR_torre.rvt', /*Nome del file esportato*/
'2017-09-18T12:11:12', /*Data e ora di esportazione*/
('Gioele Lauro'), /*Nome utente*/
('Politecnico di Torino'), /*Company*/
'The EXPRESS Data Manager Version 5.02.0100.07 : 28 Aug 2013',
'20170816_0615(x64) - Exporter 17.4.0.0 - Alternate UI 17.4.0.0',
''); /*informazioni sul software system*/
FILE_SCHEMA(('IFC2X3'));/*File schema utilizzato*/
ENDSEC;
```

A seguito dell'intestazione troviamo la sezione dati (*data*) che contiene tutte le istanze delle entità IFC. Ogni riga rappresenta un'entità diversa e viene definita dallo STEP Id che ha valenza solo per uno specifico file; infatti la stessa entità, esportata una seconda volta dallo stesso modello, potrebbe cambiare numero di riga e posizione. Il file STEP è stato pensato per non avere informazioni ripetute o ridondanti: i singoli dati vengono quindi dichiarati una volta sola e potranno essere richiamati da altre entità attraverso l'uso dello STEP Id.

```
DATA;
#1= IFCORGANIZATION($,'Autodesk Revit 2017 (ITA)',$,$,$);
#5=      IFCAPPLICATION(#1,'2017','Autodesk      Revit      2017
(ITA)','Revit');
#6= IFCCARTESIANPOINT((0.,0.,0.));
#11= IFCDIRECTION((1.,0.,0.));
#31= IFCAXIS2PLACEMENT3D(#6,#11,$);
ENDSEC;
```

Le informazioni delle entità sono contenute tra le parentesi e vengono divise dal carattere ",". Dall'esempio precedente, è possibile notare come lo *STEP ID* "#31" richiami al suo interno le entità "#6" e "#11" lasciando invece il terzo campo vuoto (utilizzando il carattere "$"). Il significato dei singoli campi all'interno delle parentesi cambia a seconda delle entità e a seconda della posizione. Utilizzando il "*data model*" fornito dalla *buildingSMART* è possibile ricavare le informazioni necessarie per leggere e interpretare i vari attributi; in particolare è possibile leggere il significato dei vari campi dal *inheritance graph* di ogni entità. A completamento dell'esempio precedente viene quindi riportato il grafico riferito all'entità *IFCAXIS2PLACEMENT3D*.

Inheritance Graph:

```
ENTITY IfcAxis2Placement3D
 ENTITY IfcRepresentationItem
 INVERSE
  LayerAssignment            : SET [0:1] OF IfcPresentationLayerAssignment FOR AssignedItems;
  StyledByItem               : SET [0:1] OF IfcStyledItem FOR Item;
 ENTITY IfcGeometricRepresentationItem
 ENTITY IfcPlacement
  Location                   : IfcCartesianPoint;
 DERIVE
  Dim                        : IfcDimensionCount := Location.Dim;
 ENTITY IfcAxis2Placement3D
  Axis                       : OPTIONAL IfcDirection;
  RefDirection               : OPTIONAL IfcDirection;
 DERIVE
  P                          : LIST [3:3] OF IfcDirection := IfcBuildAxes(Axis, RefDirection);
END_ENTITY;
```

Figura 22 - Inheritance Graph dell'entità IfcAxis2Placement3D

Lo schema riporta tutti i possibili attributi radicati secondo una ben definita gerarchia. In particolare, è possibile risalire al valore dei campi nel file IFC guardando in ordine dall'alto verso il basso gli attributi riferiti alle entità (*entity*). Nel precedente esempio sono quindi riportati i campi: Location, *Axis* e *RefDirection*. A destra della voce è possibile vedere il tipo di entità che può essere contenuta nel campo (es. *IfcCartesianPoint* e *IfcDirection*).

2.1.1 Rappresentazione degli spazi

Ogni file IFC creato per il settore AEC, raggruppa le definizioni dei vari componenti edilizi in entità spaziali seguendo un concetto gerarchico ben definito.

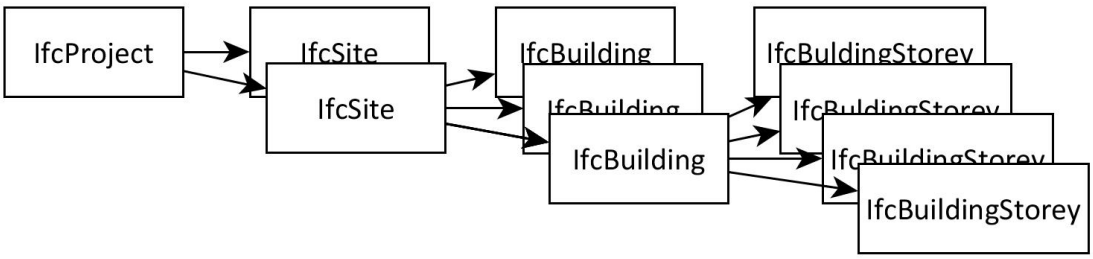


Figura 23 - Gerarchia delle entità spaziali

Ad ogni entità spaziale vengono assegnate determinate caratteristiche in modo da completare le informazioni che la sola descrizione dei componenti edilizi non potrebbe dare. In ordine gerarchico avremo le entità:

- **IfcProject**: definisce tutte le informazioni condivise con l'intero progetto come il sistema di unità di misura, il sistema di riferimento per le coordinate, la precisione, l'indicazione della differenza tra Nord di progetto e Nord reale;
- **IfcSite**: include le informazioni sulle coordinate geografiche in funzione del sistema di riferimento definito nel *project* (longitudine, latitudine e la quota altimetrica) e indica l'indirizzo e il paese di riferimento;
- **IfcBuilding**: rappresenta la struttura fondamentale del progetto e fornisce informazioni aggiuntive come la quota di costruzione rispetto a quanto definito nel *site*;
- **IfcBuildingStorey:** indica l'elemento spaziale base nel quale vengono raggruppati tutti i componenti edilizi. Ogni *storey* viene identificato con un nome e con la quota (*Elevation*) rispetto a quella di riferimento indicata nel *building*.

2.1.2 Rappresentazione dei componenti edilizi

I componenti edilizi che permettono di descrivere il mondo delle costruzioni vengono classificati e racchiusi dall'entità astratta *IfcBuildingElement*. Come ogni elemento BIM, anche queste entità coniugano una rappresentazione geometrica a una descrizione delle proprietà.

2.1.2.1 Rappresentazione geometrica

Ogni oggetto reale che può essere descritto tramite un'entità IFC, deriva dall'entità *IfcProduct* secondo la seguente scomposizione gerarchica.

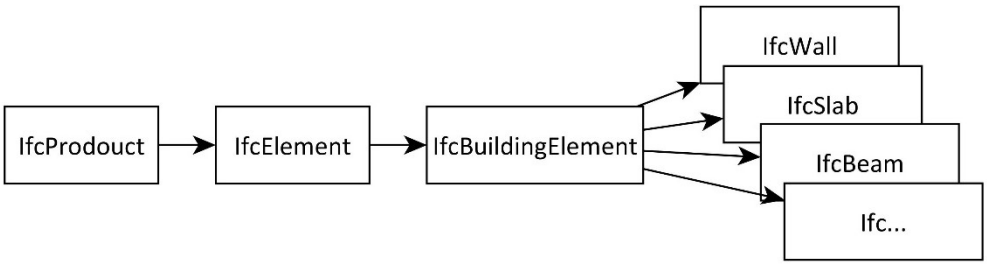


Figura 24 - Scomposizione degli elementi dall'entità IfcProduct

La descrizione geometrica degli elementi appartenenti all'entità *IfcProduct* è resa possibile dall'implementazione di apposite entità che permettono la descrizione di elementi bidimensionali o tridimensionali nello spazio. L'entità astratta che racchiude tutte le possibili rappresentazioni degli elementi prende il nome di *IfcProductRepresentation*.

```
ENTITY IfcProduct
 ObjectPlacement     :OPTIONAL IfcObjectPlacement;
 Representation      :OPTIONAL IfcProductRepresentation;
```

Figura 25 - Specifiche EXPRESS dell'entità IfcProduct

Tutte le geometrie sono definite in riferimento ad una posizione descritta secondo l'opportuno sistema di riferimento (*ObjectPlacement*). Vi sono due entità fondamentali e distinte che permettono la rappresentazione geometrica:

- **IfcProductDefinitionShape**: è l'entità di rappresentazione riferita in modo diretto a *IfcProduct*, permette di attribuire un nome e una descrizione e ha diretto riferimento con *l'ObjcetPlacement* sopra descritto;
- **IfcShapeRepresentation**: è l'entità che permette di gestire le rappresentazioni più generica e comprende al suo interno tutte le entità utili alla descrizione geometrica di elementi bidimensionali e tridimensionali; a differenza dell'entità *IfcProductDefinitionShape* la posizione è definita in base a quanto definito in *IfcRepresentationContext*.

Essendo di carattere molto più generale e potendo essere richiamata dall'entità *IfcProductDefinitionShape*, si è deciso di analizzare i principali metodi di rappresentazione solida definiti dall'entità *IfcShapeRepresentation* (coerentemente con quanto descritto nella ISO 10303-41.)

Inheritance Graph:

```
ENTITY IfcShapeRepresentation
 ENTITY IfcRepresentation
  ContextOfItems          :IfcRepresentationContext;
  RepresentationIdentifier:OPTIONAL IfcLabel;
  RepresentationType      :OPTIONAL IfcLabel;
  Items                   :SET [1:?] OF IfcRepresentationItem;
 INVERSE
  RepresentationMap      :SET [0:1] OF IfcRepresentationMap FOR MappedRepresentation;
  LayerAssignments       :SET OF IfcPresentationLayerAssignment FOR AssignedItems;
  OfProductRepresentation:SET OF IfcProductRepresentation FOR Representations;
 ENTITY IfcShapeModel
 INVERSE
  OfShapeAspect          :SET [0:1] OF IfcShapeAspect FOR ShapeRepresentations;
 ENTITY IfcShapeRepresentation
END_ENTITY;
```

Figura 26 - Inheritance Graph dell'entità IfcShapeRepresentation

Osservando il grafico delle gerarchie bisogna porre particolare attenzione ai campi *RepresentationIdentifier* e *RepresentationType*; il primo indica il tipo di elemento da rappresentare, mentre il secondo specifica il metodo di rappresentazione utilizzato per la descrizione geometrica. L'uso di queste entità permettono di gestire tutte le rappresentazioni solide e piane definite dal formato IFC.

I principali valori che il campo *RepresentationIdentifier* può assumere sono:

- **box**: definisce un elemento di delimitazione nello spazio;
- **annotation**: indica un'annotazione bidimensionale che non rappresenta un elemento;
- **axis**: indica una linea o un versore nel piano o nello spazio;
- **profile**: definizione di un profilo piano nello spazio;
- **surface**: definisce una superficie tridimensionale (anche superfici analitiche);
- **body**: indica un solido nello spazio.

Mentre i principali valori del campo *RepresentationType* può assumere sono:

- **point**: rappresenta un punto nel piano o nello spazio;
- **curve**: indica una curva bidimensionale o tridimensionale;
- **surface**: utilizzato per rappresentare una curva nello spazio;

- **text**: definisce un testo;
- **SolidModel**: raggruppa i metodi di rappresentazione solida.

2.1.2.2 SolidModel

Per l'analisi dei successivi algoritmi di calcolo risulta indispensabile approfondire i metodi utilizzati in un file IFC per la descrizione geometrica di un solido. I metodi più diffusi di *RepresentationType* appartenenti alla famiglia di rappresentazione *SolidModel* sono:

- **SweptSolid**: viene generato un solido per estrusione o rivoluzione di un elemento bidimensionale;
- **AdvanceSweptSolid**: viene generato un solido per estrusione o rivoluzione lungo una generica direttrice;
- **Brep**: solido definito dalla composizione di superfici definite da linee che vengono a loro volta determinate da una lista di punti. In questo caso non avrò direttamente le informazioni riguardo le dimensioni della sezione o della lunghezza dell'elemento;
- **AdvancedBrep**: solido rappresentato in maniera analoga al *Brep* ma avente superfici definite da generiche curve nello spazio;
- **CGS**: rappresentazione del risultato dell'operazione booleana (somma o sottrazione) tra solidi o tra un solido e un semispazio;
- **Clipping**: differenza booleana tra solidi estrusi o tra un solido e un semispazio;
- **BuondingBox**: rappresentazione tridimensionale attraverso la definizione di un elemento di delimitazione nello spazio;
- **SectionedSpine**: permette la rappresentazione per estrusione di un elemento avente sezione variabile;
- **MappedRepresentation**: definisce una rappresentazione di oggetti che seguono una serie di punti; viene spesso utilizzata insieme all'estrusione lungo una direttrice.

2.1.2.3 Proprietà degli elementi

Come descritto nella ISO 10303 parte 21 e 22, le caratteristiche degli elementi vengono definite secondo una specifica gerarchia delle proprietà. La rigida struttura che raccoglie tutte le informazioni permette ai diversi software che gestiscono l'importazione IFC, di leggere in modo univoco i dati cercati; d'altro canto bisogna constatare che il fatto di avere una struttura rigida consente la condivisione delle sole caratteristiche implementate nella struttura stessa. Per questo motivo risulta fondamentale il continuo lavoro di sviluppo e aggiornamento della struttura dati per rendere il formato competitivo in ogni sua applicazione.

L'entità astratta *IfcPropertyDefinition* permette di definire le caratteristiche di un generico elemento. A titolo informativo viene riportata una delle entità più ricorrenti: *IfcMaterial*.

```
ENTITY IfcMaterial
 Name                  : IfcLabel;
 Description           : OPTIONAL IfcText;
 Category              : OPTIONAL IfcLabel;
```

Figura 27 - Attributi dell'entità IfcMaterial

Come vedremo in un semplice esempio, questa entità permette l'attribuzione del nome, della descrizione e della classificazione di un generico materiale.

2.1.3 Definizione dei sistemi di riferimento

Nel formato IFC la gestione dei sistemi di riferimento può avvenire attraverso l'utilizzo di:

- **coordinate assolute** se fanno riferimento al sistema di riferimento globale;
- **coordinate relative** se definite in funzione del posizionamento di un altro elemento (come descritto nel paragrafo successivo);
- **coordinate vincolate** se fanno riferimento agli assi di una griglia definita.

Richiamando la gerarchia della rappresentazione degli spazi precedentemente trattata (paragrafo 2.1.1), l'unica entità ad essere sempre descritta attraverso l'utilizzo di coordinate assolute è *l'IfcSite*. Il posizionamento nello spazio delle entità gerarchicamente dipendenti potrà avvenire attraverso l'utilizzo di una serie di sistemi di coordinate relative collegati al sistema di riferimento del sito. Applicando questi concetti ad un file IFC, la posizione dell'entità *IfcBuilding* può essere descritta attraverso un sistema di riferimento a coordinate relative rispetto all'entità *IfcSite*; di conseguenza anche la posizione dell'entità *IfcBuildingStorey* potrà essere definita in relazione alla posizione *dell'IfcBuilding*.

2.1.3.1 Sistema di riferimento a coordinate relative: IfcLocalPlacement

L'entità IFC utilizzata per descrivere un sistema di riferimento locale è l'*IfcLocalPlacement*.

Inheritance Graph:

```
ENTITY IfcLocalPlacement
 ENTITY IfcObjectPlacement
 INVERSE
  PlacesObject          :SET [0:?] OF IfcProduct FOR ObjectPlacement;
  ReferencedByPlacements:SET OF IfcLocalPlacement FOR PlacementRelTo;
 ENTITY IfcLocalPlacement
  PlacementRelTo    :OPTIONAL IfcObjectPlacement;
  RelativePlacement:IfcAxis2Placement;
END_ENTITY;
```

Figura 28 - Inheritance Graph dell'entità IfcLocalPlacement

Analizzando il grafico delle gerarchie va posta particolare attenzione ai campi *PlacementRelTo* e *RelativePlacement*. Il primo indica l'oggetto (o il suo sistema di riferimento) rispetto al quale voglio definire le nuove coordinate mentre il secondo indica le possibili rotazioni degli assi o traslazione dell'origine. Le caratteristiche del nuovo sistema di riferimento sono definite dalle voci:

- **IfcAxis2Placement2D**: utilizzato per definite un sistema di coordinate su un piano, viene specificata l'origine del nuovo sistema di riferimento e la direzione del versore dell'asse X (se non specificato prende il valore di default [1,0]);
- **IfcAxis2Placement3D**: utilizzato per definire un sistema di coordinate nello spazio, viene specificata l'origine del nuovo sistema di riferimento, la direzione del versore dell'asse Z (se non specificato [0,0,1]) e il vettore che permette di definire il versore dell'asse X proiettando quest'ultimo vettore sul piano perpendicolare all'asse Z e passante per l'origine del nuovo sistema di riferimento (se non specificato [1,0,0]).

Per facilitare la comprensione dei concetti precedenti, viene proposto un esempio sia della struttura all'interno del formato IFC, sia della riproduzione grafica nello spazio di un generico sistema di riferimento.

```
/*Definizione del sistema di riferimento globale*/

#1=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,0.0001,#2,$)

#2=IFCAXIS2PLACEMENT3D(#3,$,$)

/*Il sistema globale avrà come origine #3, versore Z[0,0,1] e versore
X[1,0,0]*/

#3=IFCCARTESIANPOINT(0,0,0)

/*definisco un sistema di riferimento locale*/

#4=IFCLOCALPLACEMENT($,#5)

/*il carattere $ nel campo "PlacementRelTo" indica che il nuovo sistema di

riferimento sarà definito rispetto al sistema di riferimento globale*/

#5=IFCAXIS2PLACEMENT3D(#10,#11,#12)

#10=IFCCARTESIANPOINT(0,10,6)/*l'origine del nuovo sistema di riferimento*/

#11=IFCDIRECTION(0,0,-1) /*direzione del versore del nuovo asse Z*/

#12=IFCDIRECTION(-1,0,0)

