

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

Corso di Laurea Magistrale in Ingegneria Edile-Architettura



**STRUCTURAL INFORMATION MODELING  
PROPOSTA DI SVILUPPO DI STANDARD INFORMATIVI OPEN BIM  
PER L'INTEGRAZIONE DI ANALISI STRUTTURALI DELL'ESISTENTE**

Relatrice:  
Chiar.ma prof.ssa MARIA ROSA VALLUZZI  
Correlatori:  
PhD Ing. CARLO ZANCHETTA  
Ing. ALBERTO CALABRIA

Laureando: GIOVANNI MARCOLIN

n° 1024039

**ANNO ACCADEMICO 2016-2017**



Una tesi per tutti e per nessuno, ma che non è filosofia;

sperando possa rivelarsi utile ed essere apprezzata, a chi ne trarrebbe vantaggio.

<b>1. INTRODUZIONE</b>	<b>6</b>
<b>2. METODOLOGIA</b>	<b>12</b>
<b>3. IFC</b>	<b>18</b>
<b>3A. EXPRESS-G</b>	<b>26</b>
<b>3B. IFC OGGI</b>	<b>48</b>
<b>3C. STRUCTURAL ANALYSIS DOMAIN</b>	<b>58</b>
<b>3D. STRUCTURAL LOAD RESOURCE</b>	<b>100</b>
<b>3E. A COMPLETARE</b>	<b>116</b>
<b>4. NAFEMS</b>	<b>122</b>
<b>5. NECESSITÀ</b>	<b>132</b>
<b>5A. IN ANALISI AGLI E. F.</b>	<b>134</b>
<b>5B. IN DINAMICA DELLE STRUTTURE</b>	<b>144</b>
<b>6. PROPOSTE IN LETTERATURA</b>	<b>152</b>
<b>6A. PROPOSTA GRUPPO ST-7</b>	<b>154</b>
<b>6B. PROPOSTA SOFTING SRL</b>	<b>172</b>
<b>7. UNA NUOVA PROPOSTA</b>	<b>188</b>



<b>7A. ELEMENTI FINITI</b>	<b>192</b>
<b>7B. ANALISI DINAMICHE</b>	<b>206</b>
<b>7C. UN SISTEMA COERENTE</b>	<b>218</b>
<b>8. ULTERIORI SVILUPPI</b>	<b>222</b>
<b>8A. IN PRATICA</b>	<b>228</b>
<b>9. VALIDAZIONE</b>	<b>256</b>
<b>9A. RISULTATI</b>	<b>282</b>
<b>10. CONCLUSIONI</b>	<b>296</b>
<b>11. BIBLIOGRAFIA</b>	<b>300</b>
<b>12. INDICE DELLE FIGURE</b>	<b>312</b>

# **1. introduzione**

L'obiettivo di questa tesi è proporre un ampliamento dello standard delle *Industry Foundation Classes*, abbreviato in IFC, di buildingSMART perché questo sia utilizzabile anche nell'ambito dell'ingegneria strutturale. Lo standard tratta l'informazione BIM in tutti i suoi ambiti e si pone come riferimento da seguire perché l'informazione BIM sia interoperabile efficacemente.



Figura 1 - buildingSMART. Logo.

Difatti lo standard, alla data attuale, appare ancora in fase di sviluppo e quindi, almeno in parte, incompleto. Questo nonostante nessun programma relativo a nessuna disciplina, architettonica, strutturale, impiantistica, energetica o quant'altro, riesca a sfruttare appieno la capacità semantica di trasmissione di informazioni relative all'intero ciclo di vita utile dell'edificio che lo standard offre. Ovvero non è possibile assegnare, ad esempio, ad un certo oggetto o processo tutte le proprietà previste da IFC o, addirittura, anche se inseribili all'interno del software, spesso queste non vengono totalmente mantenute una volta che si passa dal formato proprietario al formato “.ifc”.

In pratica, al giorno d'oggi sono presenti due problematiche: la prima riguarda sostanzialmente la completezza dell'informazione catalogabile attraverso lo standard, la seconda la capacità dei singoli software di sfruttarlo effettivamente.

Andando a valutare queste due problematiche per quanto riguarda la disciplina dell'ingegneria strutturale la situazione si rivela essere anche più critica.

La maggior parte dei software commerciali per analisi strutturali, di un file IFC, riesce a recepire solamente gli elementi comuni a più discipline, ossia elementi travi, pilastri, solai eccetera, senza poter recepire, o attivare, il modello analitico della struttura. Nonostante questa informazione sia già standardizzata all'interno del formato IFC e quindi sia possibile mantenerla nel formato di interscambio di buildingSMART. Questo perché, tecnicamente, i software per analisi strutturali si appoggiano a una *Model View Definition* diversa da quella a loro necessaria: la maggior parte di questi fa riferimento alla *Coordination View 2.0* invece che alla più corretta *Structural Analysis View*. I programmi che, poi, sfruttano questa, spesso si ritrovano a dover far specificare all'utente un set di proprietà di esportazione, che dia indicazioni specifiche a riguardo, per poter mantenere informazioni già assegnate

all'interno del modello. In parole povere, nell'enorme database costituito dalle *Industry Foundation Classes* i software strutturali ottengono e gestiscono informazioni in una parte del file che non è a loro dedicata né specifica, con tutte le conseguenze negative che a ciò possono essere connesse.