/*direzione di riferimento per definire il versore del nuovo asse X*/
```

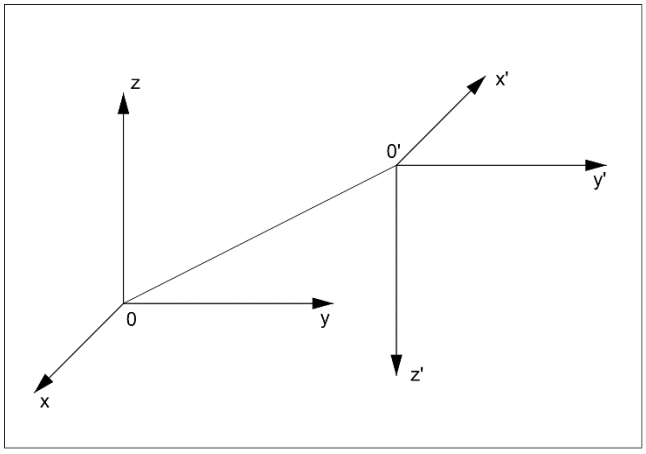


Figura 29 - Rappresentazione grafica dell'entità IfcLocalPlacement

Confrontando la riproduzione grafica con quanto letto all'interno del file IFC è possibile osservare come la distanza 00' sia definita dalla riga #10, mentre la rotazione degli assi sia rappresentata nelle righe #11 e #12.

2.1.4 Esempio IfcBeam

In questo paragrafo viene descritta la rappresentazione di una semplice trave partendo dalla sua modellazione nell'ambiente di lavoro di *Autodesk Revit 2017* fino all'esportazione in formato IFC.

2.1.4.1 *Modellazione della trave come famiglia parametrica*

Il seguente esempio riporta la creazione di una semplice famiglia di una trave rettangolare in calcestruzzo. La modellazione avviene parametrizzando le dimensioni della base e dell'altezza in modo da poter gestire differenti tipi di trave rettangolare. Nell'editor delle famiglie in *Autodesk Revit 2017* viene creata la geometria del solido come semplice estrusione. Le dimensioni del rettangolo da estrudere vengono definite su una vista di prospetto rendendo i lati solidali ad un sistema di griglie. Infine, è necessario parametrizzate le dimensioni della base e dell'altezza come quota tra le griglie.

Terminata la modellazione nell'editor delle famiglie si inserisce l'elemento creato nel progetto: viene quindi selezionata la famiglia e si inserisce la trave utilizzando i parametri "tipo" precedentemente creati (h=600 mm, b=300 mm). Nel progetto viene istanziata la famiglia in base al livello (quota) di inserimento e alla lunghezza con cui viene modellata. Per ottenere l'esportazione sarà sufficiente selezionare il comando "Esporta IFC". In questa prima fase il file IFC è stato volutamente creato senza modificare i parametri di esportazione.

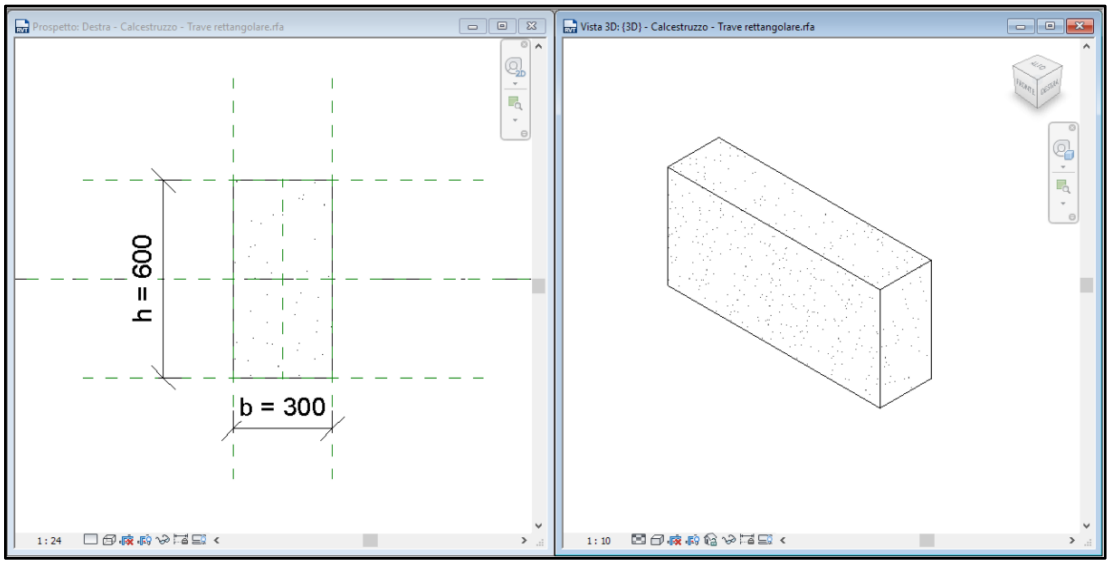


Figura 30 - Modellazione parametrica di una trave

2.1.4.2 *Visualizzazione e interpretazione del file IFC esportato*

La visualizzazione grafica dei file IFC è resa possibile da numerosi freeware disponibili in rete. Data la grande diffusione e la sicurezza di ottenere un risultato attendibile si è scelto di utilizzare il visualizzatore "*Solibri Model Viewer*". L'interfaccia grafica permette di gestire al meglio la visualizzazione con diversi comandi per spegnere elementi o per creare nuove viste in sezione.

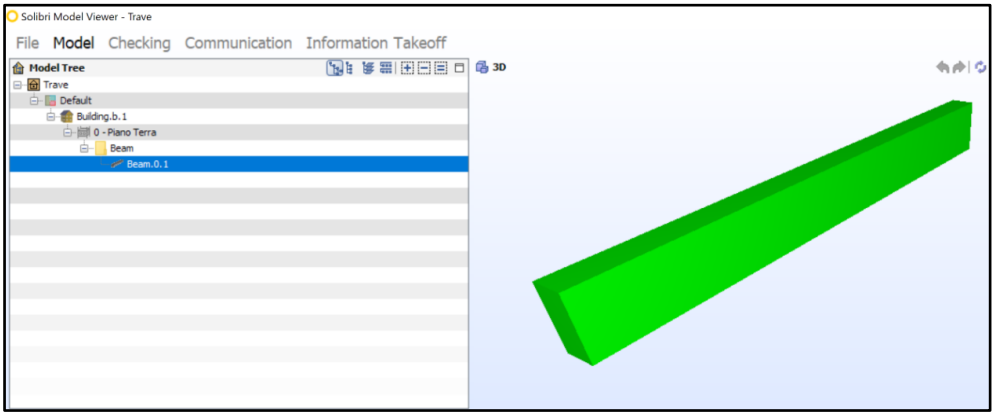


Figura 32 - Apertura del file IFC attraverso Solibri Model Viewer

In parallelo alla visualizzazione grafica, un file IFC può essere analizzato leggendone direttamente il contenuto secondo la codifica STEP: è infatti possibile aprire il file con qualsiasi editor in grado di gestire il formato ASCII (es. Blocco Note di Microsoft). Per rendere più fruibile l'analisi è stato utilizzato il software gratuito *IfcQuickBroswer*: questo applicativo permette di riordinare le righe che compongono il file in modo da rendere il contenuto più leggibile.

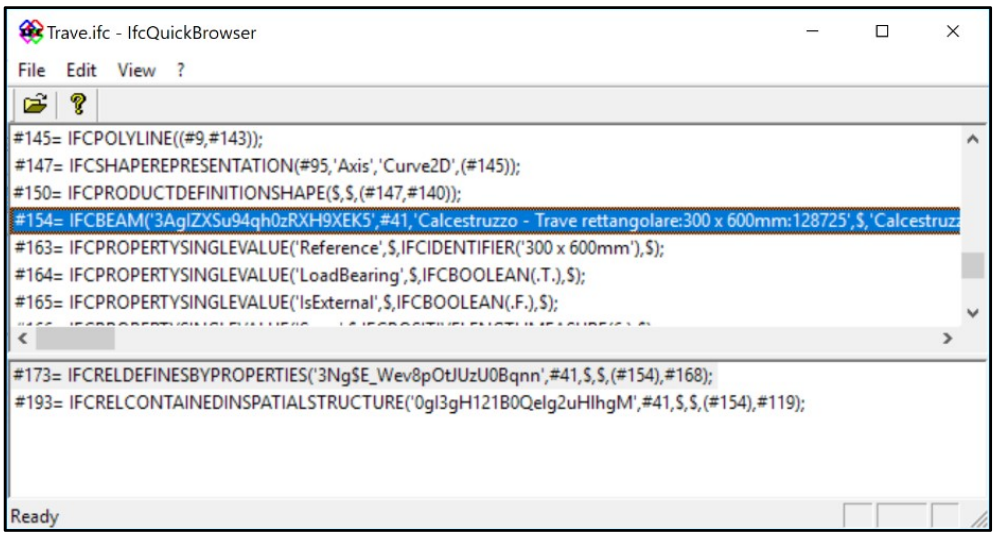


Figura 31 - Apertura del file IFC attraverso IfcQuickBrowser

L'immagine riporta la tipica interfaccia del programma. Nella parte centrale è possibile visualizzare il codice presente nel file mentre nella parte bassa sono presenti le righe che richiamano l'entità selezionata. La peculiarità del programma è che con un doppio click su un'entità si aprirà un albero riportante le righe a cui l'entità si riferisce; osservando il successivo esempio è possibile vedere come al doppio click sulla riga #154 vengano visualizzate le indentature delle entità richiamate #41, #130 e #150.

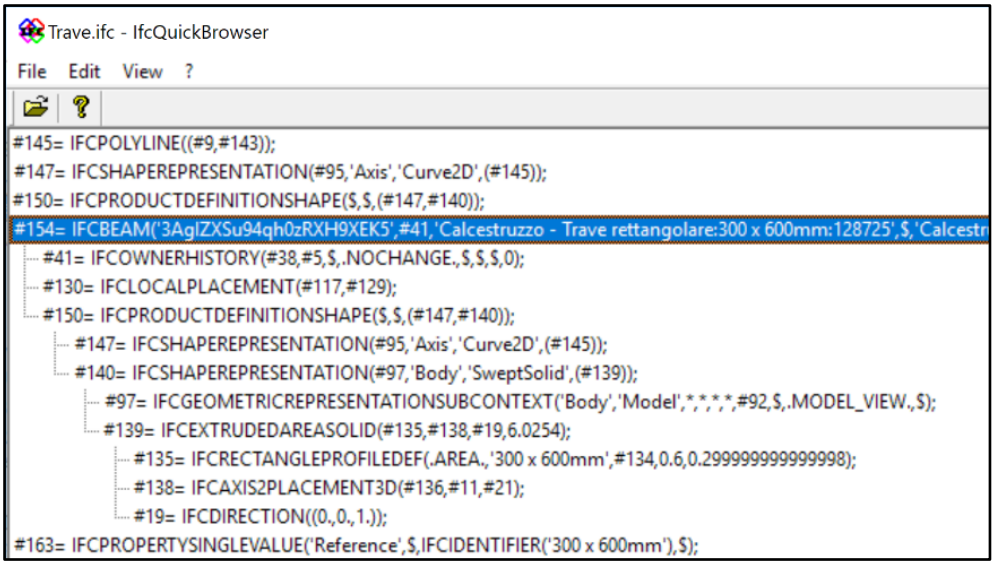


Figura 33 - Albero delle entità in IfcQuickBroswer

L'elemento modellato precedentemente in *Revit* è stato esportato come entità *IfcBeam*; tra le parentesi tonde sono presenti le proprietà degli attributi così come riportate nel grafico della gerarchia dell'entità in esame. Tra i diversi attributi è possibile notare come la posizione dell'elemento nello spazio sia definita da un sistema di riferimento a coordinate relative (riga #130) mentre la rappresentazione venga definita dall'entità *IfcProductdefinitionShape* (riga #150). Analizzando più nel dettaglio la rappresentazione geometrica, è possibile notare come venga rappresentato sia l'asse della trave (riga #147), sia il solido di estrusione (riga #140). La rappresentazione solida avviene tramite estrusione (riga #139) di un profilo rettangolare (riga #135). Per risalire al significato di ogni attributo è sufficiente analizzare, per ogni entità, le proprietà definite da *buildingSMART* all'interno del "data model" di riferimento.

2.2 IFC applicato al caso studio

2.2.1 Modellazione, obiettivi e materiale fornito

Il lavoro di modellazione ha avuto due principali obiettivi:

- ottenere un file IFC con i dati necessari per i successivi calcoli;
- definire gli standard e le criticità della modellazione di un file IFC ai fini del *code checking* strutturale.

In una prima analisi, la modellazione è iniziata da un file *Revit* unico contenente gli elementi di tutte le discipline. In questa fase la gestione del file risultava lenta e poco efficiente data la notevole pesantezza del modello. Per risolvere questo problema, il file è stato diviso in diversi modelli a seconda della disciplina trattata.

A completamento del modello *Revit* è stata fornita la documentazione relativa alle tavole esecutive per la realizzazione degli elementi strutturali. L'organizzazione delle tavole è gestita attraverso un file Excel che riporta il significato degli acronimi utilizzati e i titoli delle tavole. Il materiale è stato fornito sia in formato in formato pdf che in un formato editabile (.dwg).

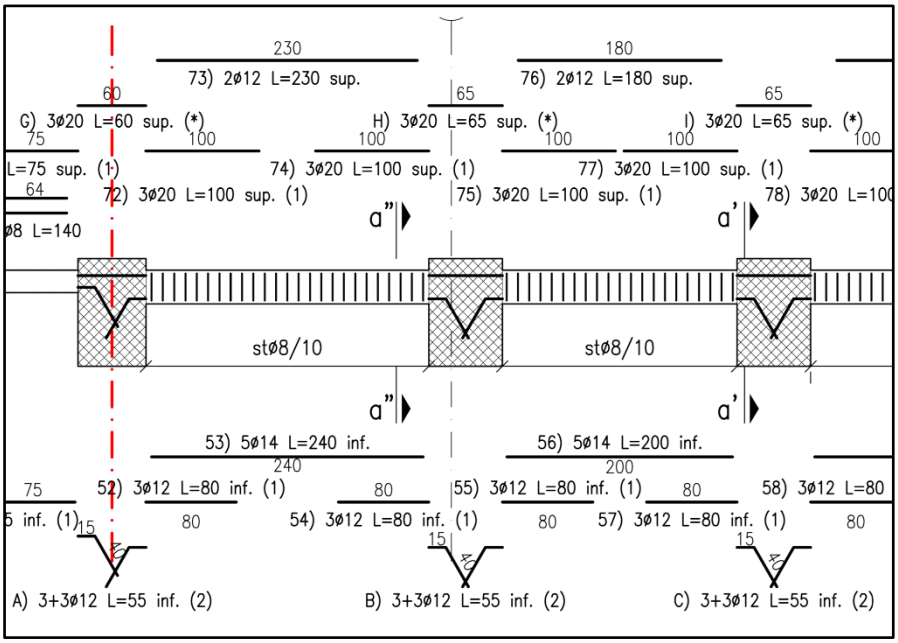


Figura 34 - Estratto di una tavola esecutiva degli elementi strutturali

L'attività di modellazione ha riguardato prevalentemente la definizione degli elementi strutturali appartenenti al satellite del settimo piano. La scelta della porzione di edificio è stata fatta in base alla maggior corrispondenza tra gli elementi modellati e i successivi calcoli da effettuare. Dopo un'accurata analisi della documentazione, sono state modellate le travi principali e secondarie, il solaio portante e le relative armature. In ausilio alla modellazione è stato sfruttato un plug-in messo a disposizione direttamente sullo store di Autodesk: *Revit Extensions*. Questo applicativo permette, a seconda dell'elemento selezionato, di inserire in modo automatico l'armatura definita dall'utente.

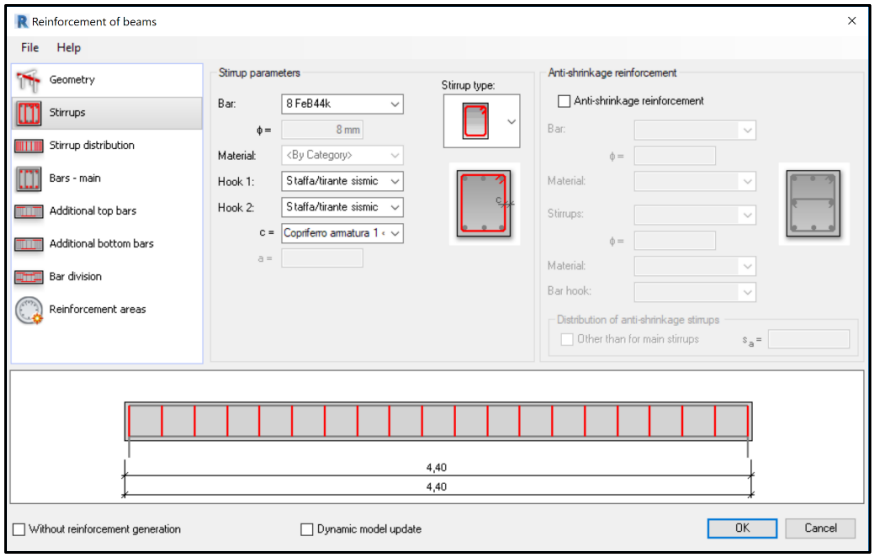


Figura 35 - Revit Extensions per la modellazione dell'armatura

Nelle finestre che l'applicativo mette a disposizione si possono definire i parametri della staffatura, le barre longitudinali superiori/inferiori ed eventuali elementi di rinforzo. Terminata questa prima modellazione si è reso necessario passare alla modellazione manuale degli elementi d'armatura non ordinari. L'applicativo infatti, è stato sviluppato per il disegno dell'armatura di elementi comuni senza tenere conto di eventuali casi particolari.

2.2.2 Esportazione del modello

L'esportazione in IFC avviene sfruttando il plug-in *IFC 2017* disponibile sullo store di Autodesk. Grazie a questo applicativo è possibile personalizzare l'esportazione e ottenere un file IFC con le caratteristiche desiderate.

Viene di seguito analizzato il processo metodologico che ha portato, passo dopo passo, al miglioramento della modellazione per l'esportazione in IFC. Come è possibile vedere dal diagramma di flusso, per ogni serie di elementi modellati (principalmente travi, pilastri o barre d'armature), il flusso di lavoro permette di utilizzare il modello IFC come riscontro grafico per valutare la correttezza del metodo usato per l'inserimento dei dati.

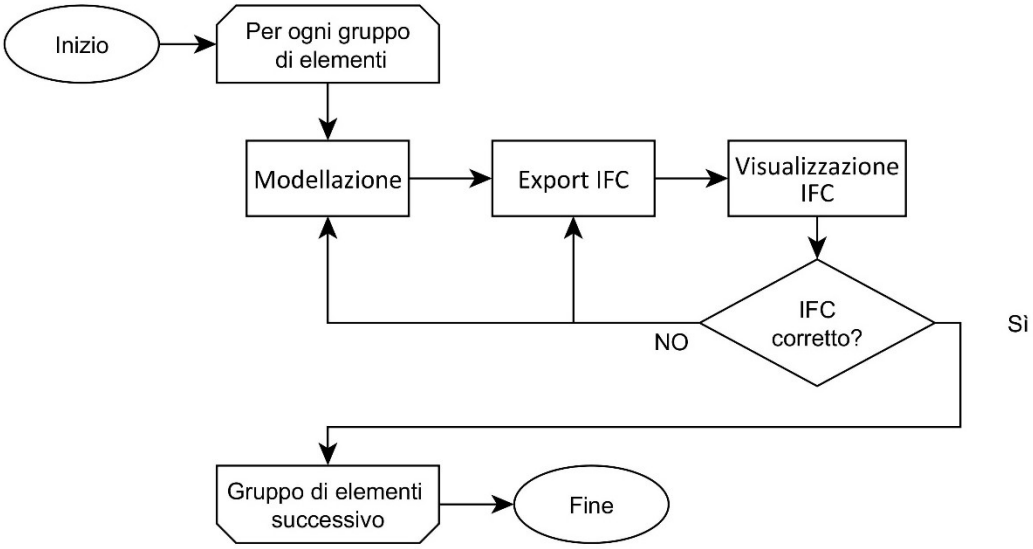


Figura 36 - Flusso delle operazioni di modellazione

Come analizzeremo in seguito, la corretta modellazione non dipende soltanto dal metodo di lavoro, ma, più in generale, dallo scopo e da come verrà utilizzato successivamente il modello. Per questo motivo è stata posta particolare attenzione all'analisi del risultato dell'esportazione IFC: attraverso l'utilizzo di visualizzatori grafici (*Solibri Model Viewer*) e visualizzatori di codice (*IfcQuickBroswer*) è stato possibile verificare sia la correttezza della modellazione, sia la scelta delle impostazioni di esportazione.

2.2.3 Impostazioni d'esportazione IFC

Come visto in precedenza, l'utilizzo dell'applicativo *IFC 2017* per *Autodesk Revit* permette l'impostazione di diversi parametri che consentono di ottenere l'esportazione desiderata.

Nella finestra di modifica delle configurazioni di esportazione il programma mette a disposizione diverse impostazioni personalizzabili.

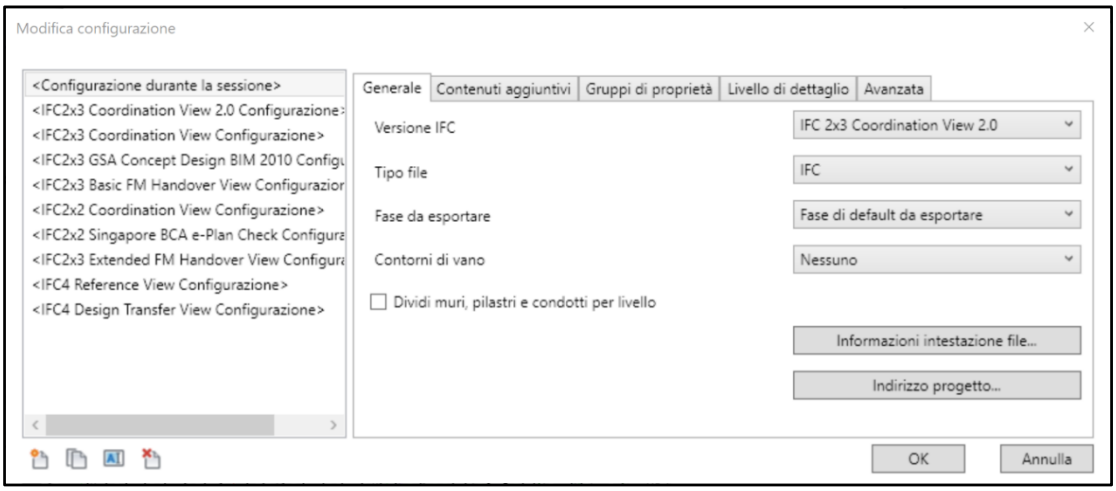


Figura 37 - Configurazioni d'esportazione IFC

Al fine di ottenere un file IFC per il *code checking* strutturale sono stati impostati i seguenti parametri:

- esportazione IFC nel formato *IFC 2x3 Coordination View 2.0* in accordo con il formato scelto per lo sviluppo del visualizzatore e dei seguenti algoritmi;
- divisione degli elementi strutturali per livello: è importante che gli elementi appartenenti a più piani vengano divisi e calcolati separandoli;
- esportazione dei soli elementi visibili nella vista: questa funzione permette di esportare solo la porzione di modello desiderata. Applicando dei filtri di visibilità risulta molto utile questa funzione per compiere delle verifiche puntuali su un gruppo limitato di oggetti anziché sull'intero modello.

In aggiunta a queste configurazioni, il programma permette all'utente di modificare le opzioni di esportazione per ogni tipo di famiglia in Revit. È infatti possibile modificare l'associazione di un tipo di famiglia di Revit in un'entità IFC scelta dell'utente. La configurazione definita di default è sicuramente quella che permette l'esportazione del maggior numero di parametri data l'associazione tra le famiglie Revit e le entità più pertinenti; in certe circostanze potrebbe però essere necessario modificare queste associazioni per ottenere delle diverse entità IFC. Analizzando il file txt che regola queste configurazioni, è possibile notare come molti elementi in Revit non vengano esportati in IFC; in altri casi l'esportazione avviene ma in entità IFC molto generiche: ne sono un esempio i numerosi elementi esportati nell'entità generica *IfcBuildingElementProxy*.

Categoria Revit	Nome classe IFC	Tipo IFC
Pavimento di massa	IfcSlab	FLOOR
Riempimento motivo	{ IfcBuildingElementProxy }	
Tetto massa	IfcRoof	
Vetrata massa	{ IfcBuildingElementProxy }	
Zona massa	{ IfcBuildingElementProxy }	
Modelli generici	IfcBuildingElementProxy	
IfcOpeningElements	IfcOpeningElement	
Linee nascoste	{ IfcBuildingElementProxy }	
Montanti della facciata continu	IfcCurtainWall	
Linee nascoste	{ IfcCurtainWall }	
Muri	IfcWall	
Bordi comuni	{ IfcWall }	
Estrusioni muro	IfcBuildingElementProxy	
Finitura 1 [4]	{ IfcWall }	
Finitura 2 [5]	{ IfcWall }	
Griglia di facciata continua	IfcWall	
Linee nascoste	{ IfcWall }	
Motivo sezione	{ IfcWall }	
Motivo superficie	IfcWall	
Muri sovrapposti	IfcWall	
Scanalature	IfcOpeningElement	
Sostrato [2]	{ IfcWall }	

Carica...
Standard
Salva con nome...

Figura 38 - Esempio di associazione tra categorie di Revit ed entità IFC

2.2.4 Modifiche nella modellazione per l'esportazione e il code checking

Il processo iterativo adottato ha messo in luce le problematiche relative alla modellazione e alla successiva esportazione. In particolare, vengono analizzati i principali accorgimenti che si è reso necessario adottare.

Ai fini del *code checking* strutturale è importante che la modellazione di una generica struttura venga svolta suddividendo ogni telaio nei singoli elementi che lo compongono. Nonostante una trave su più appoggi possa essere descritta come unico elemento, è buona pratica suddividere gli elementi per ogni campata. I vantaggi derivanti dal lavorare sui singoli elementi possono essere:

- Maggior libertà nella gestione del comportamento strutturale e dei vincoli alle estremità;
- facilità di inserimento dell'armatura sfruttando estensioni o plug-in per l'inserimento automatico;
- evitare, nell'esportazione, di avere volumi che si compenetrano.

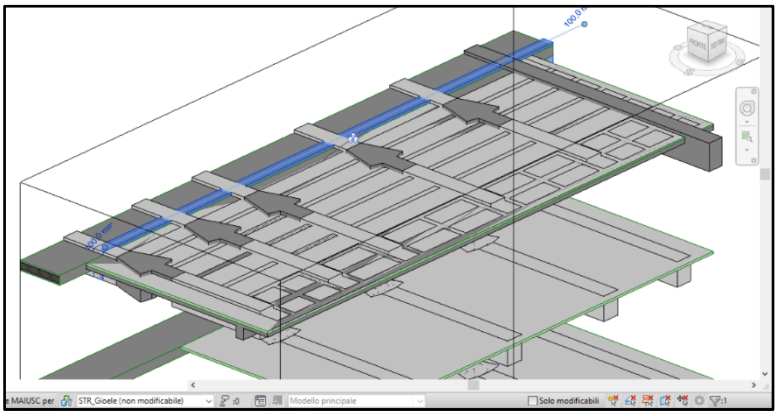


Figura 39 - Esempio di modellazione non ottimizzata per il code checking strutturale

Ciascun elemento che compone la struttura portante deve essere modellato come elemento della famiglia del telaio strutturale. Nella modellazione della famiglia delle travi di raccordo per trefoli, le diverse prove di esportazione hanno dimostrato come, se non modellato a partire da una famiglia del telaio strutturale, l'elemento non venga esportato.

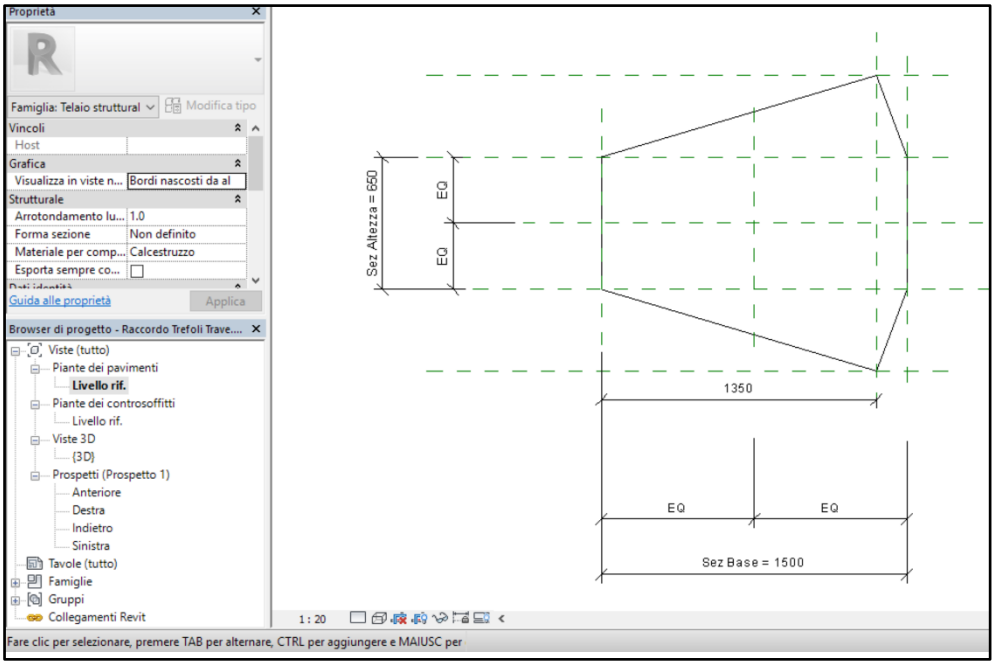


Figura 40 - Modellazione parametrica della trave di raccordo per trefoli come telaio strutturale

In ultima analisi, eseguendo la modellazione su un file strutturale separato, deve essere sempre verificata la congruenza con i file Revit che verranno collegati. Sovrapponendo tra loro i diversi file è stato possibile intercettare alcuni errori di modellazione. Come esempio viene riportato il caso di un errore di modellazione individuato grazie alla sovrapposizione del modello strutturale con quello architettonico.

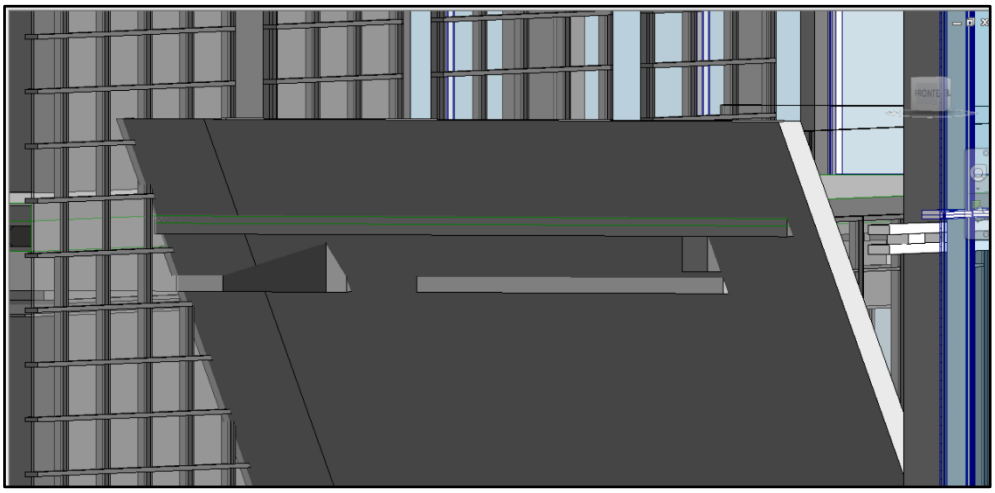


Figura 41 - Sovrapposizione dei modelli per analizzare le interferenze

2.3 Ottimizzazione della gestione dei dati per il code checking strutturale

Come accennato nell'introduzione, il fine dell'analisi sul formato IFC è quello di collaborare allo sviluppo di un software in grado di leggere un file IFC e di poter effettuare il *code checking* strutturale utilizzando le regole fornite dalle NTC. Una prima analisi ha permesso di individuare, tra le indicazioni fornite dalla norma, i controlli da effettuare sul modello. Per ogni regola sono stati quindi individuati i dati di input necessari per effettuare il controllo del soddisfacimento del requisito normativo. La ricerca dei dati all'interno del file IFC ha portato alla luce le prime difficoltà operative. I principali problemi emersi sono stati:

- la possibile assenza delle informazioni attese;
- la differente esportazione delle medesime informazioni da parte di diversi software;
- il non avere a disposizione direttamente le informazioni cercate ma dover effettuare dei calcoli per ottenerle.

Le possibili soluzioni si distinguono a seconda del metodo utilizzato per recuperare i dati. Nel caso in cui non si riuscissero ad ottenere le informazioni necessarie può essere inoltrata all'utente una richiesta d'inserimento di informazioni aggiuntive. In caso contrario, è possibile implementare una serie di funzioni in grado di eseguire, in modo automatico, una sequenza di operazioni per calcolare le informazioni mancanti.

Le successive trattazioni riguarderanno l'analisi, lo sviluppo e il test delle funzioni che permetteranno di sfruttare al meglio le informazioni contenute all'interno del file.

Con l'obiettivo di ricavare i dati per effettuare il *code checking* strutturale per il calcestruzzo armato, sono stati analizzati diversi file IFC esportati da più applicativi; i risultati hanno evidenziato le seguenti problematiche:

- non tutti i software esportano le relazioni che legano le armature con gli elementi che le contengono. Ne è un esempio il confronto tra l'esportazione del medesimo modello dal software *Tekla Structures* e da *Revit*. Nonostante la rappresentazione geometrica sia pressoché la medesima, l'esportazione effettuata con *Tekla* descrive in più una relazione del tipo *IfcRelAggregates* tra le armature e l'elemento in calcestruzzo che le contiene. Viene di seguito proposto l'esempio di una esportazione di una trave con un ferro d'armatura al suo interno esportato da *Tekla*. L'entità *IfcRelAggregates* della riga #149 lega, attraverso l'entità *IfcElementAssembly* (#33), la trave descritta alla riga #63 con l'elemento d'armatura #110;