Come il fatto che informazioni come le schematizzazioni degli elementi strutturali, i vincoli esterni e interni, i carichi e le combinazioni di carichi, i materiali, le sezioni e i risultati di analisi non sono recepibili dalla gran parte dei software, anche nel caso in cui queste informazioni siano presenti nel file. Questo porta come conseguenza che a ogni avanzamento di progetto è necessario, di volta in volta, reinserirli senza possibilità di mantenimento, né di confronto, con le precedenti versioni. Un duro colpo per ogni tentativo di interoperabilità e circostanza ben distante da quella che è la filosofia BIM.

Ma anche tralasciando la situazione attuale dei programmi e andando a valutare il formato IFC per saggiarne la bontà, capirne la possibilità di utilizzo e la completezza, come avverrà in seguito, ci si trova davanti uno standard che vuole classificare compiutamente il mondo delle analisi strutturali, appoggiandosi agli elementi del modello già presenti, perché interdisciplinari, e associandovi ipotesi e informazioni della suddetta disciplina.

Su iniziativa della stessa buildingSMART, vengono anche riportate quali siano alcune delle mancanze dello stesso dominio delle analisi strutturali, sottintendendo come questi siano i prossimi punti che si intendono implementare. Viene riferito che non sono ancora standardizzate le informazioni relative alle analisi dinamiche; i carichi dovuti alla precompressione; la possibilità di descrivere un elemento finito e, conseguentemente, i risultati dettagliati di una mesh di elementi finiti, siano essi espressi in sforzi o in deformazioni in un elemento strutturale.

Queste sono, quindi, le due problematiche che affliggono IFC oggi e, se la situazione attuale si mostra dinamicamente in continui sviluppo e crescita, è anche vero che questa sia ancora ampiamente e gravemente incompleta da ambo i lati, soprattutto se considerata dal punto di vista italiano, caratterizzato da una peculiare situazione fatta di un'elevata vulnerabilità sismica e da un consistente patrimonio storico, artistico e culturale da tutelare urgentemente. Non sono, infatti, molti i Paesi al

mondo che convivono con un'emergenza sismica così elevata e ancora meno sono i Paesi con un patrimonio simile da difendere e tramandare.

Alla luce di ciò risulta comprensibile una perplessità tutta italiana a riguardo della classificazione di buildingSMART, anche da un'ottica ingegneristica, proprio perché l'ingegneria strutturale italiana è legata a doppio filo con l'ingegneria sismica e con la necessità di mettere in sicurezza i luoghi storici, dando alle strutture caratteristiche tali per cui queste possano sopportare i terremoti.

Si rivela impellente la necessità di progredire nella standardizzazione, rimediando intanto alle mancanze relative la descrizione e rappresentazione di un modello agli elementi finiti e la descrizione completa del problema strutturale sismico e dinamico, cosicché sia possibile sfruttare compiutamente una versione completa di IFC all'interno di programmi di analisi strutturali, man mano che i sistemi informatici si sviluppano e progrediscono. Il passo successivo prevedrà un'estensione dello standard per andare a descrivere anche altri tipi di software per analisi strutturali, primi fra tutti i software che effettuano analisi a telaio equivalente, per edifici in muratura portante. Ciò significa, ovviamente, che anche diversi materiali strutturali dovranno essere contemplati, come in questo caso i mattoni in laterizio; ma ulteriore conseguenza sarà quella di dover descrivere esaustivamente anche il mondo del restauro in tutti i suoi aspetti, sia architettonici che strutturali.

Ancora di più, lo standard si rivela necessario per la funzione che assume nel mondo degli appalti pubblici: essendo un formato libero, aperto e open-source, quindi fruibile a tutti, le Pubbliche Amministrazioni sono, giustamente, sempre più indirizzate a preferirlo qualora richiedano formati BIM per la partecipazione ai concorsi. I vantaggi, per le stesse, nell'avere a disposizione strumenti BIM sono indiscussi e non verranno qui affrontati, ma, una volta che questa pratica sia affermata, è impensabile che un ente pubblico penalizzi o favorisca dei professionisti sulla base degli strumenti software che utilizzano per svolgere la loro professione, quindi si presenterà una vera e propria necessità delle *Industry Foundation Classes* come strumento neutro, indipendente e di qualità garantita per la trasmissione delle informazioni.

Una volta completato la classificazione IFC, o comunque ampliata sufficientemente, rimane evidente come l'ostacolo maggiore sia quello relativo al fatto che alcuni produttori di software oggi tendono

ancora a privilegiare un'interoperabilità interna, tra gli applicativi specifici per le singole discipline, appartenenti al loro portafoglio. Questa non è una strada percorribile in nessuna maniera. Già i primi programmi di disegno CAD nel 1982 nacquero accompagnati da un formato di file d'interscambio fin dalla loro origine, il formato DXF: ancora di più necessiteranno di un formato equivalente i programmi BIM. E, al di là di tutto ciò, rimane un'ottima mossa di marketing per una software house implementare il loro software dimodoché possa confrontarsi con IFC perché darebbe al loro prodotto un'indubbia miglior competitività sul mercato.

Per tirare le somme, l'ingegneria italiana, soprattutto quella sismica, ha quindi l'impellente bisogno che venga esteso il linguaggio IFC, perché questo possa avere futuro, in quanto praticamente applicabile, all'interno del nostro Paese. È, quindi, prioritario completare il prima possibile IFC, anche perché i produttori di software possano avere le linee guida e gli strumenti per poter sviluppare i loro prodotti in ottica di interoperabilità.

Ciò si traduce in una concreta e rilevante possibilità: poter contribuire a livello internazionale all'uniformazione e alla diffusione di una conoscenza scientifica importante come questa, attraverso uno strumento sempre più in via di affermazione e di indubbia efficacia.

Questo è effettivamente l'obiettivo ultimo di questa tesi.



## **2. metodologia**



Per fare ciò si è dovuto, anzitutto, comprendere:

- La struttura attuale dello standard IFC e come esso funzioni;
- Il funzionamento di un modello agli elementi finiti, i suoi risultati, e una sua, se possibile, standardizzazione;
- Infine, il mondo delle analisi dinamiche, sia a livello teorico, sia come siano effettivamente svolte in un software agli elementi finiti.

Per, infine, arrivare ad avanzare una proposta, si spera, aggiornata, coerente ed efficace ci si è poi voluti confrontare con eventuali precedenti proposte di ampliamento di IFC nello stesso tema, ossia principalmente sia le analisi dinamiche sia la descrizione di elementi finiti. Infine, una breve ma interessante applicazione pratica mostra lo stato di fatto, le potenzialità e i limiti dello standard IFC applicato in un software agli elementi finiti.

A seguito si esplicita la metodologia applicata a questo lavoro. Se l'obiettivo finale è, appunto, l'ampliamento dello standard delle *Industry Foundation Classes*, il primo step che si è reso necessario è stato il confrontarsi con lo stato dell'arte in cui questo si trova.

## **LO STATO DELL'ARTE**

È stato un lavoro ampio, poiché, come si è forse capito, prima di tutto si è dovuto imparare il linguaggio di modellizzazione EXPRESS e la sua espressione diagrammatica EXPRESS-G, perché è attraverso di questo che IFC è implementato. Poi si è dovuto comprendere a fondo cosa sia effettivamente descritto e standardizzato nello standard stesso per quanto riguarda l'esteso argomento delle strutture. Tutto ciò è stata un'attività relativamente complicata perché la conoscenza di EXPRESS-G doveva essere decisamente buona per evitare fraintendimenti e sviste.

Una volta fatto questo, cioè conosciute a fondo le *Industry Foundation Classes* per quanto riguarda le strutture si è dovuto, invece, capire quale sia lo stato dell'arte oggi per le analisi agli elementi finiti e per le analisi in ambito dinamico, poiché poter descrivere queste all'interno dello standard era prioritariamente individuato da buildingSMART come il successivo passo da compiere. E quindi, dopo

essersi guardati attorno si è scelto, appunto, NAFEMS, che lavora proprio nell'ambito della validazione dei metodi di calcolo agli elementi finiti e dei risultati delle analisi condotte. Ci si è dunque preparati sia sulla teoria degli elementi finiti, per poter capire adeguatamente l'argomento, e in questo sono tornati utili anche vari libri di testo, e per poter capire i suggerimenti di implementazione di un software siffatto. Si è dovuto comprendere tanto la teoria matematica che sta alla base del tutto, quanto il suo funzionamento, i risultati che porta e, quindi, il modo in cui sono ottenuti, ma, ancora di più si è dovuto anche guardare alle singole applicazioni pratiche che sono, dai vari software, proposte in commercio, per capire così quale sia lo stato dell'arte di questi particolari tipi di modelli e cosa sia effettivamente necessario, e come, riportare in ambiente BIM. Ciò si è concretizzato in una breve incursione su alcuni noti programmi, STRAUS7 e TNO Diana ma, di più, in un rapido confronto con i manuali d'utilizzo di quest'ultimo, su cui si è ritornati di tanto in tanto come ulteriore conferma di quanto si trovava tra le proposte di NAFEMS.