```
#148= IFCRELAGGREGATES('2iQhx_t0TFkhcWnqNxXutJ',#5,$,$,#29,(#31));
#149= IFCRELAGGREGATES('0jYvXIYuP0OQk8xccVhwVr',#5,$,$,#33,(#110,#63));
  #5= IFCOWNERHISTORY(#3,#4,$,.NOCHANGE.,$,$,$,1493907425);
  #33= IFCELEMENTASSEMBLY('1P2mT20000L34qEJCvC3Os',#5,'BEAM',$,$,#32,$,'B\\0(?)',.NOTDEFINED.,.REINFORCEMENT_UNIT.);
  #110= IFCREINFORCINGBAR('1P2mT20000Pp4qEJCvC3Sn',#5,'REBAR',$,$,#93,#109,'ID590b0742-0000-0067-3134-393339303731',$,1
  #63= IFCBEAM('1P2mT20000LJ4qEJCvC3Os',#5,'BEAM','800*400','800*400',#46,#62,'Calcestruzzo_B-1\\0(?)');
#150= IFCRELCONTAINEDINSPATIALSTRUCTURE('2t8YrgbgbAgOTZ9MIDulqv',#5,$,$,(#33),#31);
#151= IFCRELDEFINESBYPROPERTIES('3XU0asxGPEKQRTCttreSSL',#5,$,$,(#33),#41);
```

Figura 42 - Esempio dell'entità IfcRelAggregates

- è difficile riscontrate in molte esportazioni la classificazione delle armature in base alla loro tipologia. Per le successive analisi normative risulta indispensabile avere la classificazione delle armature distinte prevalentemente tra armature longitudinali e trasversali. Come è possibile notare dal seguente esempio, il penultimo campo nel quale dovrebbe essere definita la tipologia dell'armatura risulta "NOTDEFINED".

```
#526= IFCRELAGGREGATES('0m1KV3JZj9lxFqaIoIiZ9E',#5,$,$,#33,(#506,#271,#246,#210,#147,#63));
  #5= IFCOWNERHISTORY(#3,#4,$,.NOCHANGE.,$,$,$,1495179818);
  #33= IFCELEMENTASSEMBLY('1P7g7E00000p4qEJKnDpat',#5,'BEAM',$,$,#32,$,'B\\0(?)',.NOTDEFINED.,.REINFORCEMENT_UNIT.);
  #506= IFCREINFORCINGBAR('1P7g7E00005p4qEJKnDpat',#5,'',$,$,#274,#505,'ID591ea1ce-0000-0017-3134-393531373937',$,10.,0.,$,.NOTDEFINED.,$);
  #271= IFCREINFORCINGBAR('1P7g7E00005J4qEJKnDpat',#5,'',$,$,#256,#270,'ID591ea1ce-0000-0015-3134-393531373937',$,10.,0.,$,.NOTDEFINED.,$);
  #246= IFCREINFORCINGBAR('1P7g7E00004p4qEJKnDpat',#5,'',$,$,#227,#245,'ID591ea1ce-0000-0013-3134-393531373937',$,10.,0.,$,.NOTDEFINED.,$);
  #210= IFCREINFORCINGBAR('1P7g7E00004J4qEJKnDpat',#5,'',$,$,#185,#209,'ID591ea1ce-0000-0011-3134-393531373937',$,19.,0.,$,.NOTDEFINED.,$);
  #147= IFCREINFORCINGBAR('1P7g7E00003p4qEJKnDpat',#5,'',$,$,#92,#146,'ID591ea1ce-0000-000f-3134-393531373937',$,19.,0.,$,.NOTDEFINED.,$);
  #63= IFCBEAM('1P7g7E0000134qEJKnDpat',#5,'BEAM','800*400','800*400',#46,#62,'Calcestruzzo_B-1\\0(?)');
```

Figura 43 - Esempio in cui le armature vengono classificate come NOTDEFINED

Per risolvere tali problematiche si è deciso di sviluppare delle funzioni in grado di ricostruire in maniera automatica le proprietà ricercate. Per farlo è stato utilizzato come ambiente di sviluppo il programma per sviluppatori Microsoft Visual Studio 2013. Questo applicativo rientra tra i software IDE (*Integrated development environmnet*) e permette la gestione sia della parte grafica che del codice delle funzioni. Il linguaggio di programmazione utilizzato è Visual Basic .NET.

Il flusso di calcolo dei dati mancanti prevede le seguenti fasi:

- apertura del file IFC dall'ambiente di sviluppo: come già anticipato il file può essere gestito come un semplice file di testo, quindi può essere letto e interpretato dal software;
- lettura e salvataggio delle informazioni in un *object model*: i dati letti dal file vengono memorizzati temporaneamente all'interno dell'applicativo e salvati secondo una struttura gerarchica definita in base alle esigenze di calcolo. In prima analisi è stata pensata un'organizzazione dei dati simile a quella di un file IFC in modo che sia più facile avere la diretta corrispondenza con il *object model*;
- applicazione delle funzioni di calcolo: gli algoritmi avranno in ingresso i dati letti dell'*object model* e restituiranno le informazioni necessarie per i calcoli successivi;
- aggiornamento del *model object*: le informazioni aggiuntive vengono salvate in maniera apposita aggiornando o aggiungendo attributi. Ad esempio alla trave che contiene un determinato elemento d'armatura verrà aggiunto un attributo che permette di mettere in relazione i due oggetti.

La successione delle operazioni viene schematizzata del seguente diagramma di flusso.

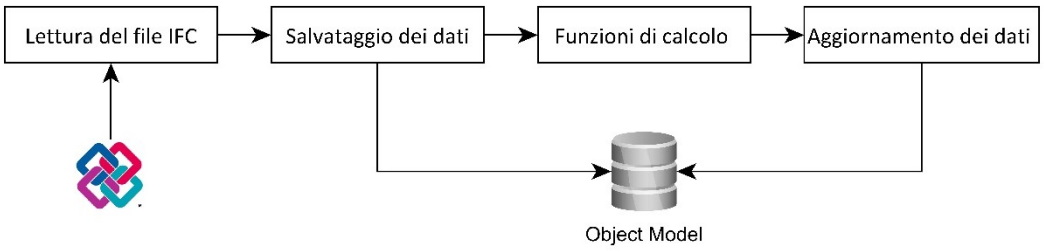


Figura 44 - Flusso di calcolo dei dati mancanti

I successivi paragrafi hanno il compito di analizzare nel dettaglio le funzioni implementate. Per ognuna di esse verrà ripercorso il flusso metodologico che ne ha permesso la scrittura. Oltre alle funzioni che permettono, prima l'associazione delle armature, poi il loro riconoscimento, le analisi preliminari hanno messo in evidenza la necessità di sviluppare una terza funzione in grado di semplificare notevolmente i calcoli. Verrà infatti creata una funzione con l'obiettivo di trasformare le coordinate da un sistema di riferimento ad un altro in modo che i calcoli delle successive funzioni possano essere eseguiti secondo il riferimento appropriato.

Adottare in maniera rigorosa un efficiente flusso metodologico ha permesso di semplificare notevolmente la scrittura del codice. Nonostante la diversa natura delle funzioni, l'utilizzo più opportuno degli strumenti informatici, unito ad un know-how consolidato in azienda, ha permesso, non solo un notevole risparmio di tempo, ma ha anche fornito utili strumenti per ridurre le possibilità di errore. L'analisi e la scrittura delle funzioni hanno seguito il flusso logico illustrato successivamente.

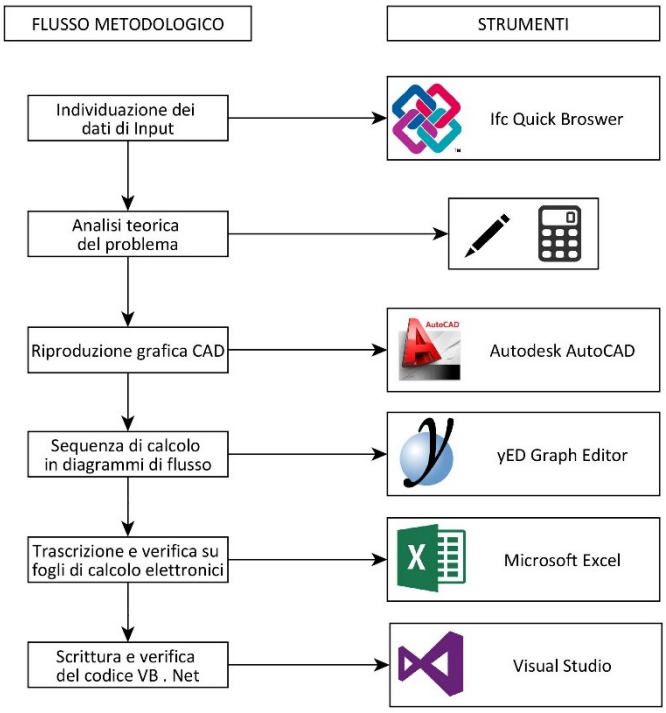


Figura 45 - Flusso metodologico seguito per la determinazione delle funzioni

A partire da dati letti in un file IFC viene in un primo momento riprodotto e analizzato il problema per via teorica. La rappresentazione grafica permette di verificare la correttezza delle analisi e dei risultati attesi. Successivamente viene trascritta la sequenza di calcolo in un diagramma di flusso in modo da semplificare la successiva trascrizione delle operazioni in un foglio di calcolo. Nel caso in cui i risultati ottenuti continuino a corrispondere a quelli attesi può avvenire la trascrizione all'interno dell'ambiente di programmazione. Analizzare lo stesso problema con diversi strumenti permette, in modo rapido ed efficiente, di verificare la corrispondenza dei risultati e rintracciare velocemente eventuali errori. Seguendo il flusso metodologico illustrato vengono quindi analizzati nel dettaglio gli algoritmi di calcolo.

2.3.1 Trasformazioni dei sistemi di riferimento

La funzione ha l'obiettivo di convertire in maniera automatica le coordinate di un punto (o di una lista di punti) da un sistema di riferimento ad un altro. Lo scopo è quello di utilizzare questo algoritmo nelle successive funzioni per semplificare le sequenze di calcolo. Risulta infatti indispensabile eseguire i calcoli nel sistema di riferimento locale dell'elemento analizzato: ad esempio è molto più semplice valutare l'appartenenza di un elemento d'armatura ad una trave se le coordinate sono riferite ad un sistema locale avente origine nel baricentro di una faccia della trave e orientamento degli assi noto.

Dopo la lettura di un file IFC, all'interno dell'*object model* tutte le coordinate sono definite nel sistema di riferimento globale. Ogni elemento avrà però memoria della serie di sistemi di riferimento che collegano il sistema locale a quello globale. Risalendo quindi la gerarchia, saremo in grado di ottenere una matrice di rotazione e la traslazione delle origini che permettono di trasformare le coordinate globali nel sistema locale dell'elemento analizzato.

Osservando il seguente esempio, la funzione deve essere in grado di calcolare la matrice di rotazione e la distanza tra le due origini per passare dal sistema di riferimento $\{S_0\}$ a $\{S_3\}$.

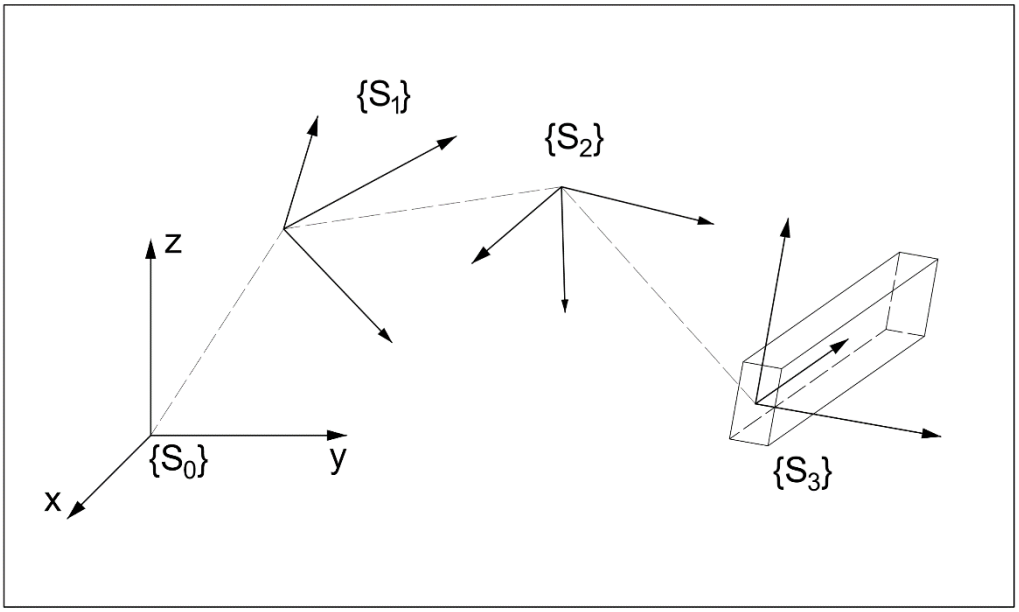


Figura 46 - Serie di sistemi di riferimento

Per fare questo è necessario analizzare ogni passaggio di coordinate da un sistema di riferimento al successivo e, in seguito, generalizzare le funzioni per una serie indefinita di sistemi di riferimenti.

2.3.1.1 Descrizione della rototraslazione e traslazione nel file IFC

Come descritto nel paragrafo 2.1.3.1, in un file IFC la descrizione di un nuovo sistema di riferimento tridimensionale è definita da IfcAxis2Placement3D. Questa entità permette di definire i seguenti attributi:

- Location: indica la posizione dell'origine del nuovo sistema di riferimento;

- **Axis**: indica la direzione del nuovo asse z;
- **RefDirection**: è un vettore la cui proiezione sul piano passante per la nuova origine (*Location*) e perpendicolare all'asse z (*Axis*) rappresenta la direzione dell'asse x.

Il successivo esempio riporta una possibile applicazione dell'entità analizzata. A fianco delle righe è stato aggiunto un commento per indicare l'attributo descritto.