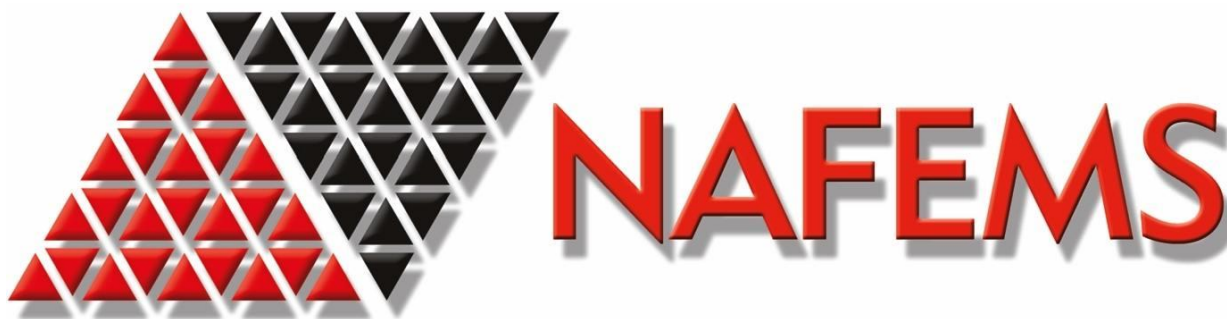


Figura 2 - NAFEMS. Logo.

Contemporaneamente si è reso necessario una formazione simile per quanto riguarda le analisi dinamiche: nuovamente attraverso libri di testo ci si è costruita una solida base conoscitiva teorica. Si è poi scoperto che sempre NAFEMS, occupandosi anche, in generale, di validazione di metodi di calcolo, ha esplicitato, in modo abbastanza generale, anche come debbano essere condotte le analisi dinamiche, con quali dati e quali risultati ci si debba aspettare. Anche qui ci si è dovuti confrontare con la teoria fisica che spiega questi fenomeni e con il loro aspetto applicativo, ossia come sono affrontate in pratica nella consuetudine e all'interno dei software commerciali, per far sì che i risultati che queste portano siano compatibili con entrambi all'interno di un ambiente BIM.

Questo, in pratica, è stato il processo conoscitivo e ha portato alla conoscenza dei quattro standard utili per questa tesi. Arrivati a questo punto difatti si conosceva la struttura attuale dello standard IFC,

e il suo linguaggio di implementazione, e come esso funzioni; il funzionamento di un modello agli elementi finiti e la standardizzazione relativa ad essi come definita dalla *National Agency for Finite Element Methods & Standards*; infine, il mondo delle analisi dinamiche, sia a livello teorico, sia come siano effettivamente svolte in un software agli elementi finiti.

## **ANALISI IN LETTERATURA**

Il passo successivo è stato il confronto in letteratura: attraverso siti dedicati alla catalogazione e alla consultazione di pubblicazioni scientifiche si è andati in cerca di tutto ciò che potesse essere utile agli obiettivi che ci si era posti. Purtroppo però la maggior parte delle pubblicazioni che hanno a che fare con i software BIM si concentrano sull'utilizzo di particolari software e, soprattutto, sulla loro interoperabilità all'interno della stessa *Model View Definition* o anche tra diverse MVD o, spesso, sull'interoperabilità al di fuori di IFC. Poi, le principali pubblicazioni di software agli elementi finiti che lavorano in ambito BIM sono i relativi fogli di validazione dei risultati in certi casi di riferimento così come proposti e garantiti da NAFEMS, ad esempio i risultati di sforzi e deformazioni di una trave a mensola con profilo a Z, a seconda della *mesh* con cui la si modella, con carico all'estremità eccentrico rispetto all'asse della trave, ma appartenente al piano ortogonale all'asse stesso. Ciò è servito a riconfermare l'autorevolezza dell'agenzia inglese, scelta dai maggiori produttori di software come riferimento per garantire i loro prodotti. In sostanza, dell'immenso patrimonio scientifico odierno sono emerse solamente due proposte di ampliamento dello standard IFC, in due date diverse. La prima piuttosto datata, perché di oramai dieci anni fa, la seconda molto più recente, sviluppata durante il corso del 2016. Si è proceduto quindi ad analizzarle a fondo e a valutarle: queste due attività sono andate di pari passo, sempre confrontandosi criticamente con i riferimenti proposti da NAFEMS e la conoscenza scientifica da libri di testo, così che fosse chiaro non solamente il “cosa” stessero proponendo i ricercatori, ma anche il “perché” la loro proposta assumeva esattamente quella forma.

## **STANDARDIZZAZIONE E NORMAZIONE**

A seguire, vista l'incompletezza e l'imprecisione di queste in alcune loro parti, è stato più che naturale voler proseguire, implementando una proposta il più valida possibile, basata sulla conoscenza dello

stato dell'arte che ci si era formati e su cui ci si è sempre conformati. Ovviamente queste proposte non si sono rivelate da cestinare in maniera assoluta, poiché, in molte loro consistenti parti, sono sostanzialmente corrette e molto valide, se non altro nella loro struttura concettuale: sono servite quindi come base di partenza per la proposta che si è avanzata. Si è quindi avanzata una proposta di ampliamento di IFC, secondo il linguaggio di modellazione EXPRESS-G, in modo che questo standard possa andare a includere modelli, analisi e risultati agli elementi finiti e azioni, reazioni e risultati delle analisi dinamiche.