```
#133= IFCAXIS2PLACEMENT3D(#131,#19,#15);
        #131=IFCCARTESIANPOINT((-9.50087,-5,9405,0.75)); /*Location*/
        #19= IFCDIRECTION(0.,0.,1); /*Axis*/
        #15= IFCDIRECTION(0.,1.,0.); /*RefDirection*/
```

Come è possibile osservare, una generica posizione viene definita attraverso le sue coordinate mentre una direzione viene descritta in base alle componenti di un versore. Continuando l'analisi del medesimo esempio, una rototraslazione così definita sposterebbe l'origine del nuovo sistema di riferimento nel punto #131, generando una rotazione positiva della terna di π/2 intorno a z. Infatti, mentre la direzione della z rimane la medesima (riga #19), la direzioni dell'asse x dal nuovo sistema di riferimento coinciderà con la direzione y della precedente terna (riga #15).

2.3.1.2 Traslazione e rotazione semplice

Dato un generico punto P di coordinate descritte rispetto al sistema di riferimento $\{S_0\}$, viene di seguito analizzata la procedura che permette di definire le coordinate di P rispetto ad un generico sistema $\{S_1\}$. L'analisi viene effettuata a partire dai dati letti all'interno del file IFC.

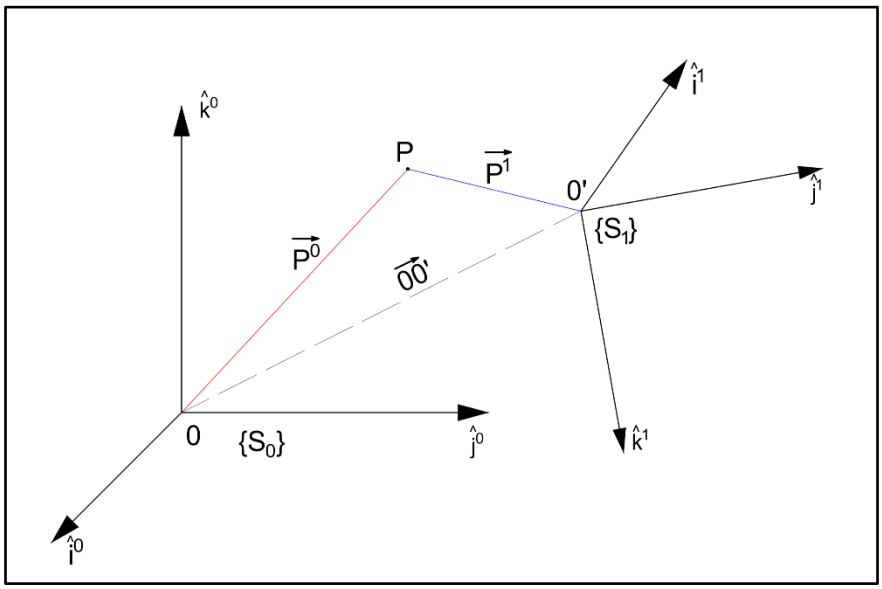


Figura 47 - Trasformazioni di coordinate tra due sistemi di riferimento

La notazione utilizzata ha il seguente significato:

- $\{S_i\}$: indica il sistema di riferimento i-esimo descritto dai versori (i^i, j^i, k^i);
- un generico versore $\hat{v}^i$ è descritto secondo le componenti: $\hat{v}^i = (v_1^i, v_2^i, v_3^i)$;

- $\overrightarrow{P^0}$ indica la posizione del punto P rispetto al sistema di riferimento $\{S_0\}$.

Data una rotazione e traslazione semplice valgono le seguenti ipotesi:

- $\hat{\imath}^0 = (1,0,0)$, versore dell'asse x;
- $\hat{\jmath}^0 = (0,1,0)$, versore dell'asse y;
- $\hat{k}^0 = (0,0,1)$, versore dell'asse z.

Analizzando il contenuto del file IFC avremo:

- la posizione di O^1 descritta con le coordinate *IfcCartesianPoint* nel campo *Location*;
- la direzione $\hat{k}^1$ (versore dell'asse z del sistema $\{S_1\}$) è definita dalle componenti *IfcDirection* nel campo "*Axis*";
- la direzione $\hat{\imath}^1$ (versore dell'asse x del sistema $\{S_1\}$ viene calcolata a partire dalle componenti *IfcDirection* nel campo "*RefDirection*".

Come descritto precedentemente, il valore del versore $\hat{\imath}^1$ è dato dalla proiezione del versore "RefDirection" sul piano perpendicolare a $\hat{k}^1$ e passante per l'origine. In prima analisi viene calcolato il prodotto scalare tra i due versori: in caso di risultato nullo, i due versori saranno ortogonali quindi le componenti di "*RefDirection*" coinciderà con le componenti della sua proiezione sul piano perpendicolare a $\hat{k}^1$; in caso contrario, definita la direzione *RefDirection* $(i^1_{Ref}, j^1_{Ref}, k^1_{Ref})$ si avrà:

- $i^1_1 = i^1_{Ref}$
- $i^1_2 = j^1_{Ref}$
- $i^1_3 = -\frac{i^1_1 \cdot k^1_1}{k^1_3} - \frac{i^1_2 \cdot k^1_2}{k^1_3}$

In questo modo si ottiene la direzione del versore dell'asse x ($\hat{\imath}^1$) perpendicolare al versore dell'asse z ($\hat{k}^1$); la direzione del versore $\hat{\jmath}^1$ viene invece calcolata sfruttando la proprietà del prodotto vettoriale tra i due versori precedenti imponendo il segno in modo da ottenere una terna destrorsa:

$$\hat{\jmath}^1 = -\left(\hat{\imath}^1 \times \hat{k}^1\right) = -det \begin{bmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ i^1_1 & i^1_2 & i^1_3 \\ k^1_1 & k^1_2 & k^1_3 \end{bmatrix}$$

Da questo avremo:

- $j^1_1 = -(i^1_2 \cdot k^1_3 - i^1_3 \cdot k^1_2)$
- $j^1_2 = -(i^1_1 \cdot k^1_3 - i^1_3 \cdot k^1_1)$
- $j^1_3 = -(i^1_1 \cdot k^1_2 - i^1_2 \cdot k^1_1)$

Dopo aver definito i versori del nuovo sistema di riferimento, viene calcolata la matrice di rotazione R [3x3] che, insieme alla traslazione, permette di definire le coordinate di un punto secondo il nuovo sistema di riferimento attraverso la relazione:

$$\overrightarrow{P^1} = \overrightarrow{00'} + R_{0\to1} \cdot \overrightarrow{P^0}$$

Dove:

- $\overrightarrow{00'}$ descrive la traslazione rigida dal sistema di riferimento da $\{S_0\}$ a $\{S_1\}$;
- $R_{0\to1}$ rappresenta la matrice di rotazione degli assi dal sistema $\{S_0\}$ a $\{S_1\}$.

Vengono di seguito descritte le caratteristiche della matrice R a partire da un caso semplificato con traslazione $\overrightarrow{00'}$ nulla.

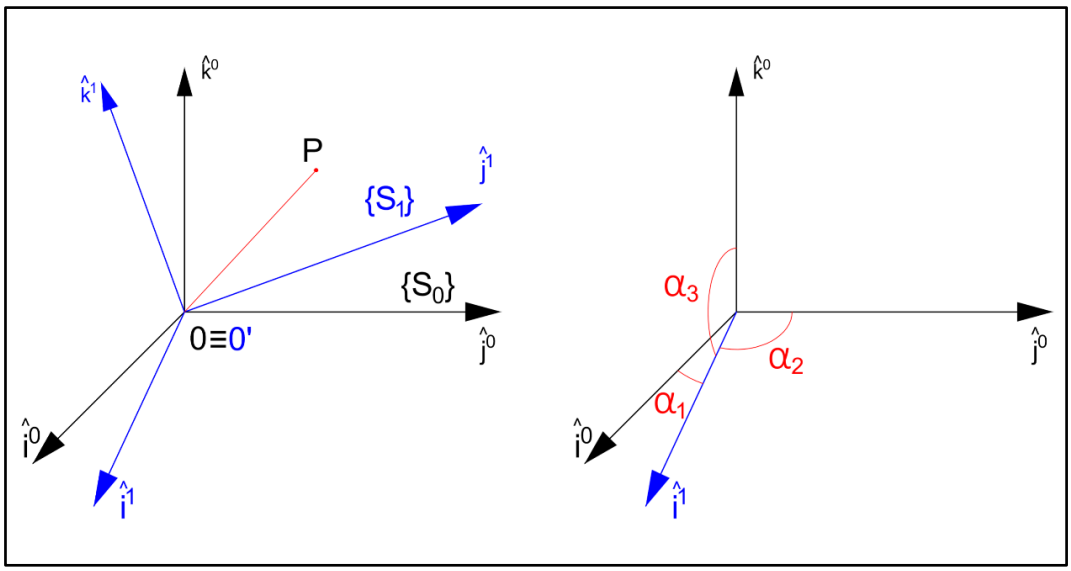


Figura 48 - Trasformazione di coordinate a seguito della rotazione degli assi

Nel caso rappresentato le coordinate del punto P sono definite secondo la relazione:

$$\overrightarrow{P^1} = R_{0\to1} \cdot \overrightarrow{P^0}$$

La matrice di rotazione rappresenta una trasformazione di coordinate che permette di descrivere la posizione di uno stesso punto in due terne aventi origine in comune. I vettori colonna della matrice sono rappresentati dai coseni direttori degli assi della terna ruotata. Definita la matrice di rotazione come:

$$R_{0\to1} = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}$$

Dove il primo vettore colonna, come rappresentato nell'immagine precedente, è descritto da:

$$\overrightarrow{r_0} = \begin{bmatrix} \cos\alpha_1 \\ \cos\alpha_2 \\ \cos\alpha_3 \end{bmatrix}$$

Dati due generici vettori $\vec{u}$ e $\vec{v}$ il coseno dell'angolo compreso viene calcolato sfruttando la definizione del prodotto scalare:

$$\cos\alpha = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |\vec{v}|}$$

Dove, per componenti, verrà calcolato come:

$$\cos\alpha = \frac{u_1 \cdot v_1 + u_2 \cdot v_2 + u_3 \cdot v_3}{\left(\sqrt{{u_1}^2 + {u_2}^2 + {u_3}^2}\right) \cdot \left(\sqrt{{v_1}^2 + {v_2}^2 + {v_3}^2}\right)}$$

Applicando le precedenti relazioni alle componenti dei versori che definiscono le terne, è possibile definire tutte le componenti della matrice di rotazione a partire dai dati letti in un file IFC. Calcolata la matrice di rotazione devono essere verificate le seguenti proprietà:

- R dev'essere una matrice ortogonale ossia deve valere la relazione: $R^T R = I$;
- il determinante di una matrice di rotazione dev'essere sempre pari a 1;
- la trasposta della matrice R deve coincidere con la sua inversa.

Essendo quindi la matrice R sempre invertibile, vale la relazione:

$$\overrightarrow{P^0} = {R_{0\to1}}^{-1} \cdot \overrightarrow{P^1}$$

2.3.1.3 Traslazione e rotazione di una serie di sistemi di riferimento

Data una serie di sistemi di riferimento dipendenti, si descrive il flusso di calcolo che permette di definire la traslazione e la matrice di rotazione tra il sistema di riferimento globale $\{S_0\}$ e una generica terna $\{S_i\}$. L'analisi viene effettuata per semplicità su una serie composta da 3 terne ma lo stesso processo può essere iterato per un numero indefinito di sistemi di riferimento.

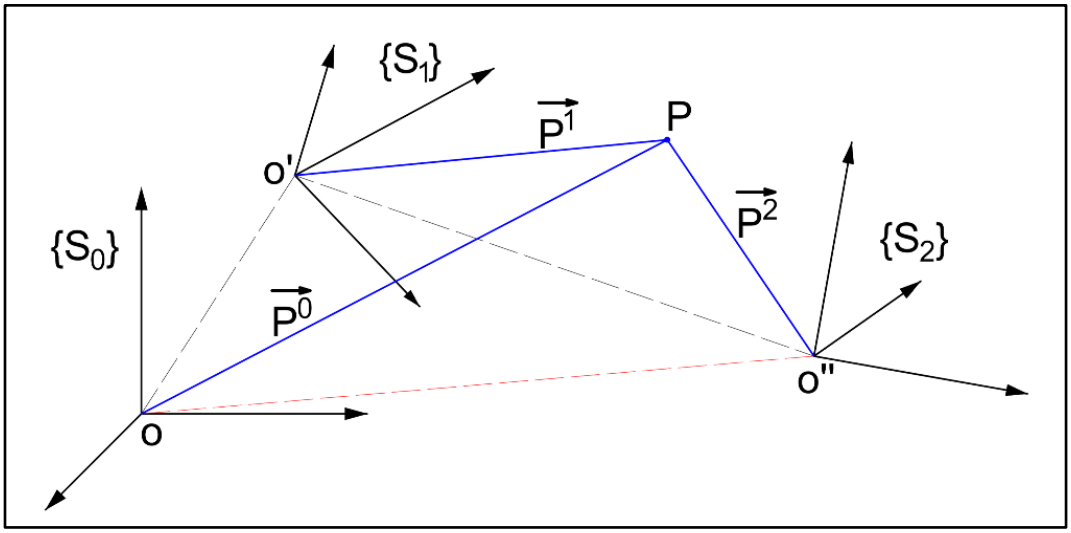


Figura 49 - Trasformazioni di coordinate per una serie di sistemi di riferimento

Dal precedente paragrafo sono note le coordinate del punto P, la matrice di rotazione e la distanza tra le origini per i sistemi $\{S_0\}$ e $\{S_1\}$. Viene quindi definito un ulteriore sistema di riferimento $\{S_2\}$ in funzione della terna $\{S_1\}$. Lo scopo dei seguenti calcoli è quello di trovare la matrice di rotazione e la distanza tra le origini per i sistemi $\{S_0\}$ e $\{S_2\}$.

La distanza tra le origini $\overrightarrow{00''}$ non può essere calcolata come somma diretta tra $\overrightarrow{00'}$ e $\overrightarrow{0'0''}$ perché questi vettori sono definiti in due sistemi di riferimento differenti.

Sfruttando la matrice $R_{0\to1}$ precedentemente calcolata, si avrà quindi:

$$\overrightarrow{00''} = \overrightarrow{00'} + {R_{0\to1}}^{-1} \cdot \overrightarrow{0'0''}$$

Successivamente può essere calcolata la matrice $R_{1\to2}$ seguendo lo stesso flusso logico del precedente paragrafo.

Sfruttando le proprietà della matrice R, la rotazione tra il sistema $\{S_0\}$ e $\{S_2\}$ può essere calcolata attraverso il prodotto matriciale:

$$R_{0\to2} = R_{1\to2} \cdot R_{0\to1}$$

Le coordinate del punto P rispetto al sistema $\{S_2\}$ potranno quindi essere definite come:

$$\overrightarrow{P^2} = \overrightarrow{00''} + R_{0\to2} \cdot \overrightarrow{P^0}$$

Data una serie indefinita di sistemi di riferimento sarà quindi possibile calcolare le coordinate rispetto a qualsiasi sistema di riferimento $\{S_i\}$ iterando il precedente flusso di calcolo. L'intero processo viene riprodotto per via grafica attraverso un diagramma di flusso nell'allegato 5.3.

2.3.1.4 Trascrizione nel linguaggio vb .NET

All'interno dell'ambiente di sviluppo Visual Studio viene descritto il processo che ha permesso la stesura del codice con il linguaggio di programmazione vb .NET. L'obiettivo in questa fase è quello di creare una funzione che possa essere richiamata in maniera ricorsiva a seconda del numero di sistemi di riferimento da processare. Rispetto all'analisi precedente, risulta quindi necessario trovare un meccanismo per automatizzare la sequenza di operazioni.

All'inizio del calcolo, è necessario inizializzare la matrice di rotazione e la traslazione tra il sistema $\{S_0\}$ e il generico sistema $\{S_i\}$ nel seguente modo:

```
Dim matrRot(2, 2) As Double
Dim NewLocation(2) As Double
For i = 0 To 2
            NewLocation(i) = 0
            For j = 0 To 2
                If i = j Then
                    matrRot(i, j) = 1
                Else
                    matrRot(i, j) = 0
                End If
            Next
Next
```

Le dichiarazioni iniziali definiscono una matrice di rotazione 3x3 e il vettore "Location" descritti come:

$$matrRot(2,2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, NewLocation(2) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

La funzione sviluppata ha in ingresso gli attributi definiti dall'entità *IfcAxis2Placement3D* (*Location*, *Axis*, *RefDirection*), la matrice di rotazione e la distanza tra le origini dei sistemi di

riferimento. In base alla nuova terna, l'algoritmo restituisce i parametri aggiornati della matrice e della distanza di traslazione. Vengono eseguite nell'ordine le seguenti operazioni:

1. moltiplicazione riga per colonna dell'inverso della matrice di rotazione per l'attributo *Location* dell'entità *IfcAxis2Placement*;
2. somma del vettore ottenuto con il vettore *NewLocation*(2) così da ottenere la nuova distanza di traslazione;
3. calcolo i versori degli assi definiti dall'entità *IfcAxis2Placement*;
4. calcolo di una nuova matrice di rotazione;
5. moltiplicazione riga per colonna tra la nuova matrice di rotazione e *matrRot*(2,2);
6. restituzione dei valori aggiornati di *matrRot*(2,2) e di *NewLocation*(2);

Questa funzione, essendo richiamata per ogni sistema di riferimento da $\{S_0\}$ a $\{S_i\}$, fornisce i parametri per definire in modo semplice la trasformazione di coordinate tra il sistema globale a quello locale. Volendo ottenere un preciso orientamento della terna del sistema di riferimento locale di ogni elemento, la funzione viene richiamata un'ultima volta imponendo l'orientamento finale in modo da avere:

- centro nel baricentro di una faccia di estrusione;
- asse x coincidente con l'asse di estrusione;
- asse z e asse y paralleli rispettivamente ai lati che definiscono l'altezza e la base della sezione d'estrusione.

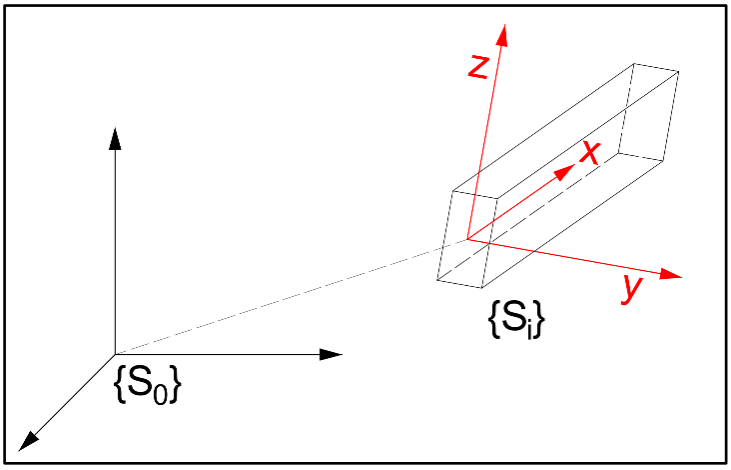


Figura 50 - Definizione dell'orientamento del sistema di riferimento locale

Per calcolare le coordinate di un punto secondo il nuovo sistema di riferimento sarà quindi sufficiente:

- sottrarre le coordinate del punto a quelle della distanza di traslazione;
- eseguire la moltiplicazione tra la matrice di rotazione e le coordinate del punto precedentemente traslato.

Inserendo queste due semplici istruzioni in una funzione che ha in ingresso la matrice di rotazione (*pMatrRot*), le coordinate di traslazione e quelle dal punto da rototraslare (*pPoint*), si avrà:

```
Public Function ConvertPoint(ByVal pMatrRot(,) As Double,
                                ByVal pLocation() As Double,
                                ByRef pPoint() As Double) As Boolean
        Try
            Dim mypoint(2) As Double

 'calcolo le coordinate traslate
mypoint(0) = pPoint(0) - pLocation(0)
mypoint(1) = pPoint(1) - pLocation(1)
mypoint(2) = pPoint(2) - pLocation(2)

'calcolo le coordinate ruotate
pPoint(0) = pMatrRot(0, 0) * mypoint(0) + pMatrRot(0, 1) * mypoint(1) + pMatrRot(0, 2) * mypoint(2)
pPoint(1) = pMatrRot(1, 0) * mypoint(0) + pMatrRot(1, 1) * mypoint(1) + pMatrRot(1, 2) * mypoint(2)
pPoint(2) = pMatrRot(2, 0) * mypoint(0) + pMatrRot(2, 1) * mypoint(1) + pMatrRot(2, 2) * mypoint(2)

            Return True
        Catch ex As Exception
            MsgBox(ex.ToString, MsgBoxStyle.Critical)
            Return False
        End Try
End Function
```

La funzione, restituendo le coordinate di un punto secondo un nuovo sistema di riferimento, può essere applicata al singolo punto oppure in maniera ricorsiva ad una lista di coordinate.

Per ogni elemento analizzato (trave o pilastro) vengono opportunamente richiamate le precedenti funzioni in modo che le coordinate dei punti possano essere definite nel sistema locale dell'elemento. Nelle seguenti analisi sarà possibile notare la semplificazione del calcolo dovuta all'introduzione delle funzioni di rototraslazione.

2.3.2 Associazione elemento-armatura

La funzione descritta in questo capitolo ha l'obiettivo di individuare, all'interno di un file IFC, le armature contenute o passanti in un elemento in calcestruzzo (trave o pilastro). L'analisi della sequenza di operazioni ha portato alla definizione degli input indispensabili per ottenere le relazioni cercate. Le informazioni necessarie sono:

- i dati geometrici dell'elemento in calcestruzzo, in particolare le dimensioni di base, altezza e lunghezza di estrusione;
- le coordinate dei punti della polilinea che descrive le armature nello spazio rispetto il sistema di riferimento globale;
- la traslazione e la matrice di rotazione per il passaggio di coordinate dal sistema globale a quello locale dell'elemento.

2.3.2.1 IfcReinforcingBar

L'entità IFC che definisce gli elementi d'armatura è rappresentata dalla voce *IfcReinforcingBar*; osservando i principali attributi dal grafico delle gerarchie. Questa entità, richiamando la classe *IfcReinforcingBarType*, è in grado di definire:

- la posizione e la rappresentazione geometrica dell'armatura;
- il diametro nominale (*NominalDiameter*);
- l'area effettiva della sezione trasversale (*CrossSectionArea*);
- la lunghezza totale delle barre (*BarLength*);
- il tipo di superficie, distinguendo la barra liscia da quella ad aderenza migliorata (*BarSurface*).

Nonostante nel file IFC le armature siano rappresentate geometricamente in diversi modi, l'analisi effettuata per la lettura di questa entità permette la definizione geometrica mediante l'estrusione di un cerchio di diametro noto lungo un percorso definito dalle coordinate di punti appartenenti ad una polilinea. All'interno del *model object* vengono quindi salvate, oltre agli attributi precedentemente letti, le coordinate nel sistema di riferimento globale dei punti che compongono le polilinee.

La peculiarità dell'entità *IfcReinforcingBar* risiede nel fatto che attraverso la sua definizione possono essere rappresentati più ferri d'armatura. Introducendo l'esempio dell'armatura di una trave, in alcune esportazioni è possibile osservare come le armature con uguali caratteristiche e ugual passo siano raggruppate in un'unica entità avente più di una rappresentazione geometrica. Se da una parte questa proprietà potrebbe semplificare l'analisi del problema, dall'altra potrebbero descrivere situazioni limite in grado di compromettere la veridicità dei risultati. Viene di seguito riportato il caso limite in cui la rappresentazione di una sola *IfcReinfocingBar* porta a un errore nei calcoli.

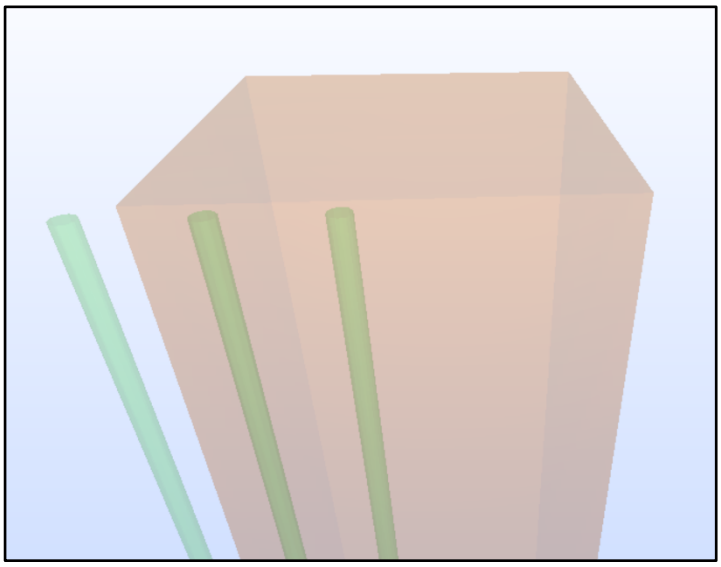


Figura 51 - Rappresentazione dell'armatura con una sola entità IfcReinforcingBar

L'esempio riporta l'esportazione IFC di un pilastro e di 3 barre d'armatura descritte tramite un'unica entità *IfcReinforcingBar*. Si osserva che, mentre due barre sono interamente contenute all'interno dell'elemento in calcestruzzo, una terza è posizionata nello spazio esterno. Volendo mettere in relazione l'armatura all'elemento che la contiene, è evidente come l'associazione non possa essere fatta utilizzando l'entità *IfcReiforcingBar* ma ci sia la necessità di analizzare gli elementi separatamente. Per questa ragione, dopo la lettura del file IFC, gli elementi d'armatura vengono salvati nel *model object* separati gli uni dagli altri.

Le funzioni sviluppate in seguito andranno quindi a eseguire i calcoli analizzando in modo separato ogni elemento d'armatura. Nel caso in cui uno stesso elemento d'armatura sia contenuto in più elementi (come un ferro d'armatura longitudinale che percorre più travi) ai fini del *code checking* strutturale, è indispensabile che tale elemento venga messo in relazione con tutti gli quelli che lo contengono.

2.3.2.2 *Analisi della funzione*

Per ogni elemento in calcestruzzo vengono analizzate le armature verificando geometricamente se siano o meno contenute all'interno del volume analizzato. Applicando la funzione di trasformazione dei sistemi di riferimento per passare dal sistema globale a quello locale, è possibile definire tutte le coordinate dell'armatura in funzione dell'elemento analizzato; inoltre, conoscendo le principali dimensioni (base, altezza e profondità d'estrusione), può essere definito in modo univoco il volume dell'elemento nello spazio.

Eseguendo il controllo che ogni punto di ogni armatura sia contenuto all'interno di un elemento, la funzione potrebbe restituire i risultati in troppo tempo e in modo poco efficace. Per questo motivo il seguente flusso operativo è stato studiato ponendo particolare attenzione all'ottimizzazione del codice, evitando quindi, passaggi superflui o ridondanti.

Secondo quest'ottica, è stato studiato un algoritmo che permette di analizzare se nella lista di punti che identifica la polilinea di un'armatura, vi siano punti consecutivi aventi le stesse coordinate. È infatti stato riscontrato come la traduzione delle diverse rappresentazioni in punti di una polilinea, possa riportare in successione punti con coordinate coincidenti. Avendo in ingresso la lista di punti della polilinea, la funziona restituirà la lista depurata dagli eventuali punti ridondanti.

```
Public Function RemovePointCoinc(ByVal pLstPoint As List(Of Cls_IFC_Point_3D))
        Try
            Dim myListPoint As New List(Of Cls_IFC_Point_3D)
            myListPoint.Add(pLstPoint(0))

            For i = 0 To pLstPoint.Count - 2
                If PntCoinc(pLstPoint(i), pLstPoint(i + 1)) = False Then
                    myListPoint.Add(pLstPoint(i + 1))
                End If
            Next

            Return myListPoint
        Catch ex As Exception
            Return pLstPoint
        End Try
End Function

Public Function PntCoinc(ByVal p1 As Cls_IFC_Point_3D, ByVal p2 As Cls_IFC_Point_3D)

        Dim tolleranzanza As Double = 0.01  'tolleranza del decimo di millimetro

        Try
            If Math.Abs(p1.X - p2.X) < tolleranzanza then
                if Math.Abs(p1.Y - p2.Y) < tolleranzanza then
                    if Math.Abs(p1.Z - p2.Z) < tolleranzanza then
                Return True
                    Else: Return False
                Else: Return False
            Else: Return False
            End If

        Catch ex As Exception
            Return False
        End Try
    End Function
```

Ottenendo la lista delle armature depurate dai punti ridondanti e decritte dalle coordinate definite rispetto al sistema di riferimento dell'elemento in esame, è possibile procedere con

il confronto geometrico per verificare la relazione d'appartenenza. Per alleggerire il flusso di calcolo sono stati definiti due algoritmi con obiettivi differenti: il primo, definito come "funzione d'esclusione", ha il compito di effettuare una scrematura andando ad escludere le armature che sicuramente non appartengono all'elemento; il secondo, definito invece "funzione d'inclusione", è un'analisi più approfondita che permette di stabilire se vi sia o meno la relazione tra elementi. Nell'immagine seguente viene riportato un elemento in calcestruzzo orientato in modo generico nello spazio avente dimensioni note; conoscendo le coordinate dell'armatura nel sistema di riferimento $\{S_i\}$ è possibile verificare la relazione tra elementi con dei semplici controlli sulle coordinate.

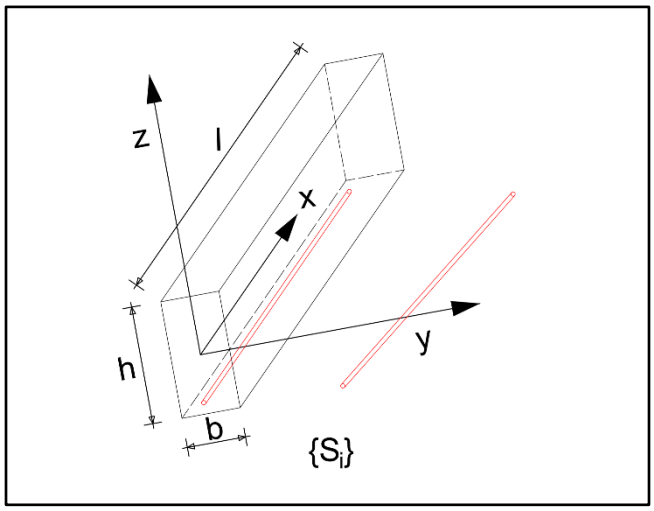


Figura 52 - Analisi della relazione tra elemento e armatura

Entrando nel dettaglio dell'algoritmo, la funzione d'esclusione è composta da 6 disuguaglianze che permettono di stabilire se l'armatura non appartiene all'elemento analizzato. Per rendere più comprensibile la trattazione, viene riportato il precedente problema sul piano xy del sistema di riferimento $\{S_i\}$.

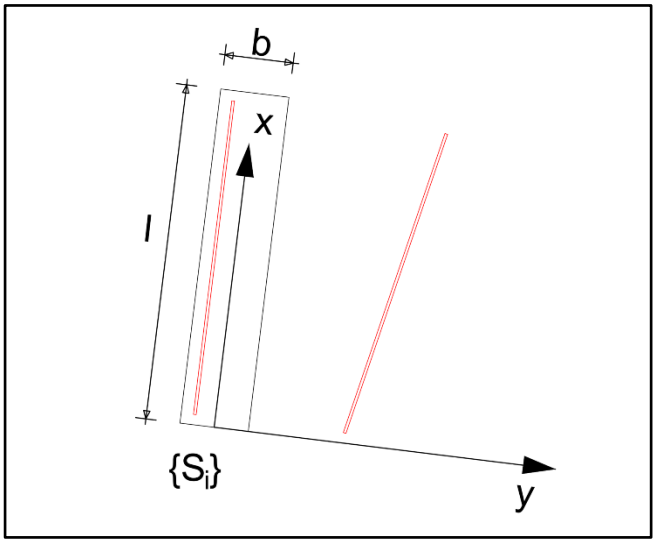


Figura 53 - Analisi della relazione tra elemento e armatura sul piano xy

Per ogni armatura vengono analizzate le coordinate della lista di punti che ne definiscono la polilinea. Lo studio viene effettuato dividendo il problema per i tre assi che descrivono la terna, in particolare, analizzando il problema lungo l'asse y, vengono salvati 2 valori:

- y_{max}: valore massimo delle coordinate y dei punti che descrivono la polilinea;
- y_{min}: valore minimo delle coordinate y dei punti che descrivono la polilinea.

Note le dimensioni dell'elemento, per l'asse y vengono analizzate le seguenti disuguaglianze:

$$y_{max} < -\frac{b}{2}, y_{min} > +\frac{b}{2}$$

Estendendo il caso agli altri assi si avrà:

$$z_{max} < -\frac{h}{2}, z_{min} > +\frac{h}{2}$$

$$x_{max} < 0, x_{min} > l$$

Se una delle precedenti disuguaglianze dovesse risultare verificata, allora l'armatura non appartiene sicuramente all'elemento analizzato. Attraverso questa prima analisi si è in grado di escludere la quasi totalità di armature non incluse nell'elemento senza appesantire eccessivamente il processo di calcolo.

Le armature non escluse dalla precedente funzione vengono ulteriormente analizzate attraverso la funzione d'inclusione. Studiando ogni punto che descrive un'armatura, se almeno uno è compreso all'interno del volume allora l'armatura appartiene all'elemento. In termini matematici, dato il punto di un'armatura P (x_p, y_p, z_p), devono essere rispettate le seguenti disuguaglianze:

$$0 \leq x_p \leq l$$

$$-\frac{b}{2} \leq y_p \leq \frac{b}{2}$$

$$-\frac{h}{2} \leq z_p \leq \frac{h}{2}$$

Se anche solo una delle precedenti disuguaglianze non dovesse essere verificata, allora il punto analizzato non è interno al volume dell'elemento.

Il test delle funzioni su delle esportazioni IFC di progetti reali, ha evidenziato come le sole 2 funzioni precedentemente analizzate non siano sufficienti a gestire tutte le possibili casistiche. Esistono infatti alcune rappresentazioni dell'armatura che, processate con le funzioni, non permettono di ottenere i risultati attesi. Per questo motivo si è resa necessaria un'ulteriore analisi e la modifica del codice di calcolo. Le maggiori problematiche sono state riscontrate analizzando l'armatura avente due punti in successione esterni al volume ma con la congiungente che attraversa tutto l'elemento. In questo caso infatti, nonostante superi i controlli della funzione d'esclusione, l'inclusione non risulterà verificata perché non vi è alcun punto descritto all'interno al volume.

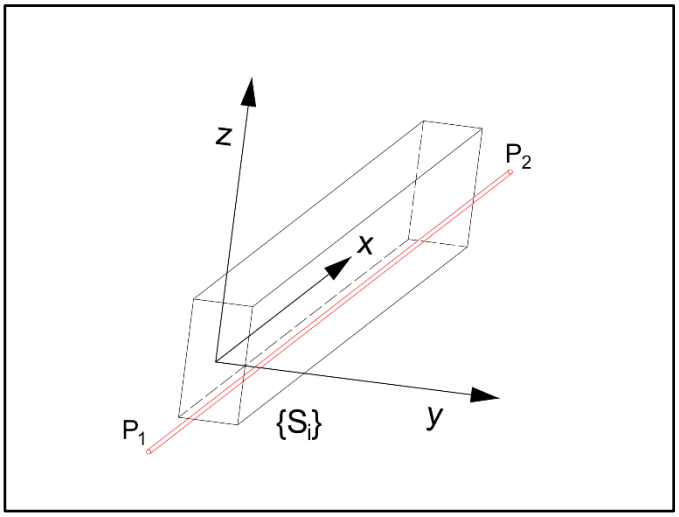


Figura 55 - Armatura descritta da due punti esterni al volume

Il semplice esempio riporta l'armatura definita attraverso la polilinea descritta dai solo punti P_1 e P_2. Per risolvere in modo corretto questo caso, è stata analizzata un'ulteriore funzione in grado di effettuare la verifica per inclusione su più punti dell'armatura oltre a quelli che ne descrivono la polilinea. Le seguenti analisi vengono effettuate a partire da un progetto reale in cui le armature longitudinali di una trave vengono modellate prolungandone le estremità per innestarsi sui pilastri. Per semplicità viene analizzato l'esempio precedente sul piano xy.

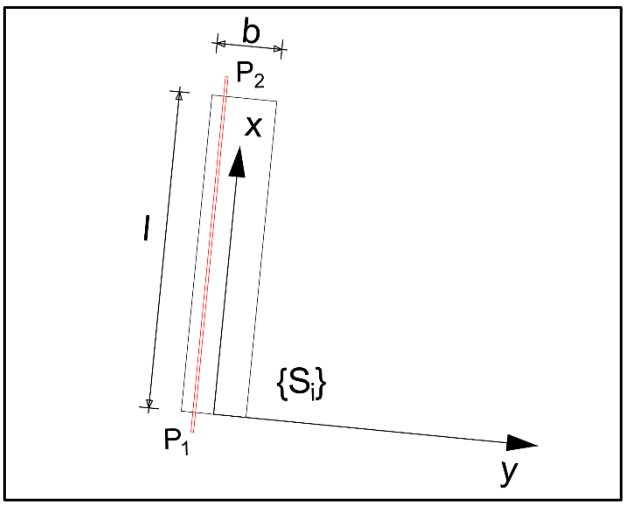


Figura 54 - Analisi dell'armatura sul piano xy

In una lista di coordinate che descrivono l'armatura, per individuare due punti consecutivi che rappresentino la situazione dell'immagine precedente, possono verificarsi diverse situazioni.

La prima situazione analizzata prevede le seguenti disuguaglianza:

- $-\frac{b}{2} \leq y_{P_1} \leq \frac{b}{2}, -\frac{h}{2} \leq z_{P_1} \leq \frac{h}{2}$ per il punto P1;
- $-\frac{b}{2} \leq y_{P_2} \leq \frac{b}{2}, -\frac{h}{2} \leq z_{P_2} \leq \frac{h}{2}$ per il punto P2;
- $x_{P_1} < 0 \; e \; x_{P_2} > l$ oppure $x_{P_2} < 0 \; e \; x_{P_1} > l$.

In questo caso si avrebbe quindi un ferro d'armatura che interseca l'elemento avente le estremità esterne al volume solo lungo la direzione x.

Una seconda situazione può essere data dal verificarsi delle disuguaglianze:

- $-\frac{b}{2} \leq y_{P_1} \leq \frac{b}{2}$;
- $-\frac{b}{2} \leq y_{P_2} \leq \frac{b}{2}$;
- $x_{P_1} < 0 \; e \; x_{P_2} > l$ oppure $x_{P_2} < 0 \; e \; x_{P_1} > l$;
- $z_{P_1} < -\frac{h}{2} \; e \; z_{P_2} > -\frac{h}{2}$ oppure $z_{P_2} < -\frac{h}{2} \; e \; z_{P_1} > -\frac{h}{2}$.

La situazione così descritta vede un ferro d'armatura che interseca l'elemento e ha le estremità esterne al volume lungo z e lungo x.

Nel caso più generico possibile si può avere:

- $x_{P_1} < 0 \; e \; x_{P_2} > l$ oppure $x_{P_2} < 0 \; e \; x_{P_1} > l$;
- $y_{P_1} < -\frac{b}{2} \; e \; y_{P_2} > -\frac{b}{2}$ oppure $y_{P_2} < -\frac{b}{2} \; e \; y_{P_1} > -\frac{b}{2}$;
- $z_{P_1} < -\frac{h}{2} \; e \; z_{P_2} > -\frac{h}{2}$ oppure $z_{P_2} < -\frac{h}{2} \; e \; z_{P_1} > -\frac{h}{2}$.

In questo caso l'armatura interseca l'elemento ma le estremità sono esterne al volume lungo tutte le direzioni. Generalizzando il problema in modo da tenere conto di tutti i casi possibili (non solo quelli riferiti al piano xy) possono verificarsi 7 diverse situazioni.

Individuati quindi i due punti bisogna verificare che la congiungente intersechi realmente l'elemento; a questo scopo è stata studiata una funzione in grado di determinare, al netto di una certa tolleranza, se vi sia o meno l'intersezione. Analizzando l'esempio sul piano xy, risulta facile estendere il ragionamento a tutte le altre casistiche: in particolare, note le coordinate dei punti P_1 e P_2, vengono calcolate le coordinate del punto P_0 avente coordinata x pari alla metà della lunghezza dell'elemento, e coordinate y e z calcolate in modo che il punto P_0 appartenga alla retta passante per i punti P_1 e P_2. Le coordinate vengono calcolate attraverso l'equazione della retta nello spazio scritta in forma parametrica, dove, definito t come parametrizzazione della retta, si avrà:

$$t = \frac{x_{P_0} - x_{P_1}}{x_{P_2} - x_{P_1}}, \text{ dove } x_{P_0} = \frac{l}{2}$$

$$y_{P_0} = \frac{y_{P_1} + t}{y_{P_2} - y_{P_1}}$$

$$z_{P_0} = \frac{z_{P_1} + t}{z_{P_2} - z_{P_1}}$$

Note quindi le coordinate di un nuovo punto, calcolato in corrispondenza della mezzeria della lunghezza dell'elemento e appartenente alla retta che definisce l'armatura, viene eseguito l'algoritmo della funzione d'inclusione per valutare se il punto sia interno al volume dell'elemento. Nel caso in cui il punto trovato appartenga all'elemento allora vi è intersezione; in caso contrario l'analisi dovrà proseguire analizzando un nuovo punto appartenente alla retta.

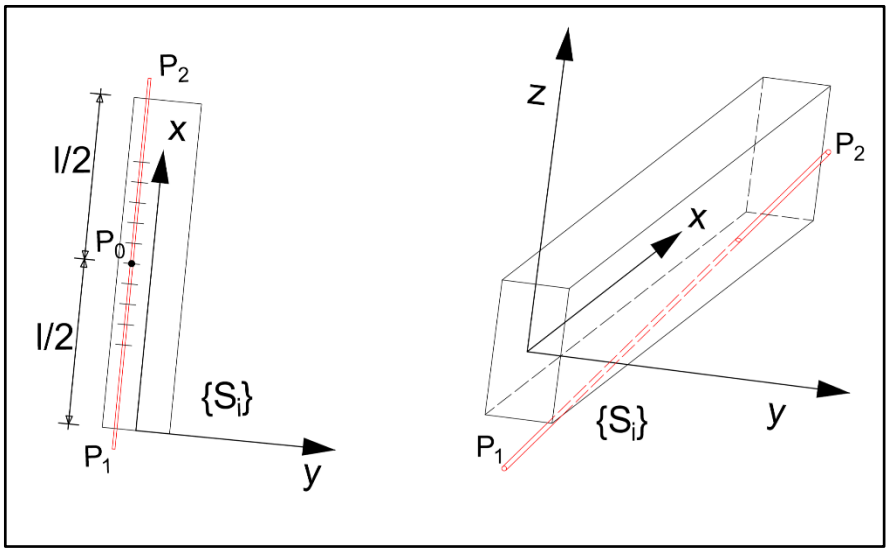


Figura 56 - Individuazione dei nuovi punti da verificare

La figura mostra infatti sul piano xy e nello spazio il problema di individuare un punto dell'armatura appartenente all'elemento. Proseguendo con il calcolo vengono analizzati altri punti aventi come distanza dal punto P_0 (proiettata sull'asse x) una certa tolleranza. Essendo il controllo puntuale, viene stabilita una tolleranza sotto la quale, anche ci fosse una reale intersezione, si può trascurare la relazione tra armatura ed elemento in calcestruzzo ai fini del *code checking* strutturale. Come nel caso di P_0, per ogni nuovo punto individuato lungo l'asse x, vengono calcolate le coordinate y e z in modo che il punto appartenga alla retta P_1P_2; nel caso in cui la funzione di inclusione, applicata su questo nuovo punto, desse un riscontro negativo si procederebbe all'individuazione di nuovi punti in successione a destra e sinistra di P_0 ad una distanza incrementale in funzione della tolleranza scelta. Nel caso in cui la funzione d'inclusione continui a dare esito negativo e la distanza da P_0 al punto in analisi risulti maggiore di l/2 allora l'armatura non interseca l'elemento. Come accennato in precedenza, i calcoli vengono estesi lungo l'asse y (confrontando le lunghezze con la dimensione della base della sezione) e lungo z (confrontando l'altezza della sezione).

Questo processo, iterato per ogni elemento, permette di ottenere la lista delle armature contenute nella trave o nel pilastro in analisi. L'ordine delle operazioni viene riepilogato sotto forma di diagramma di flusso nell'allegato 5.4.

2.3.3 Distinzione armatura longitudinale-trasversale

Ottenuta la lista delle armature contenute in un elemento, ai fini del *code checking* strutturale, risulta indispensabile la classificazione dei ferri in base alla loro funzione. In particolare si procede con l'analisi di un flusso di operazione in grado di distinguere le

armature a seconda che siano longitudinali, trasversali o non definite. Anche per questa funzione il calcolo avverrà sfruttando l'algoritmo di trasformazione dei sistemi di riferimento. Vengono di seguito analizzati due diversi flussi di calcolo che permettono di definire la tipologia d'armatura.

2.3.3.1 *Armatura trasversale*

La sequenza di operazioni ha l'obiettivo di verificare se sussistano o meno le condizioni che descrivono un'armatura come trasversale. Per ogni ferro, nota la polilinea descritta mediante punti definiti nel sistema di riferimento locale della trave o del pilastro, viene effettuato un controllo geometrico stabilendo se tutti i punti appartengano ad un piano perpendicolare all'asse d'estrusione dell'elemento. La descrizione dei calcoli avviene con l'ausilio di un semplice esempio.

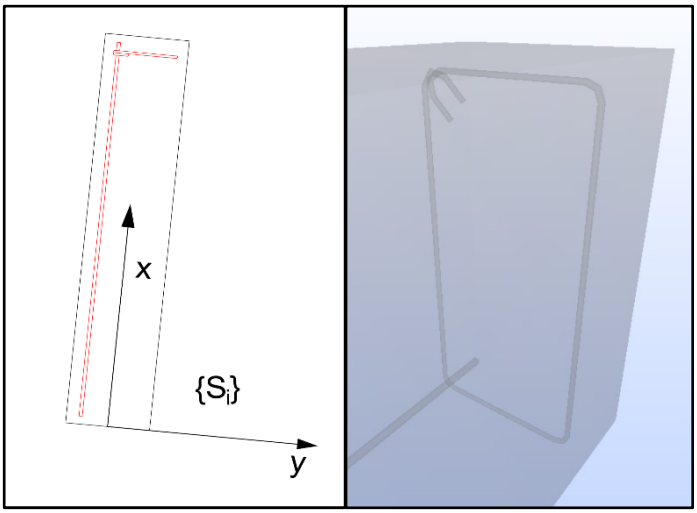


Figura 57 - Analisi della classificazione dell'armatura sul piano e nello spazio