## **ULTERIORI SVILUPPI**

Poiché quanto appena proposto, cerca di rispondere a ciò che buildingSMART richiedeva direttamente, è parso utile poi fare altre considerazioni, accompagnate da una sintetica ulteriore sperimentazione, sugli sviluppi futuri di IFC. Se l'idea è quella di descrivere l'intera industria delle costruzioni, allora è evidente come di strada da fare ce ne sia ancora: difatti la maggior parte di ciò che riguarda le murature portanti e il restauro, ad esclusione di buona parte della sola modellazione geometrica, è ancora oggi escluso da questa standardizzazione.

Ed è proprio in questa strada che questa tesi si inserisce, nel lavoro già cominciato da Giulio Pasqualotto con la sua tesi, di standardizzazione dell'informazione relativa all'informazione BIM del degrado degli edifici. Si vuole quindi poter descrivere sempre più compiutamente, sempre meglio, sempre correttamente, l'industria delle costruzioni in tutti i suoi aspetti, specialmente quelli successivi alla realizzazione iniziale del manufatto. Si spera, con questo lavoro di esserci riusciti.

## **VALIDAZIONE, RISULTATI E CONCLUSIONI**

Alla luce di tutto ciò si è provveduto a valutare un programma di analisi strutturali sulla base della sua capacità di interfacciarsi con IFC. La scelta è ricaduta su TNO Diana, un software agli elementi finiti specializzato in analisi dinamiche e Nòlian. Se questi si rivelano ottimi solutori di modelli FEM, non è stato questo l'unico parametro di valutazione che ha fatto sì che si scegliessero questi per una breve validazione: la gran parte dei software agli elementi finiti si confronta agli ambienti BIM per lo più

attraverso i formati proprietari con possibilità importative specifiche per lo specifico formato di una o l'altra software house produttrice di programmi BIM. Ad esempio, altri software famosi e validi agli elementi finiti, come midas GEN e Straus7, permettono un'interoperabilità non standardizzata attraverso IFC, ma attraverso i più comuni formati proprietari: midas GEN è ottimizzato, ad esempio, per l'importazione di formati Autodesk Revit. Similmente si comporta Tekla Structures. Punto interessante di Tekla Structures e midas GEN è la possibilità di produrre in uscita un file IFC, cioè comodamente leggibile da molti software BIM che implementano IFC; ciononostante, non si è potuti scegliere questi poiché l'impossibilità di introdurre un file in ingresso secondo le formulazioni dello standard delle *Industry Foundation Classes* non dà alcuna possibilità di interoperabilità propriamente detta, di interscambio di un'informazione in evoluzione e aggiornamento, dovendo, di volta in volta, rimodellare il tutto, virtualmente da zero ogni volta. TNO Diana, seppur con alcuni limiti non indifferenti, permette di importare, invece un modello IFC già sviluppato, per cui lo si è scelto per una poter valutare come queste informazioni viaggino e siano mantenute, analizzando criticamente i risultati ottenuti. Purtroppo nemmeno con TNO Diana è possibile un'interoperabilità completa, poiché non può produrre file IFC. Similmente Nòlian: permette anch'esso l'importazione di un modello IFC. Le analisi così condotte con entrambi i software e i risultati ottenuti sono stati poi confrontati.

Nonostante questi limiti, sulla base di ciò che è stato possibile effettuare con questi software e sulla base dell'attività di normazione svolta sono, infine, state stilate le conclusioni.

# 3. IFC

Il formato delle *Industry Foundation Classes*, abbreviato in IFC nasce nel 1995 come formato di scambio libero, cioè gratuito, e neutrale per l'edilizia. L'obiettivo che l'associazione sviluppatrice IAI, acronimo di International Alliance for Interoperability e poi diventata buildingSMART nel 2008, si pone è che IFC possa essere adatto per:

- Sviluppare nuovi standard internazionali per il BIM;
- Per promuovere il processo di sostenibilità del mercato attraverso appositi progetti;
- Creare una rete di specifiche tecniche, *data models* e guide scritte, liberamente consultabili.



Figura 3 - buildingSMART. Logo.

Si procede a una disamina completa dello standard per poter capire al meglio come sia oggi strutturato, in modo da poter avanzare una proposta coerente ed efficace.

## **PREFAZIONE IFC<sup>1</sup>**

buildingSMART International Limited è un'associazione no-profit internazionale formata da capitoli regionali e nazionali.

La visione di buildingSMART è: “Sostenibilità costruendo PIÙ INTELLIGENTEMENTE.”;

La missione di buildingSMART è: “Contribuire all'ambiente costruito sostenibile attraverso una condivisione e comunicazione PIÙ INTELLIGENTE di informazioni, usando standard aperti e internazionali, nel settore degli edifici e delle costruzioni, pubblico e privato.”;

I valori dell'associazione sono:

---

1 - buildingSMART, “IFC4 - Add.2; Foreword”

Neutralità e indipendenza e quindi nessun legame con nessun gruppo o compagnia, sia di fornitori di software, sia imprese edili, sia architetti né nessun altro, nemmeno governi;

Apertura e trasparenza, perché i processi per cui buildingSMART opera necessitano di essere chiari e compresi da tutti per dimostrare indipendenza e neutralità;

essere un'organizzazione no-profit, significa che ogni profitto che può giungere da pubblicità o attività semi-commerciali sono utilizzati per scopi non commerciali, come lo sviluppo degli standard.

Gli obiettivi dell'associazione nel perseguire la sua missione sono:

Sviluppare e mantenere standard aperti e internazionali per il Building Information Modeling (open BIM);

Accelerare l'assimilazione dell'interoperabilità da parte del mercato attraverso il successo di progetti sostenibili;

Provvedere opportunità di conoscenza, specificazioni e guide scritte;

Risolvere i problemi di alto costo che ostacolano la condivisione di dati;

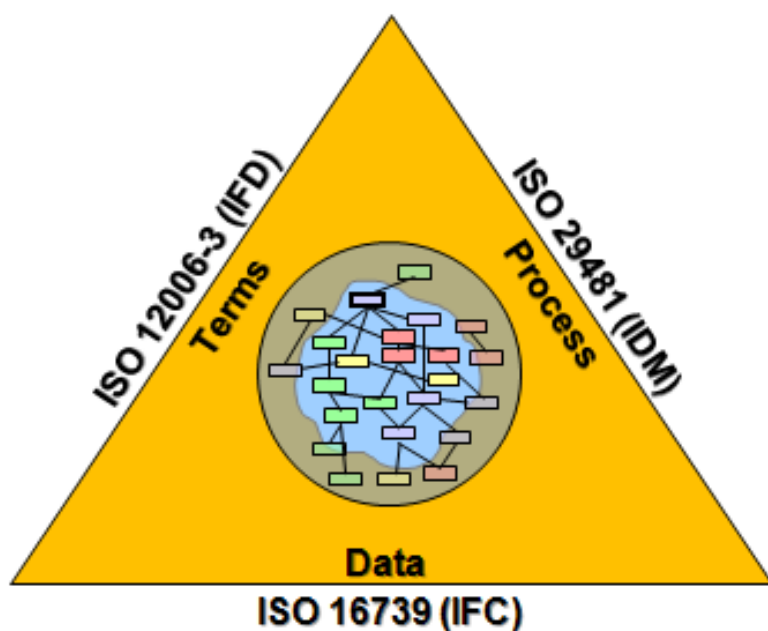


Figura 4 - schema interscambio dati IFC.

Estendere i processi e le tecnologie di buildingSMART all'intero ambiente costruito, attraverso il suo ciclo di vita e comprendendo la direzione, la produzione, la gestione dell'organismo e l'ingegnerizzazione delle manutenzioni.

buildingSMART sviluppa standard aperti per il Building Information Modelling, chiamato openBIM. Gli standard coinvolgono modelli di dati, processi e termini. Le specifiche delle



*Industry Foundation Classes*, letteralmente Classe di Fondamento dell'Industria (delle costruzioni), abbreviate in IFC, costituiscono le specificazioni buildingSMART per i *data model* buildingSMART. Termini di proprietà, definiti in IFC; sono collegati ai termini definiti nel dizionario di dati di buildingSMART.

L'associazione si coordina con l'International Standardization Organization ISO, con corpi regionali e nazionali di standardizzazione per lo sviluppo internazionale, nazionale e regionale degli standard. Le specificazioni delle *Industry Foundation Classes* sono registrate come ISO 16739; buildingSMART lavora attivamente come un contatto di eccellenza con ISO/TC 59/SC 13; ulteriori edizioni di IFC verranno sottoposte a ISO/TC 59/SC 13/JWG 12 per l'accettazione come successiva appendice della ISO 16739.<sup>2</sup>



Figura 5 - ISO. Logo.

---

2 - buildingSMART, "IFC4 - Add.2; Foreword"

## INTRODUZIONE IFC

*Industry Foundation Classes*, abbreviato in “IFC” rappresenta una specifica, gratuita e *open source*, descrizione dettagliata per le informazioni che vengono scambiate e condivise attraverso software di *Building Information Modeling*, abbreviato in “BIM”, tra i vari partecipanti al processo edilizio o alla gestione del complesso. È lo standard internazionale openBIM.

Questo standard consiste di un sistema di schemi gerarchici, i cosiddetti *data schemas*, che relazionano definizioni di proprietà e di quantità, il tutto rappresentato con le specificazioni di uno schema di specificazione EXPRESS o, alternativamente, in XML.

Un’applicazione software conforme allo standard è richiesta per supportare un ben definito sottoinsieme dei diagrammi di informazioni referenziati; il sottoinsieme viene chiamato *Model View Definition*, un ben preciso sottoinsieme dei modelli di dati e dei loro riferimenti. Una particolare *Model View Definition*, abbreviato in MVD, è definite per supportare un, o più di uno, riconosciuto flusso di lavoro all’interno della costruzione o gestione dell’immobile, nell’industria delle costruzioni. Ogni flusso di lavoro identifica richieste di scambio di dati che devono essere possibile all’interno dalle applicazioni software conformi. buildingSMART pubblica le specificazioni per individuare e scambiare le informazioni di ciascuna *Model View Definition*.