Viene quindi analizzato un generico elemento in calcestruzzo contenente un'armatura longitudinale e una staffa. Come è possibile notare nella proiezione del problema sul piano xy, eseguendo alcuni confronti geometrici sulle coordinate si è in grado di distinguere la tipologia d'armatura. L'analisi secondo il sistema di riferimento locale permette una notevole semplificazione: sarà infatti sufficiente eseguire il controllo sulle sole coordinate x (anziché calcolare il piano che contiene i punti) per stabilire se l'armatura sia di tipo trasversale. Date le coordinate della lista di punti di un'armatura, è di tipo trasversale se per ogni punto vale la relazione:

$$x_0 - t \leq x_i \leq x_o + t$$

Dove:

- x_0 è la coordinata x del primo punto dell'armatura;
- x_i è la coordinata x del generico punto dell'armatura;
- t è la tolleranza che deve tenere conto della sovrapposizione delle estremità di una staffa e delle possibili imprecisioni nella modellazione.

2.3.3.2 *Armatura longitudinale*

Il processo per l'individuazione dell'armatura longitudinale prevede un flusso di operazioni simile al caso precedente. L'obiettivo è quello di determinare il piano che contiene l'armatura e verificare che sia parallelo alla direzione d'estrusione. Differentemente dall'algoritmo precedente, il flusso di calcolo per l'individuazione dell'armatura longitudinale prevede l'analisi rispetto all'asse y e z. Non sarà quindi sufficiente analizzare una sola coordinata ma bisognerà eseguire il confronto calcolando il piano che contiene l'armatura. Il calcolo può seguire strade diverse a seconda del numero di punti che descrivono la polilinea di un'armatura.

Come prima analisi, viene descritto il caso di armatura definita da soli due punti. Essendo matematicamente impossibile individuare l'equazione di un piano, il confronto avverrà calcolando le componenti del vettore che congiunge i due punti. Definiti i due punti che descrivono l'armatura P_0 e P_1 ed essendo $\vec{r}$ il vettore che li unisce, si avrà:

$$r_x = x_{P_1} - x_{P_0}$$

$$r_y = y_{P_1} - y_{P_0}$$

$$r_z = z_{P_1} - z_{P_0}$$

Trovato il vettore che permette di descrivere la direzione dell'armature, bisogna verificare il parallelismo con il versore dell'asse dell'elemento (asse x) definito per componenti come:

$$\vec{v}_x = (1,0,0)$$

Sfruttando la proprietà del prodotto vettoriale, due vettori sono paralleli se le componenti del vettore ortogonale $\vec{o}$ sono nulle. Calcolando il prodotto vettoriale come:

$$\vec{o} = \vec{v}_x \times \vec{r} = det \begin{bmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ 1 & 0 & 0 \\ r_x & r_y & r_z \end{bmatrix}$$

Dovranno essere verificate simultaneamente le seguenti condizioni:

$$|o_x| < t$$

$$|o_y| < t$$

$$|o_z| < t$$

Dove t rappresenta la tolleranza che tiene conto delle possibili imprecisioni nella modellazione. Se le componenti del vettore ortogonale sono nulle, l'armatura descritta dai punti P_0 e P_1 è di tipo longitudinale.

In seconda analisi viene descritto il flusso di calcolo per valutare se un'armatura descritta da 3 o più punti sia di tipo longitudinale. Definiti i punti P_0, P_1 e P_2, vengono calcolati i vettori $\vec{r}$ tra P_1 e P_0 e $\vec{s}$ tra P_1 e P_2 eseguendo le stesse operazioni matematiche viste per il caso precedente. Calcolando il loro prodotto vettoriale è possibile determinare le componenti del vettore $\vec{o}$ ortogonale al piano individuato dai vettori $\vec{r}$ e $\vec{s}$ (piano α).

Se le componenti del vettore ortogonale dovessero risultare nulle l'algoritmo prevede la prosecuzione dell'analisi passando al punto successivo ricalcolando quindi, le componenti di $\vec{o}$ con i punti P_0, P_1 e P_n. Nel caso in cui, dopo aver eseguito i calcoli per tutti i punti, il prodotto vettoriale restituisse ancora valori nulli, significa che tutti i punti che descrivono quell'armatura sono allineati. Il problema viene quindi trattato in maniera analoga a quanto visto in precedenza per il caso di armatura descritta da soli due punti.

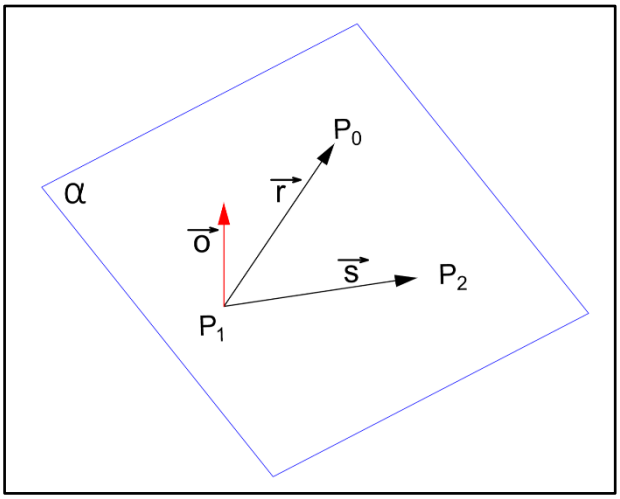


Figura 58 - Individuazione del piano contenente 3 punti

Nel caso in cui le componenti del vettore ortogonale siano diverse da zero, viene calcolata l'equazione del piano α imponendo la condizione aggiuntiva che il piano passi per uno dei punti precedentemente individuati; data quindi l'equazione generica di un piano $ax + by + cz + d = 0$ e il vettore ortogonale $\vec{o} = (o_x, o_y, o_z)$, valgono le seguenti uguaglianze:

$$a = o_x$$

$$b = o_y$$

$$c = o_z$$

Calcolando l'equazione del piano rispetto al punto P_0 si avrà:

$$d = -a \cdot x_{P_0} - b \cdot y_{P_0} - c \cdot z_{P_0}$$

Nota quindi l'equazione del piano passante per 3 punti, per determinare se l'armatura sia di tipo longitudinale devono essere verificate le seguenti condizioni:

- tutti i punti che descrivono l'armatura devo essere contenuti nel piano individuato;
- un qualsiasi vettore appartenente al piano α dev'essere parallelo all'asse d'estrusione dell'elemento.

Verificare che tutti i punti appartengano al piano significa che, per ogni punto P_i, dev'essere verificata l'uguaglianza:

$$a \cdot x_{P_i} + b \cdot y_{P_i} + c \cdot z_{P_i} + d = 0$$

Verificare che un qualsiasi vettore appartenente al piano sia parallelo alla direzione d'estrusione $\overrightarrow{v_x} = (1,0,0)$, significa verificare che le componenti del vettore $\vec{o}$, calcolato

mediante il prodotto vettoriale, siano nulle. Avendo già calcolato i vettori $\vec{r}$ e $\vec{s}$ sarà sufficiente sceglierne uno ed eseguire il prodotto vettoriale con il versore $\overrightarrow{v_x}$ verificando che le componenti del vettore risultante siano nulle. Soddisfatta anche quest'ultima condizione è possibile classificare l'armatura come longitudinale.

2.3.3.3 Armatura non definita

In alcuni casi è possibile che certe armature non soddisfino le condizioni né di elemento trasversale, né longitudinale. Per queste particolari armature è stata definita un'ulteriore tipologia: agli elementi non classificabili verrà infatti assegnato il tipo "non definito".

Un altro caso limite è rappresentato dalla doppia assegnazione ad uno stesso elemento d'armatura della tipologia traversale e longitudinale. Come forma di controllo del codice, vengono processate entrambe le funzioni e se alla fine del calcolo risulta che una stessa barra ha soddisfatto entrambe le condizioni, il programma notifica il problema all'utente classificando l'armatura come tipo "non definito".

2.3.4 Restituzione dei risultati

Terminate le funzioni di calcolo avviene l'analisi della restituzione dei risultati. L'obiettivo è quello di trovare lo strumento più efficace che non solo permetta di rendere visibili i dati calcolati, ma che consenta anche in modo efficace di rintracciare eventuali errori. I metodi di rappresentazione individuati sono:

- la creazione automatica di un **report**;
- la rappresentazione grafica sotto forma di "***albero***" all'interno del software.

2.3.4.1 Creazione del Report

La finalità di questa ulteriore funzione è quella di creare un report automatico capace di esporre in modo ordinato i dati letti e calcolati nel *model object*. L'algoritmo ha quindi il compito di leggere gli attributi desiderati e di scriverli secondo una certa logica all'interno di un nuovo documento che, al termine del calcolo, verrà esportato e messo a disposizione dell'utente. Il formato scelto per questa esportazione è l'***HTML*** (HyperText Markup Language).

L'HTML rappresenta un tipo di linguaggio in grado di gestire i meccanismi di rappresentazione grafica di un testo attraverso l'uso di apposite convenzioni definite dal *World Wide Web Consortium* (W3C). La gestione della visualizzazione grafica avviene tramite l'utilizzo di *tag* standardizzati. L'elasticità del formato ha permesso un notevole sviluppo soprattutto come linguaggio per la gestione di pagine web. Data la grande diffusione, un documento *HTML* ha il vantaggio di poter essere aperto e interpretato dai numerosi dispositivi che ne condividono gli standard. È infatti possibile, attraverso qualsiasi browser web, aprire il file e riprodurne l'aspetto grafico adattatolo al dispositivo utilizzato. Analizzando un file *HTML* attraverso un lettore di testo è possibile leggere il codice ed individuare tra i segni minore "<" e maggiore ">" i *tag* che gestiscono l'impaginazione. Un altro vantaggio del formato è dovuto alla diffusione di programmi che permettono un'ulteriore esportazione nel formato PDF oppure in formati di testo editabili (ad esempio ".docx" per Microsoft Word).

La struttura del report per la riproduzione dei dati calcolati prevede, per ogni elemento trave/pilastro, una descrizione iniziale delle caratteristiche dell'elemento stesso e una tabella dove vengono riportate le armature contenute e classificate in base al tipo. Per come è stato concepito, il report dev'essere quindi un documento di natura dinamica che possa ripetere la rappresentazione di determinate informazioni a seconda del numero di elementi presenti nel progetto. Nell'ambiente di programmazione di Visual Studio, in base al contenuto da descrivere, devono essere ripetute le stringhe di codice inserendo gli opportuni dati letti nel *model object*.

In supporto ad una preliminare analisi del formato, è stato utilizzato il software "*KompoZer*". Questo applicativo, in prima analisi, può apparire come un normale strumento per l'editing di testi con il supporto di diversi stili di formattazione e la possibilità di inserire tabelle e immagini. In realtà il software è in grado di tradurre in formato *HTML* le informazioni sia testuali sia di formattazione inserite nell'ambiente di editing.

In seguito viene riportato il report creato e formattato con il programma *KompoZer*.

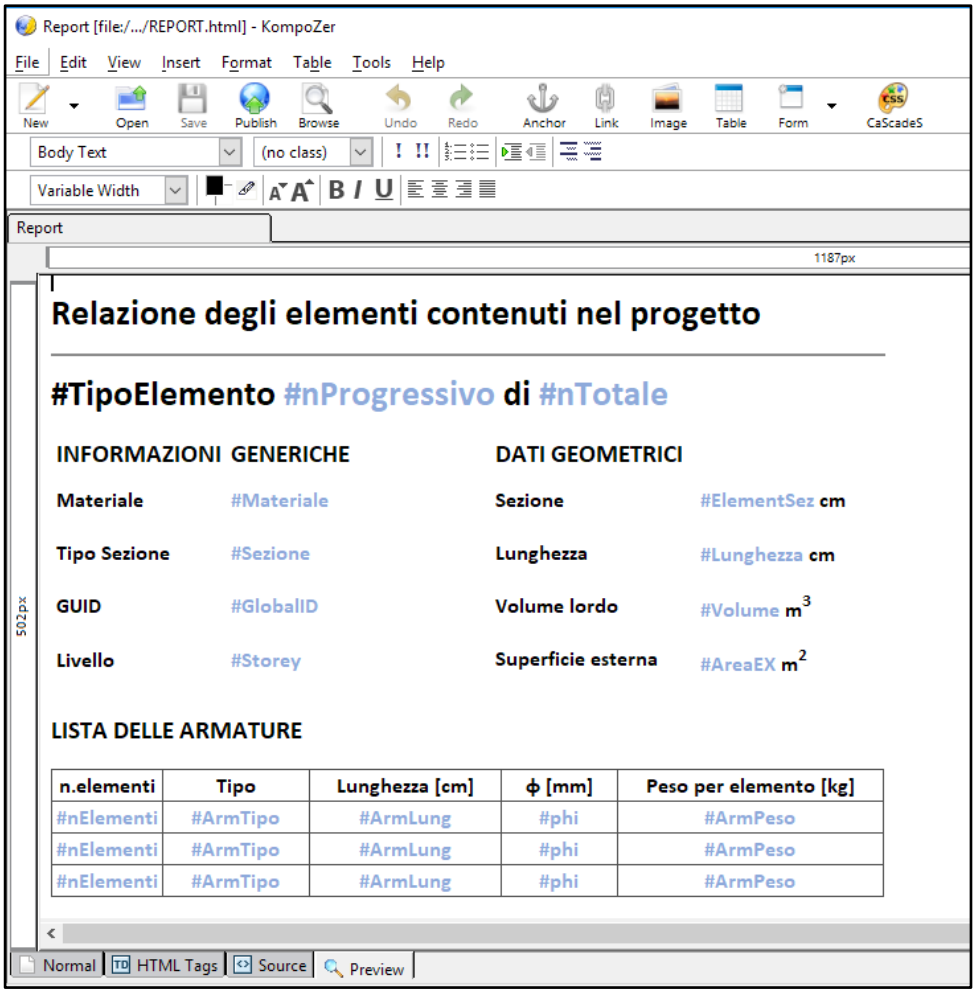


Figura 59 - Definizione del report con l'utilizzo del programma KompoZer

Analizzando la traduzione in *HTML*, è indispensabile studiare la struttura per capire quali stringhe di codice dovranno ripetersi in modo da poter descrivere tutti gli elementi del progetto. In prima analisi, un documento *HTML* si divide in due principali blocchi di codice:

- *Header*: è la porzione di codice compresa tra i *tag* "*<head>*" e "*</head>*", rappresenta l'intestazione del documento e contiene informazioni di controllo che non hanno un diretto riscontro grafico;
- Body: è la porzione di codice compresa tra i *tag* "*<body>*" e "*</body>*", contiene la parte informativa con i dati di testo, le immagini e le informazioni che permettono la riproduzione grafica.

L'individuazione delle stringhe da ripetere avviene quindi all'interno della sezione "body". L'analisi del codice *HTML* del report precedentemente creato, avviene in un'apposita finestra del software. Attivando la finestra "Source" è possibile visionare e modificare direttamente il codice; le modifiche apportate andranno direttamente ad aggiornare la visualizzazione grafica nella finestra "Preview". Sfruttando questa caratteristica del programma, è stato possibile individuare le parti di codice per creare un report dinamico. Le stringhe estratte sono state salvate in diversi file di testo che verranno richiamati nell'ambiente di programmazione per comporre il file da esportare. Per ogni parte di codice *HTML*, salvata come "txt", verrà seguita la seguente procedura:

- apertura del file txt in Visual Studio;
- lettura e salvataggio del contenuto in una stringa;
- sostituzione dei caratteri speciali con i dati letti nel *model object*:

come è possibile vedere nella figura 60, alcuni nomi sono preceduti dal carattere "#", attraverso questo simbolo sarà possibile rintracciare nel codice la parola desiderata e sostituirla con il relativo valore numerico o letterale letto nel *model object*.

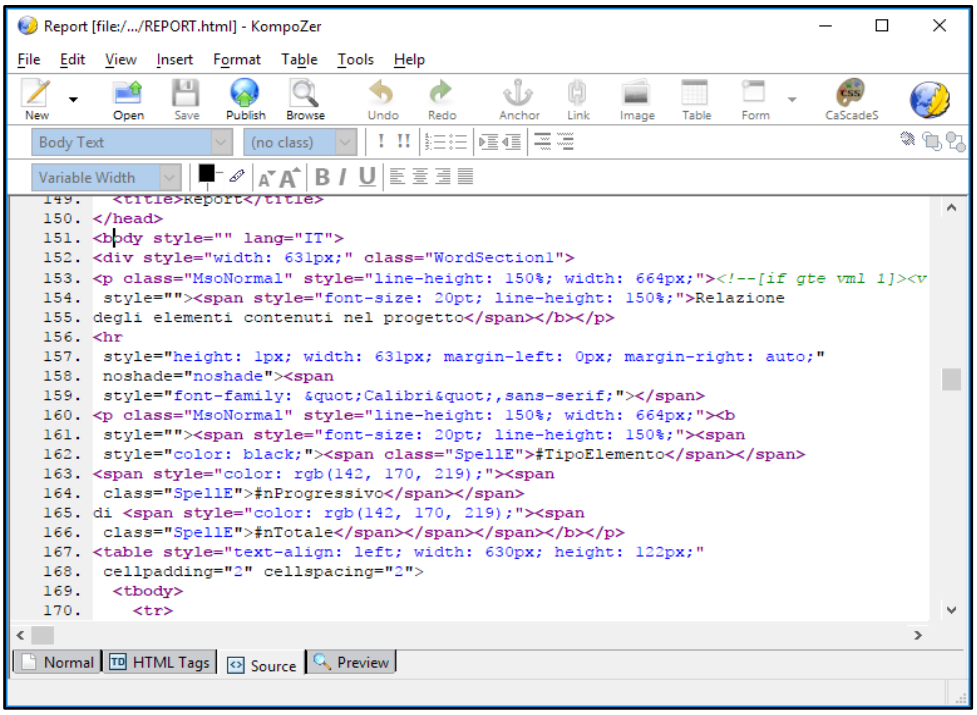


Figura 60 - Visualizzazione del codice HTML

I file di testo creati e richiamati secondo l'opportuno ordine nell'ambiente di programmazione, permettono di ottenere il codice *HTML* da scrivere e salvare in un nuovo file per l'esportazione. Nell'ordine vengono richiamati i seguenti file di testo:

- **Intestazione.txt**: contiene i dati dell'intestazione di un file *HTML* (head) e il titolo del Report;
- **Elemento.txt**: contiene tutte le caratteristiche riferite all'elemento, il titolo e l'intestazione della tabella delle armature, viene richiamato nel codice per ogni elemento pilastro/trave presente nel file IFC;
- **RigaTabella.txt**: come descritto nella seguente immagine, permette di aggiungere una riga alla tabella delle armature, viene richiamato per ogni tipologia di armatura presente nell'elemento analizzato;