Il sito ufficiale è <http://buildingSMART-tech.com> in cui vengono pubblicate queste specifiche, le relative MVD e le loro richieste di scambi, oltre a materiali di supporto come accordi di implementazioni, esempi di set di dati, riferimenti a strumenti di sviluppo, forum di discussioni, database di problematiche varie e programmi per la certificazione.

Le specifiche di IFC includono termini, concetti e *data item* degli oggetti che provengono dalle discipline, i settori e le professionalità del settore industriale delle costruzioni e della gestione del complesso edilizio. I termini e i concetti si prevede siano espressi in un inglese semplice, mentre i *data item*, seguono una precisa convenzione di nomenclatura:

I nomi dei *data item* per *type*, *entity*, *rule* e *function* iniziano con il prefisso “Ifc” e continuano con il loro nome in inglese scritto secondo la convenzione chiamata “CamelCase”. Ossia che le varie parole siano scritte unite tra di loro, con l’iniziale maiuscola, senza spazi né altri segni atti a separare le parole stesse;

Il nome degli *attribute* di una *entity* seguono la stessa convenzione, sempre in inglese, in CamelCase, senza alcun tipo di prefisso;

I nomi dei *property set*, anch’essi parte dello standard, iniziano con il prefisso “Pset\_” e continuano con il loro nome in inglese scritto in CamelCase;

I nomi dei *quantity set*, anch’essi parte dello standard, iniziano con il prefisso “Qto\_” e continuano con il loro nome in inglese scritto in CamelCase.

L’architettura dei *data schemas* di IFC vanno a definire quattro livelli concettuali, che catalogano e raccolgono tutti i modelli usati per definire le *entities* usate in IFC. Ti tutto è organizzato secondo una gerarchia molto precisa, tale per cui ogni *data schema* è assegnato a esattamente a un livello concettuale, e ogni *entity* può essere messa in relazione solo ad altre *entities* di livello pari o inferiore al suo, mai superiore. Questi livelli, dal più basso al più alto, sono:

**1. Resource Layer:** è il livello più basso e contiene 26 *data schemas* particolari contenenti definizioni di risorse; queste definizioni non includono un identificatore globale e non possono essere utilizzate indipendentemente da una definizione specificata in un layer superiore. Ciò vuol dire che contiene degli “attivatori” del modello, cioè dei concetti essenziali, delle “risorse” come nel nome del livello, che completano gli specifici domini disciplinari. Difatti i *data schemas* di questo livello definiscono le proprietà base, come geometria, materiali e costi, racchiudendo informazioni a si può assegnare un valore o una descrizione

**2. Core Layer:** questo livello include il *data schema* del nucleo, il *Kernel Schema*, e i tre *Core Extension Schemas* che contengono le *entities* con le definizioni più generali. Tutte le *entities* definite in questo livello, o nei successivi, contengono un ID globale e unico e sono compilabili informazioni sui proprietari e sulla storia. È in questo livello che avviene la distinzione tra *product* e *process*, ossia tra elementi fisici e fasizzazione del processo.

**3. Interoperability Layer:** questo livello include *data schemas* che contengono definizioni specifiche di *entities* per un generale prodotto, un processo o una specializzazione di risorsa usato trasversalmente a varie discipline, fornendo, quindi, *objects* e *relationships* più precisi, come le *entities* trave o solaio. Queste definizioni sono tipicamente usate per scambi e condivisione di informazioni costruttive fra vari domini.

**4. Domain Layer:** il livello più alto include *data schemas* contenenti definizioni di *entities* che sono specificazioni di prodotti, processi o risorse specifiche di una certa disciplina e, inoltre, sono completamente autonome, non potendo essere fare riferimento a *entities* di altri livelli. Queste definizioni sono tipicamente utilizzate per scambi e condivisioni di informazioni all'interno di uno stesso dominio disciplinare<sup>3</sup>.