```
   <tr>
     <td
style="font-family: Calibri; width: 78px; text-align: center;"><b
style=""><span
style="font-size: 12pt; color: rgb(142, 170, 219);"><span
class="SpellE">#nElementi</span></span></b></td>
     <td
style="font-family: Calibri; text-align: center; width: 105px;"><b
style=""><span
style="font-size: 12pt; color: rgb(142, 170, 219);">#ArmTipo</span></b></td>
     <td
style="font-family: Calibri; text-align: center; width: 138px;"><b
style=""><span
style="font-size: 12pt; color: rgb(142, 170, 219);">#ArmLung</span></b></td>
     <td
style="font-family: Calibri; text-align: center; width: 82px;"><b
style=""><span
style="font-size: 12pt; color: rgb(142, 170, 219);"><span
class="SpellE">#phi</span></span></b></td>
     <td
style="font-family: Calibri; text-align: center; width: 196px;"><b
style=""><span
style="font-size: 12pt; color: rgb(142, 170, 219);">#ArmPeso</span></b></td>
   </tr>
```

Figura 61 - Estratto di codice che rappresenta una riga della tabella

- **FineTabella.txt**: contiene la parte di codice per terminare una tabella;
- **Fine.txt**: terminato il ciclo sui pilastri, sulle travi e sulle relative armature, viene completato il file *HTML* con le ultime istruzioni necessarie.

Ogni frammento di codice viene concatenato in una stringa in modo che vi sia un'unica variabile contenente tutto il testo del report. Al termine della procedura, viene creato un nuovo file *HTML* in cui viene scritto tutto il contenuto.

2.3.4.2 Rappresentazione grafica

I risultati delle funzioni analizzate in precedenze sono alla base della ricostruzione gerarchica dell'albero delle dipendenze nel software che effettuerà il *code checking*. All'interno del programma, dopo aver selezionato il file IFC da importare, verranno eseguite le funzioni in modo che vengano calcolati i dati ricercati. L'albero delle dipendenze viene modificato per adattarlo ai fini del *code checking* strutturale. Infatti, da ogni trave o pilastro si potrà accedere direttamente alla lista delle armature contenute e verificare graficamente che siano state correttamente classificate.

Nel programma sarà quindi possibile visualizzare l'albero delle dipendenze contenente tutti gli elementi; selezionata un'armatura, vengono riportate a video sia le caratteristiche lette

nel file IFC sia quelle calcolate con gli algoritmi. Nello spazio dedicato alla rappresentazione grafica è possibile vedere tridimensionalmente l'armatura in modo da effettuare un ulteriore verifica dei dati trovati.

3. Risultati

Vengono riportati i risultati dell'applicazione critica del flusso metodologico al caso studio. Nel dettaglio vengono analizzati i risultati:

- della modellazione e della successiva esportazione IFC;
- dell'applicazione delle funzioni con i relativi output.

3.1 Modellazione ed esportazione IFC

Il lavoro svolto utilizzando l'applicativo Revit Architecture, ha permesso di ottenere un modello tridimensionale in grado di essere correttamente esportato in IFC. Come visto in precedenza, è stata modellata una porzione di edificio (satellite del 7° piano) per definire precise regole da seguire nella modellazione per il *code checking* strutturale. Integrando il modello esistente con le informazioni contenute nelle tavole strutturali, sono state modellate le geometrie e definite le caratteristiche delle travi con le relative armature.

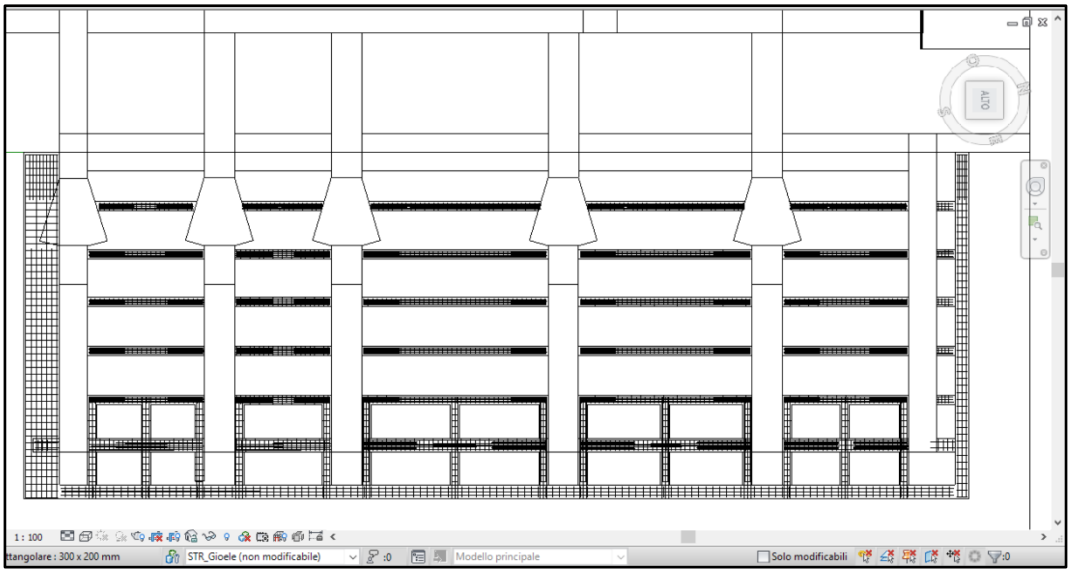


Figura 62 - Vista in pianta del satellite del settimo piano in Revit Architecture, non in scala

Il processo di continue esportazioni ha permesso di definire il modo migliore per ottenere un file con le informazioni desiderate. A differenza della modellazione BIM, in cui si è lavorato su un progetto contenente l'intero edificio, l'esportazione IFC è stata svolta solo sulla porzione utile per essere processata nelle funzioni. Precedenti esportazioni, effettuate

sull'intero modello di un progetto complesso, hanno infatti dimostrato come il processo di scrittura e di successiva lettura del file IFC sia molto lento.

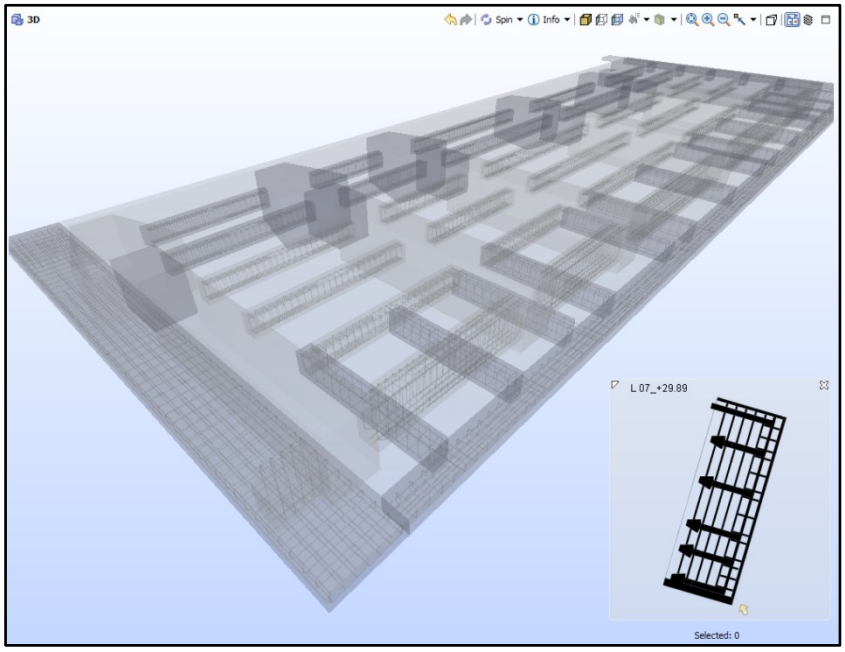


Figura 63 - Visualizzazione del file IFC con Solibri Model Viewer

Applicando opportuni filtri di visibilità sono stati esportanti soltanto gli elementi strutturali modellati con le relative armature. Il file è stato quindi processato dalle funzioni implementate.

3.2 Output

3.2.1 Report

Dopo aver applicato secondo l'ordine opportuno le funzioni analizzate, il programma è in grado di restituire il report in formato *HTML* che descrive, per ogni file IFC, gli elementi trave/pilastro con le relative armature. Oltre ai dati letti, attraverso alcuni semplici calcoli si è in grado di fornire informazioni aggiuntive utili a diversi scopi. In particolare, per ogni elemento, vengono definite le principali caratteristiche geometriche e viene calcolato il peso relativo al singolo elemento di armatura. Noto il volume, la superficie esterna, il peso delle armature e le dimensioni di un elemento in calcestruzzo, è possibile esportare il report *HTML* in un foglio di calcolo e aver un'indicazione di massima sul costo di quel singolo elemento. A seconda delle esigenze è infatti possibile personalizzare la raccolta di dati da inserire nel report.

Trave 38 di 69

INFORMAZIONI GENERICHE		DATI GEOMETRICI	
Materiale	Calcestruzzo - Calcestruzzo gettato in opera	Sezione	30x20 cm
Tipo Sezione	Rettangolare	Lunghezza	250 cm
GUID	3J7HRNpqX6e9tAOUcFPRqY	Volume lordo	0,15 m^3
Livello	L 07_+29.89	Superficie esterna	2,62 m^2

LISTA DELLE ARMATURE

n.elementi	Tipo	Lunghezza [cm]	ϕ [mm]	Peso per elemento [kg]
25	Trasversale	84,6	8	0,33
5	Longitudinale	184,5	14	2,22
5	Longitudinale	60,8	14	0,73
8	Longitudinale	77,5	12	0,68
2	Longitudinale	245,3	12	2,16
6	Longitudinale	97,5	20	2,39

Figura 64 - Estratto dal report dei risultati

Il report *HTML* contenente il risultato delle analisi è stato salvato in PDF e viene messo a disposizione attraverso il seguente QR Code.

Figura 65 - Report Satelliti, attivo a ottobre 2017

Questo strumento ha messo in luce alcuni limiti delle funzioni analizzate. Nel report è possibile notare una trave la cui geometria non è stata riconosciuta. Studiando le ragioni che hanno portato a questo risultato, è emerso che la trave viene descritta all'interno del formato IFC come un solido dato dalla composizione di facce e non come un'estrusione di un rettangolo.

Trave 1 di 69			
INFORMAZIONI GENERICHE		**DATI GEOMETRICI**	
Materiale	Calcestruzzo - Calcestruzzo gettato in opera	**Sezione**	Non definita cm
Tipo Sezione	Non assegnato	**Lunghezza**	Non definita cm
GUID	3QixwEaSbAwx9oG00WNkDs	**Volume lordo**	Non calcolabile m^3
Livello	L 07_+29.89	**Superficie esterna**	Non calcolabile m^2

Figura 66 - Trave la cui sezione non è stata riconosciuta

Analizzando la trave per via grafica è facilmente intuibile come non sia possibile descriverne la forma attraverso una semplice estrusione.

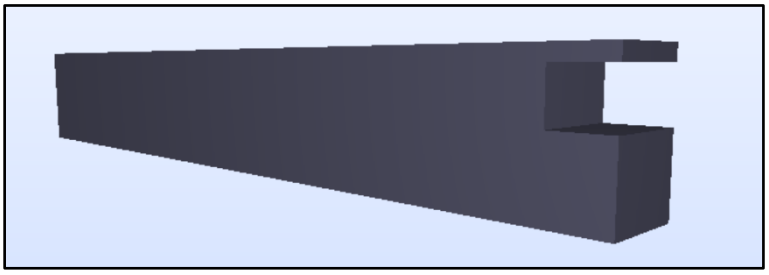


Figura 67 - Trave non descritta attraverso l'estrusione

La trave assume questa particolare forma perché il volume è generato dalla **sottrazione solida** di due travi che si compenetrano. Per risolvere questo problema è possibile modificare il modello dal quale è stato generato il file IFC; ove non fosse possibile risalire al file sorgente, le funzioni dovranno essere implementate in modo che possano restituire un risultato valido anche in presenza di elementi non estrusi.

3.2.2 Rappresentazione grafica nel software del code checking

I processi di calcolo permettono, all'interno dell'applicativo che svolgerà il *code checking*, di raggruppare e riorganizzare i dati in maniera ottimizzata per effettuare le verifiche di corrispondenza con la normativa. Come è possibile osservare dalla seguente immagine, l'**albero delle gerarchie** viene modificato mettendo in relazione gli elementi in calcestruzzo con le relative armature. Conseguentemente alla selezione di un elemento, vengono visualizzate sia le principali caratteristiche (tra cui la classificazione del tipo), sia la collocazione nell'ambiente grafico per un immediato riscontro tridimensionale.

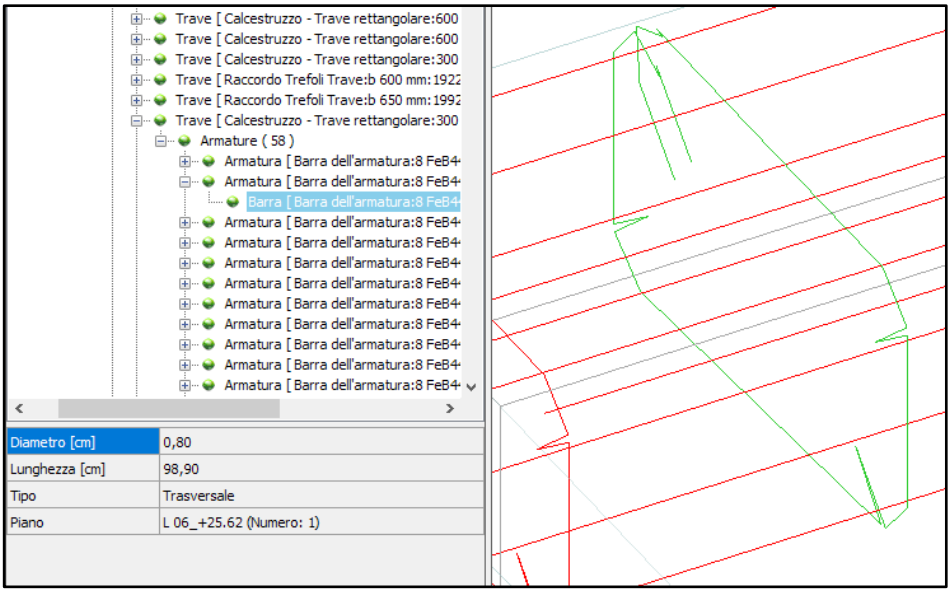


Figura 68 - Interazione tra gli algoritmi di calcolo e il programma del code checking

4. Conclusioni

L'approccio alla progettazione tramite la metodologia BIM rappresenta il futuro del mondo della progettazione. In un sistema in cui assume sempre maggiore importanza lo scambio di dati, diventa indispensabile lo sviluppo di formati file aperti e condivisi da più piattaforme. Le diverse analisi hanno permesso di individuare le **potenzialità** dell'utilizzo del formato IFC ma, allo stesso tempo, ne sottolineano i **limiti** mettendo in luce alcune difficoltà metodologiche. Il lavoro è stato svolto coniugando le competenze apprese in ambito accademico a quelle informatiche sviluppate grazie ad un percorso di collaborazione aziendale. La tesi può essere uno strumento per coloro che, confrontandosi con il formato IFC, intendono analizzare a fondo non solo la rappresentazione grafica, ma anche il contenuto del codice.

4.1 Criticità, limiti e future implementazioni delle funzioni

Come già emerso dai risultati, le funzioni di calcolo hanno diversi limiti di applicabilità. Il principale problema è che non esiste un modo univoco per la descrizione delle forme e delle

caratteristiche di un medesimo elemento. All'interno dello stesso software, oppure, rappresentando lo stesso elemento su diverse piattaforme, è possibile osservare come possa cambiare il metodo di rappresentazione a seconda dell'esportazione. Implementando sempre nuove entità in grado di gestire più casi possibili, vi è il rischio di creare ambiguità nella scelta dell'entità IFC da utilizzare. Dal lato dello sviluppo software vi è quindi la necessità di uno sforzo maggiore per cercare di coprire le diverse casistiche. I principali limiti delle precedenti funzioni sono:

- analisi limitata agli elementi in calcestruzzo descritti per estrusione;
- tra le possibili forme della sezione d'estrusione solo quella rettangolare è stata implementata (le sezioni a T e circolari sono solo state analizzate).

Un'attenta analisi ha dimostrato come, tra le possibili implementazioni future, risulti relativamente semplice estendere le funzioni di calcolo agli elementi aventi sezione a T. L'algoritmo può infatti essere riutilizzato dividendo la sezione nei due rettangoli che la compongono. Con poche modifiche, quindi, sarà possibile ottenere validi risultati applicando in successione le precedenti funzioni.

Differente risulta essere il caso della sezione circolare. Analizzando, ad esempio, un pilastro circolare, le funzioni andrebbero riviste utilizzando come parametro in ingresso il raggio della sezione e la posizione del centro d'estrusione.

Di maggiore difficoltà è la gestione di elementi definiti attraverso profili arbitrari o non rappresentati come estrusioni: gli algoritmi di calcolo, basandosi su analisi effettuate su sezioni trasversali costanti, non possono essere riutilizzati su elementi rappresentati con operazioni booleane tra solidi o attraverso una composizione di facce.

4.2 Il formato IFC, sviluppi futuri

Le potenzialità che offre il formato IFC sono ormai note e condivise dai professionisti che operano nel settore secondo una metodologia BIM. Le analisi hanno permesso di comprendere a fondo la capacità del formato di descrivere con precisione la quasi totalità delle situazioni progettuali in ambito edilizio.

Un limite del formato, dovuto principalmente a come è stato concepito, è dato dalla perdita della parametrizzazione dell'oggetto una volta esportato. Se in un ambiente di modellazione un oggetto viene descritto mediante la valorizzazione di determinati parametri (rappresentazione dinamica), l'esportazione IFC riporterà soltanto una propria descrizione dimenticando le relazioni parametriche che l'hanno creato (rappresentazione statica). Risulta molto semplice osservare quanto precedentemente dichiarato, esportando un oggetto e importandolo nuovamente nel medesimo ambiente di modellazione. Il motivo alla base della perdita di parametrizzazione è dato dal diverso modo di rappresentazione utilizzato dai software BIM. Volendo creare una piattaforma comune e aperta, risulta impossibile avere un'unica rappresentazione dinamica condivisa e riconosciuta dai diversi

applicativi; per questo motivo è stato deciso di avere una rappresentazione neutra a discapito della perdita di parametrizzazione.

In un periodo di forte sviluppo tecnologico, anche chi si occupa del mantenimento del formato si è posto degli obiettivi ben precisi sugli sviluppi futuri. *BuildingSMART* è costantemente al lavoro per cercare di mantenere aggiornato il sistema a seconda delle richieste del mercato e degli sviluppi tecnologici. In particolare, partendo dall'esperienza accumulata per la descrizione dei manufatti edilizi, l'obiettivo dello sviluppo del formato IFC5 è quello di migliorare le attuali funzionalità e di estenderle al campo infrastrutturale.

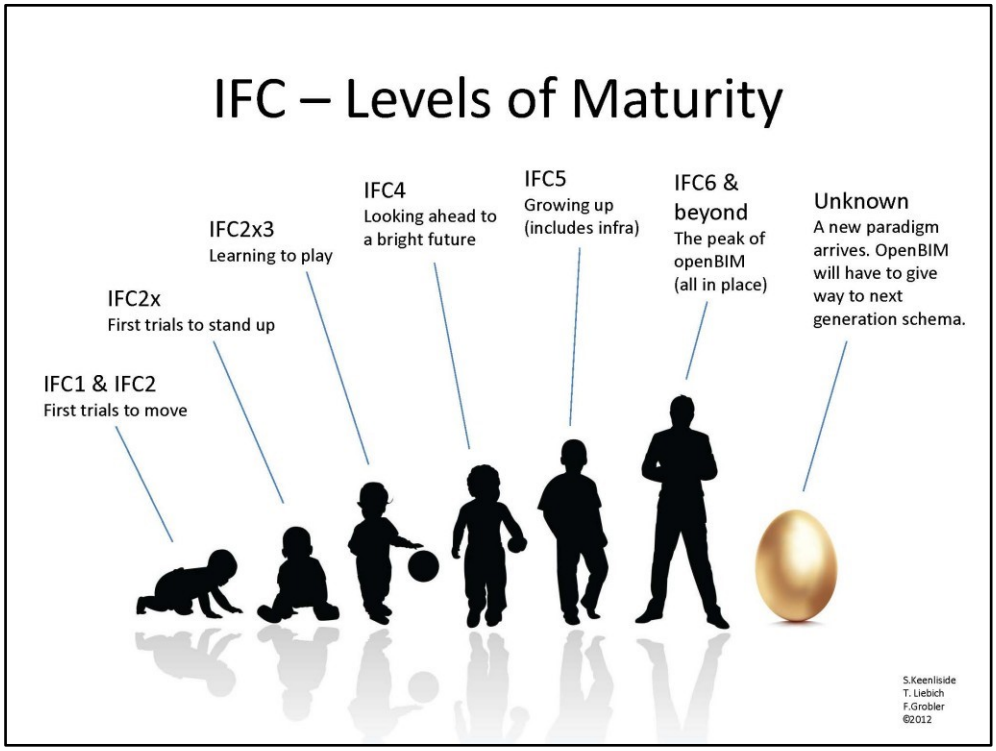


Figura 69 - Livello di maturità del formato IFC, www.buildingsmartcanada.ca

4.3 Possibili scenari e sviluppi della progettazione BIM

Lo sviluppo tecnologico e l'aggiornamento normativo dovrebbero rendere sempre più comune e diffuso l'approccio alla metodologia BIM. Seppur in ritardo rispetto a quanto fatto in altri paesi, la graduale introduzione del BIM nel mercato italiano ha dato inizio a una rivoluzione nel mondo della progettazione. Molti professionisti, per rimanere competitivi nel

mercato, dovranno cambiare la propria prassi di lavoro e seguire con costante aggiornamento l'intero processo evolutivo.

Si può dimostrare come il rendimento di un sistema innovativo, nonostante in un primo momento risulti inferiore rispetto a quello di un sistema tradizionale, nel tempo porti ad un notevole aumento dell'efficienza e alla progressiva obsolescenza dei sistemi tradizionali.

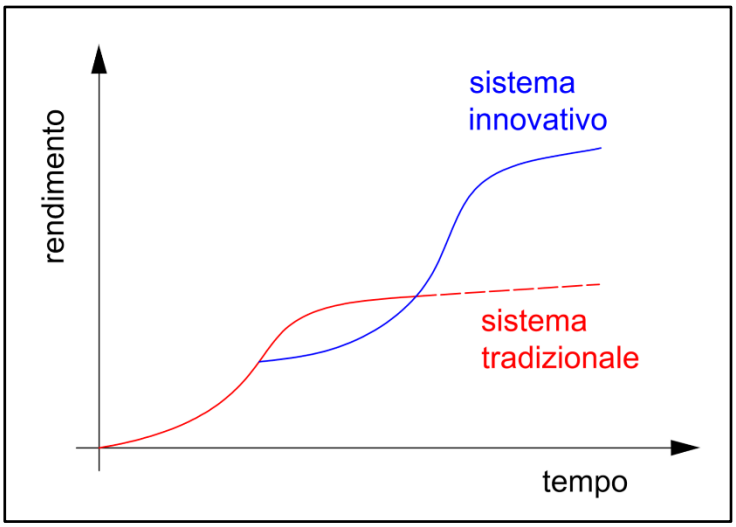


Figura 70 - Grafico del rendimento nel tempo

Il lavoro svolto ha permesso di compiere un percorso di conoscenza riguardo a tematiche con cui avevo già avuto modo di confrontarmi in precedenza. Dopo aver studiato nel dettaglio il formato IFC applicato in un sistema BIM, posso continuare ad affermare quanto sia importante proseguire sullo **sviluppo** e sulla **sperimentazione** di queste tecnologie nell'ottica di poter lavorare con un sistema performante e allo stesso tempo interoperabile.

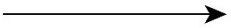
5. ALLEGATI

5.1 Specifiche sulla rappresentazione dei diagrammi di flusso

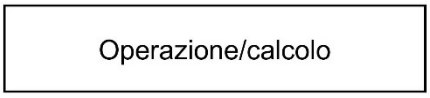
Nella rappresentazione dei diagrammi di flusso vengono utilizzate le seguenti convenzioni:

- Inizio e fine di un flusso di operazioni:

- Direzione del flusso:

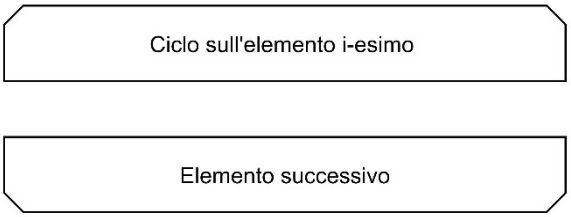


- rappresentazione di un'operazione:



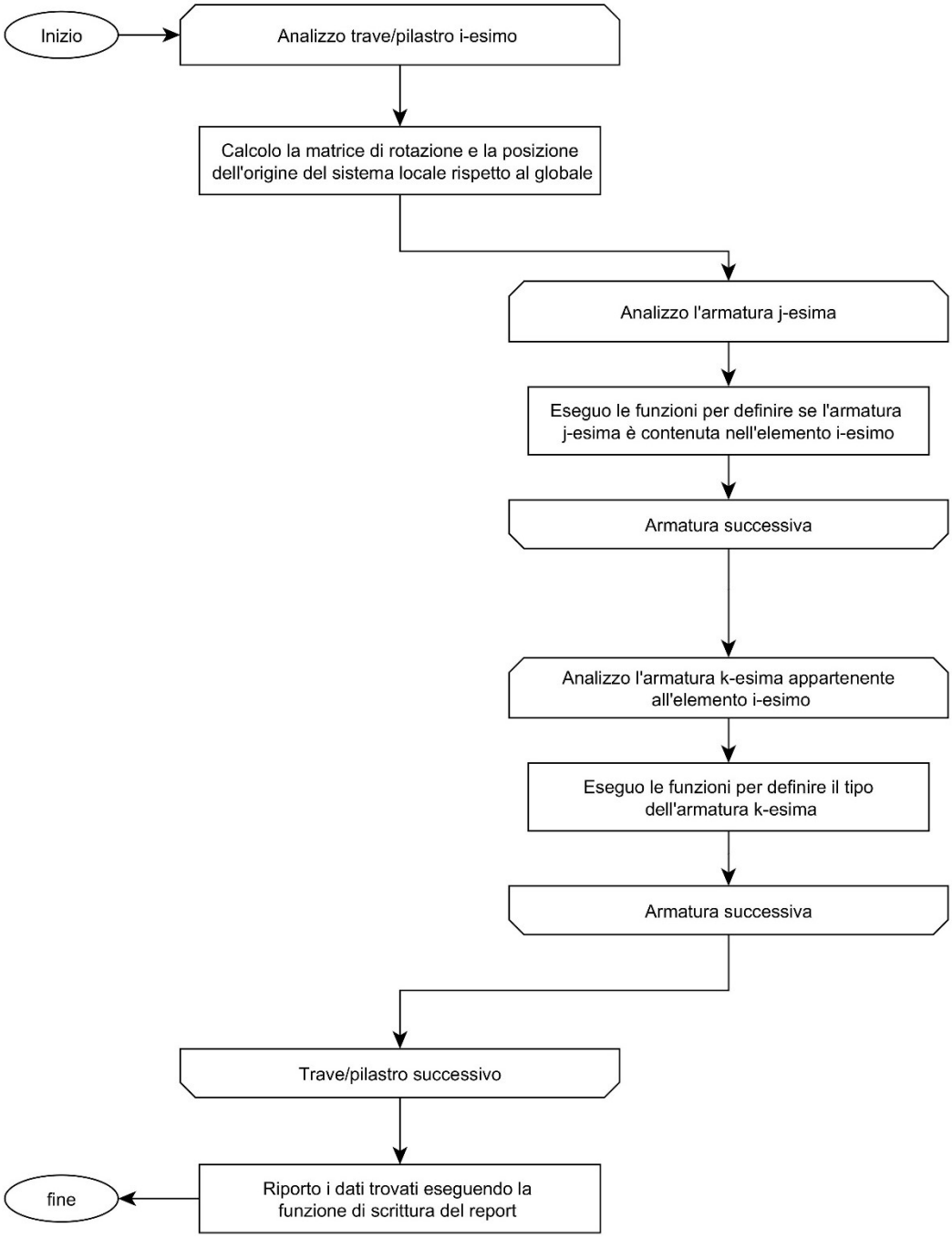
- risultato booleano di un'operazione Sì/No:

- eseguo un ciclo sulle funzioni contenute all'interno:



5.2 Flusso dell'applicazione delle funzioni

5.2.1 Diagramma di flusso



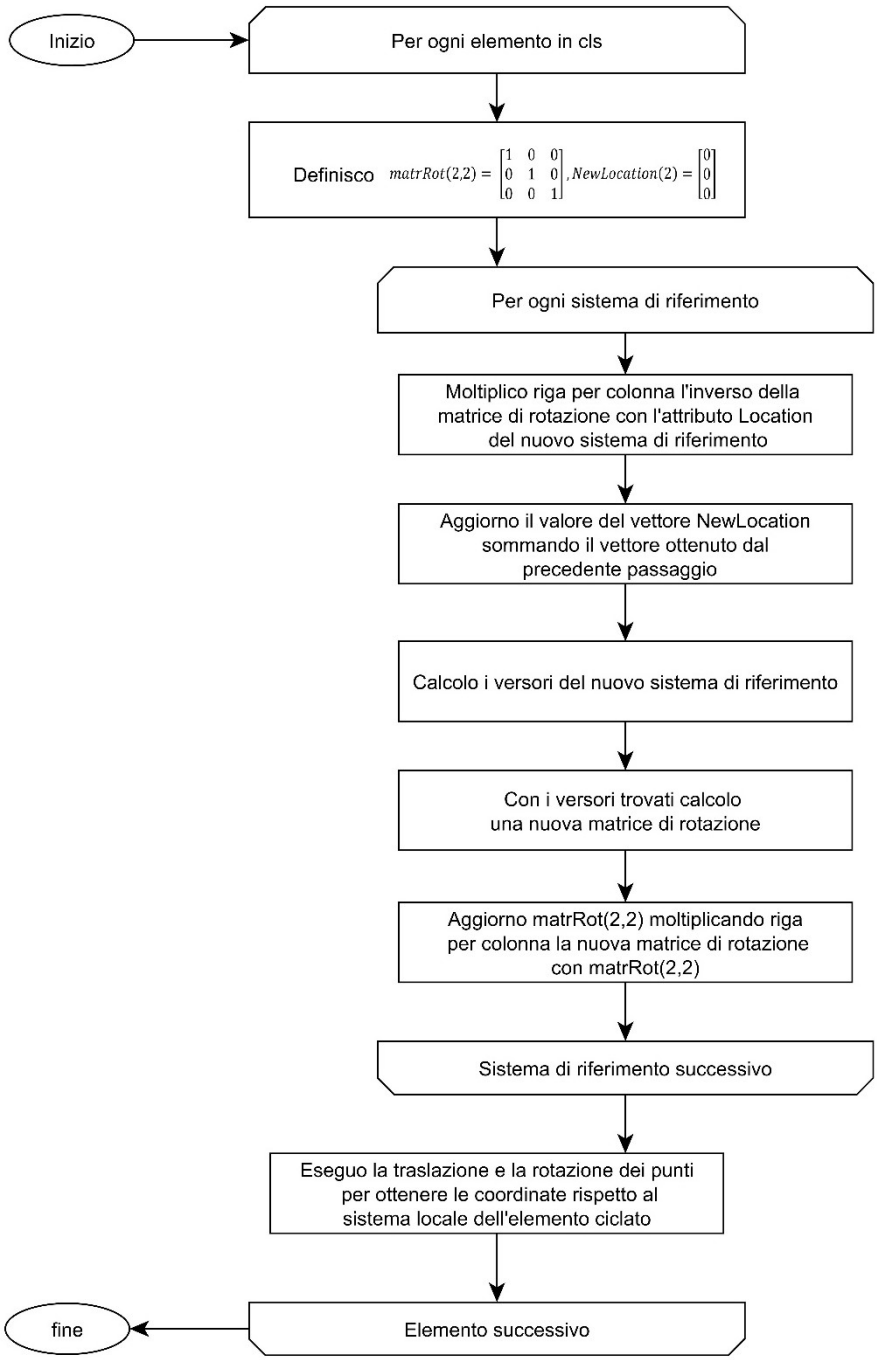
5.2.2 Codice vb.NET

Viene di seguito riportato il codice che permette di applicare le funzioni agli elementi presenti nel file IFC. Nell'esempio prevede i calcoli effettuati per l'entità *IfcBeam*.

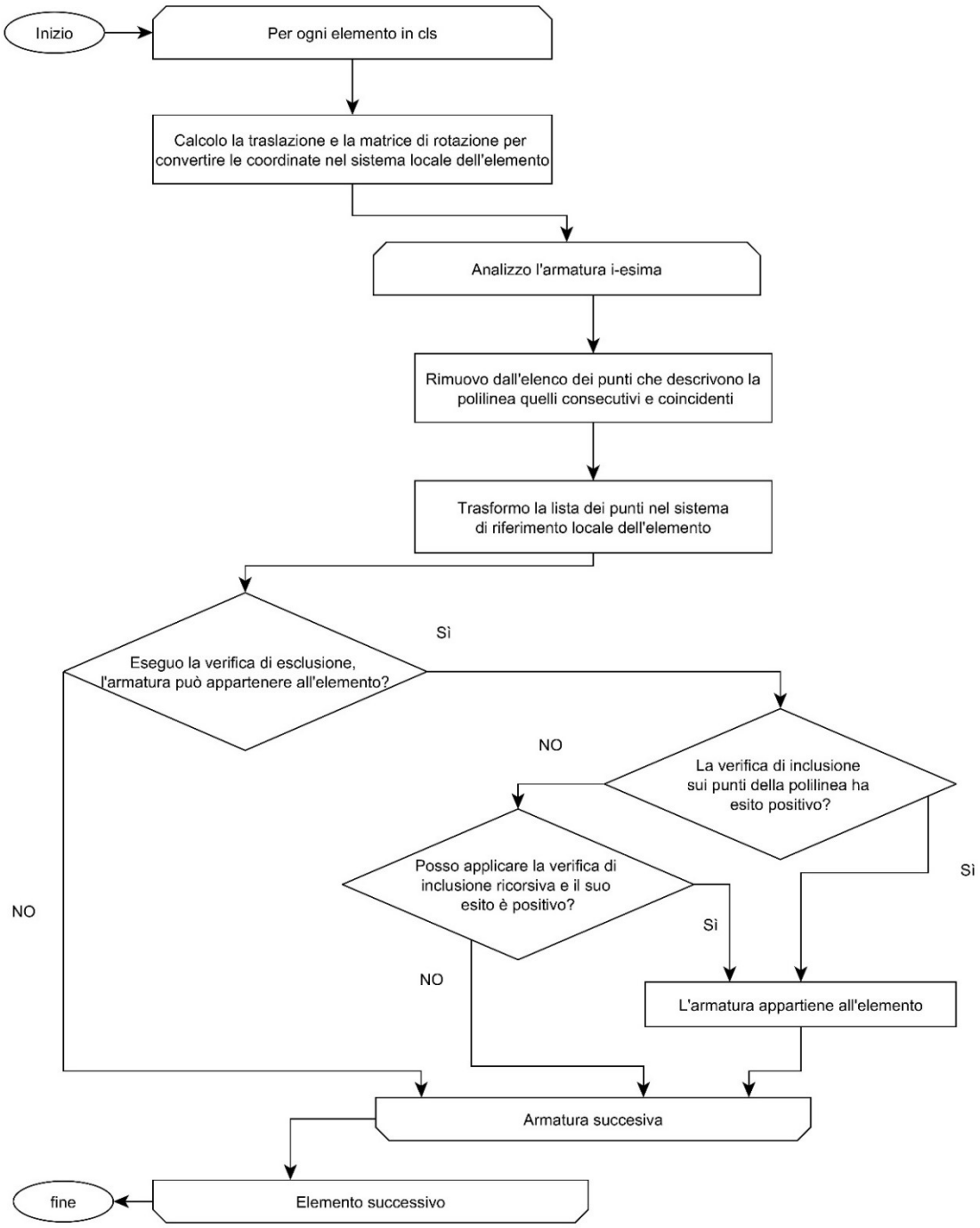
```
Public Sub BeamTipoArmatura(ByVal mybeam As IFC_ObjectModel.Cls_IFC_Beam,
ByVal myLstRBitem As List(Of IFC_ObjectModel.Cls_IFC_RebarItem))

        Try

            If mybeam.Section Is Nothing Then Exit Sub
            If mybeam.StartPoint Is Nothing Then Exit Sub
            If mybeam.EndPoint Is Nothing Then Exit Sub

            'dichiarazioni
            Dim b1 As Double = 0
            Dim h1 As Double = 0
            Dim Sez As IFC_ObjectModel.Cls_IFC_Section = mybeam.Section
            Dim LstRBItemInElement As New List(Of Cls_IFC_RebarItem)
            Dim matrRot(2, 2) As Double
            Dim Location(2) As Double
            Dim matrRotBH(2, 2) As Double
            Dim LocationBH(2) As Double
            Dim lastPlacement As New Cls_IFC_Placement
            Dim pstart(2) As Double
            Dim pend(2) As Double
            Dim myPlacement As New Cls_IFC_Placement
            Dim myLstPlacement As New List(Of Cls_IFC_Placement)
            Dim PlacementN As New Cls_IFC_Placement


            'salvo le dimensioni della sezione
            b1 = CType(Sez, IFC_ObjectModel.Cls_IFC_Section_Rect).b1
            h1 = CType(Sez, IFC_ObjectModel.Cls_IFC_Section_Rect).h1

            lastPlacement.Axis.X = 0
            lastPlacement.Axis.Y = 1
            lastPlacement.Axis.Z = 0

            lastPlacement.RefDirection.X = 0
            lastPlacement.RefDirection.Y = 0
            lastPlacement.RefDirection.Z = 1

            lastPlacement.Location.X = 0
            lastPlacement.Location.Y = 0
            lastPlacement.Location.Z = 0

            For i = 0 To 2
                Location(i) = 0
                For j = 0 To 2
                    If i = j Then
                        matrRot(i, j) = 1
                    Else
                        matrRot(i, j) = 0
                    End If
                Next
            Next
```

```
        myPlacement = mybeam.Placement
        myLstPlacement.Add(myPlacement)

        While Not myPlacement.ParentPlacement Is Nothing
            myPlacement = myPlacement.ParentPlacement

            PlacementN = New Cls_IFC_Placement
            PlacementN = myPlacement

            myLstPlacement.Add(PlacementN)
        End While

        For i = myLstPlacement.Count - 1 To 0 Step -1
            FC.CalcolaMatriceRot(myLstPlacement.Item(i), matrRot, Location)
        Next

        myLstPlacement = New List(Of Cls_IFC_Placement)


        If Not mybeam.Placement1 Is Nothing Then
        FC.CalcolaMatriceRot(mybeam.Placement1, matrRot, Location)
        end if
        If Not mybeam.Placement2 Is Nothing Then
        FC.CalcolaMatriceRot(mybeam.Placement2, matrRot, Location)
        end if
        If Not mybeam.Placement3 Is Nothing Then
        FC.CalcolaMatriceRot(mybeam.Placement3, matrRot, Location)
        end if
        If Not mybeam.Placement4 Is Nothing Then
        FC.CalcolaMatriceRot(mybeam.Placement4, matrRot, Location)
        end if

        'impongo l'asse x come asse di estrusione
        FC.CalcolaMatriceRot(lastPlacement, matrRot, Location)

        'definisco la relazione tra barre e trave
        LstRBItemInElement = New List(Of Cls_IFC_RebarItem)

        pstart(0) = mybeam.StartPoint.X
        pstart(1) = mybeam.StartPoint.Y
        pstart(2) = mybeam.StartPoint.Z

        pend(0) = mybeam.EndPoint.X
        pend(1) = mybeam.EndPoint.Y
        pend(2) = mybeam.EndPoint.Z

        FC.ConvertPoint(matrRot, Location, pstart)
        FC.ConvertPoint(matrRot, Location, pend)


        'la funzione restituisce la lista degli item appartenenti all'elemento
        LstRBItemInElement.AddRange(DefRelazione(myLstRBitem, matrRot,
        Location, b1, h1, mybeam.EndPoint))

        'funzione che definisce la differenza tra staffa e barra longitudinale
        DefReinforcingType(LstRBItemInElement, matrRot, Location,
        b1, h1, mybeam.EndPoint)

        'inserisco la relazione all'interno del model
        For Each item In LstRBItemInElement
            item.AddParent(mybeam.Global_ID.ToString,
            enumIFCElement.ifcBeam, Cls_IFC_ParentType.enum_TypeFlag.Geometry)
        Next

    Catch ex As Exception
        MsgBox(ex.ToString, MsgBoxStyle.Critical)
    End Try

End Sub
```

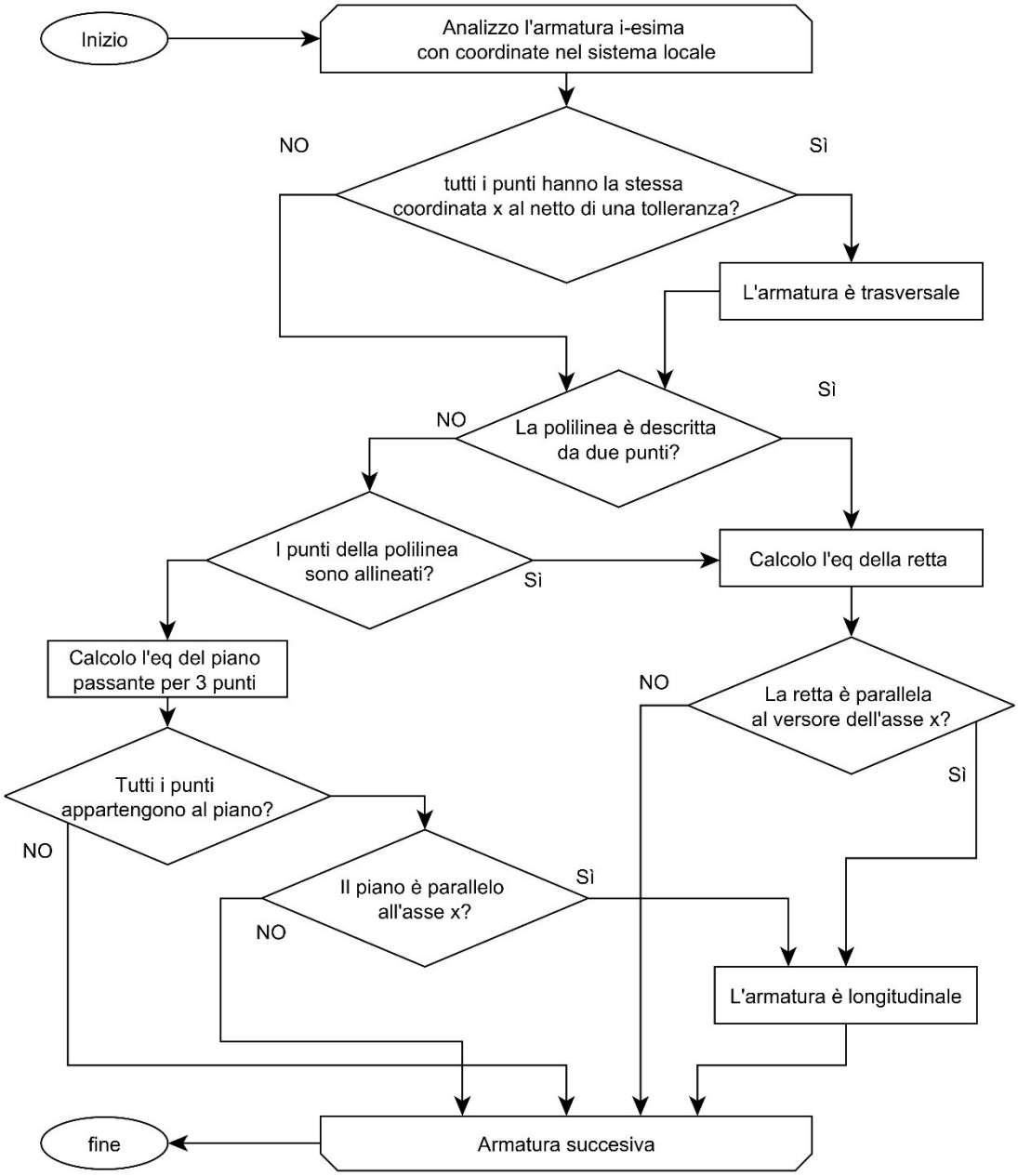
5.3 Trasformazioni dei sistemi di riferimento

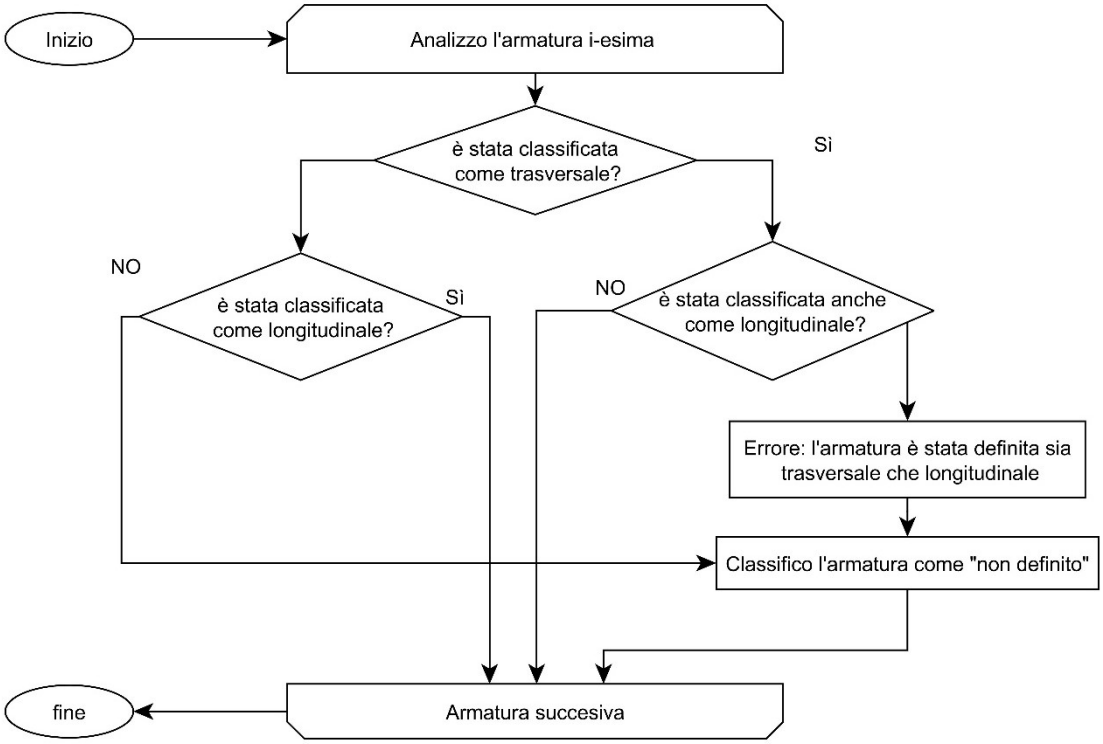


5.4 Associazione elemento-armatura



5.5 Distinzione armatura longitudinale e trasversale




Inizio
Analizzo l'armatura i-esima
è stata classificata come trasversale?
Sì
NO
è stata classificata come longitudinale?
Sì
NO
è stata classificata anche come longitudinale?
Errore: l'armatura è stata definita sia trasversale che longitudinale
Classifico l'armatura come "non definito"
fine
Armatura succesiva

Software utilizzati

- Modellazione BIM, import/export IFC:
    - Revit 2017 [www.autodesk.it]
    - Tekla Structures [www.tekla.com]
- Visualizzatori e analizzatori IFC:
    - Solibri Model Viewer [www.solibri.com]
    - RDF Ifc Viewer [http://rdf.bg]
    - BIM Vision Ifc Model Viewer [bimvision.eu]
    - IFCQuickBroswer [www.team-solution.de]
    - IFC File Analyzer [www.nist.gov]
- Ambiente di sviluppo software: Visual Studio 2013 [www.visualstudio.com]
- Disegno CAD: Autodesk AutoCAD 2016 [www.autodesk.it]
- Editor testo: Microsoft Word [www.office.com]
- Elaborazione dati per test sulle funzioni: Microsoft Excel [www.office.com]
- Flow Chart: yEd Graph Editor [www.yworks.com]
- Elaborazione file txt: NotePad ++ [notepad-plus-plus.org]
- Creazione e gestione database:
    - Revit DB Link [www.autodesk.it]
    - Access [www.office.com]
- Creazione file HTML per report: KompoZer [kompozer.net]

Bibliografia

- A. Osello, Il futuro del disegno con il BIM per ingegneri e architetti, Dario Flaccovio editore, 2012
- C. Eastman, P. Teicholz, R. Sacks, K. Liston, BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2011
- C. Eastman, P. Teicholz, R. Sacks, K. Liston, Il BIM, guida complete al building information modeling, traduzione di G. M. Di Giuda, V.Villa, editore Hoepli, 2016
- D. Bochiccio, C. Civera, M. De Sanctis, A. Leoncini, M. Leoncini, S. Mostarda, VISUAL BASIC 2015: guida completa per lo sviluppatore, Hoepli editore, 2016
- A.Osello, F. M. Ugliotti, BIM: verso il catasto del futuro, Gangemi editore, 2017
- Regione Piemonte, Un simbolo per la Regione. Presentazione dell'esito del concorso di progettazione del nuovo palazzo della Regione Piemonte, Torino, 2001
- L. Gatto, Lezioni di Algebra lineare e Geometria, Clut editore, 2013

Normativa di riferimento

- Normativa italiana:
  - D.M. 14 gennaio 2008, Norme tecniche per le costruzioni
- Norme UNI:
  - UNI 11337:2017 Gestione digitale di processi informativi delle costruzioni:
    - Parte 1: Modelli, elaborati e oggetti informativi per prodotti e processi.
    - Parte 4: Evoluzione e sviluppo informativo di modelli, elaborati ed oggetti.
    - Parte 5: Flussi informativi nei processi digitalizzati.
    - Parte 6: Linea Guida per la redazione del capitolato informativo.
- Norme ISO:
  - ISO 16739, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries;
  - ISO 29481, process definition (Information Delivery Manual) IDM;
  - ISO 16739, dictionary terms (Industry Classes Foundation) IFC.

Sitografia

- www.buildingsmart-tech.org (attivo a ottobre 2017)
- www.buildingsmart.org (attivo a ottobre 2017)
- www.buildigsmartcanada.ca (attivo a ottobre 2017)
- www.uni.com (attivo a ottobre 2017)
- www.iso.org (attivo a settembre 2017)
- www.autodesk.it (attivo a ottobre 2017)
- www.tekla.com (attivo a settembre 2017)
- http://rdf.bg (attivo a agosto 2017)
- www.visualstudio.com (attivo a ottobre 2017)
- https://bimdictionary.com/ (attivo a ottobre 2017)
- http://bimvision.eu (attivo a settembre 2017)
- www.solibri.com (attivo a settembre 2017)
- www.team-solution.de (attivo a settembre 2017)
- www.yworks.com (attivo a settembre 2017)
- http://notepad-plus-plus.org (attivo a agosto 2017)
- kompozer.net (attivo a ottobre 2017)
- www.biminteroperabilitytools.com/modelchecker/ (attivo a settembre 2017)
- http://ifcopenshell.org/ (attivo a agosto 2017)
- www.drawingtothefuture.polito.it (attivo a ottobre 2017)
- www.ifcwiki.org (attivo a luglio 2017)
- www.graphisoft.com/archicad (attivo a settembre 2017)
- www.html.it (attivo a settembre 2017)
- it.qr-code-generator.com (attivo a ottobre 2017)