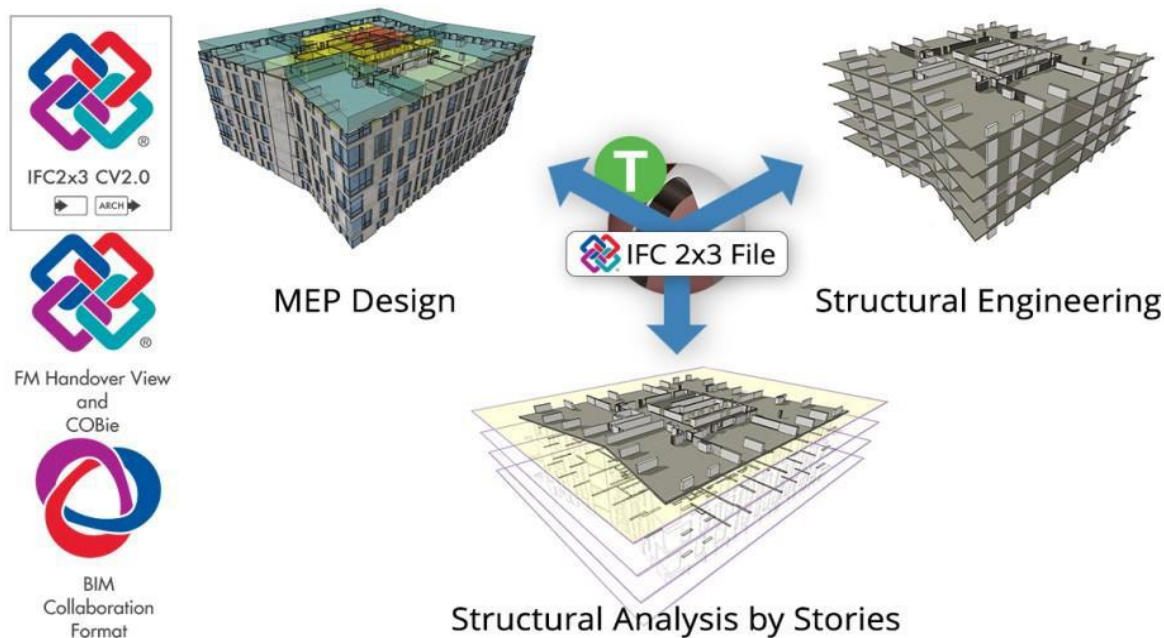


Figura 6 - buildingSMART. Schema esplicativo.

Come già visto ci si presentano alcuni termini di difficile comprensione, poiché specifiche dell'universo EXPRESS, come “*entity*” o “*attribute*”. A questo punto si pone come necessario spiegare brevemente cosa sia e come sia costituito questo linguaggio di modellizzazione.

---

3 - buildingSMART, “IFC4 – Add. 2; Introduction”



# **3A. EXPRESS-G**

A questo punto è necessario spiegare meglio alcuni concetti: lo standard IFC è implementato secondo il linguaggio di modellazione EXPRESS e, in particolare, secondo la sua interfaccia umana diagrammatica EXPRESS-G. Anch'esso risulta standardizzato da ISO nella norma ISO 10303-11<sup>4</sup> fin dal 1994. Sarà, quindi, utile spiegarlo, per capire effettivamente come sono costruite le *Industry Foundation Classes*, affrontando, poi, anche il capitolo 3 di IFC stesso, che riporta termini, definizioni e abbreviazioni, facendo riferimento anche norma relativa ad EXPRESS. Questo perché questo lavoro di tesi è fisicamente sviluppato soprattutto attraverso questo metalinguaggio. Difatti IFC si colloca nell'ambito della modellizzazione, non della modellazione, cioè della modellazione della struttura dei dati, anche detto *data modeling*, ed è quindi necessario potersi inserire correttamente nel sistema gerarchico al momento definito.

EXPRESS è formalmente definito dalla parte 11 della ISO 10303, la norma STEP<sup>5</sup>, ossia *Standard for the Exchange of Product model data*, che in pratica stabilisce uno standard per i modelli di dati per lo scambio di prodotti, trattando la rappresentazione informatica tridimensionale e lo scambio di informazioni non geometriche all'interno di software di rappresentazione grafica. L'obiettivo dello STEP è fornire uno strumento per descrivere compiutamente un'informazione in ogni momento storico del suo utilizzo ed è tipicamente sfruttato per la condivisione di informazioni tra programmi di disegni e macchine a controllo numerico, oppure per le attività di analisi ingegneristiche, oppure ancora per la gestione e la pubblicazione dei dati dei prodotti. Altro ambito in cui EXPRESS è ampiamente sfruttato è nella redazione di protocolli di rete. Infatti il fatto che la sua normativa di riferimento sia reperibile gratuitamente, nonostante sia norma ISO, ha fatto sì che si potesse sempre più raffinare e potesse coprire sempre più ambiti.

EXPRESS nasce proprio con l'obiettivo di rendere un'informazione interoperabile: è un linguaggio formale di modellazione che permette la creazione di modelli di dati per sistemi informativi, che si traduce, in IFC, nel permettere di creare schemi concettuali che forniscono relazioni tra gli oggetti informativi, potendo definire tutto esternamente in XML. Permette, in parole povere, di descrivere

---

4 - Standard ISO, 2004a;

5 - Standard ISO, 1994a.

un software ad un livello di astrazione superiore rispetto ai linguaggi di programmazione, permettendo di rappresentare una modellazione del sistema delle informazioni, sviluppando i modelli concettuali di un sistema che permettono, così, di scambiare informazioni. Infatti, un'informazione può essere definita interoperabile quando due differenti sistemi possono accedervi, duplicarla e modificarla, perché aderiscono ad uno stesso modello di dati per sistemi informativi, che può essere definito da EXPRESS, mentre l'integrazione di un'informazione è possibile quando si costituiscono archivi di informazioni unificate e razionalizzate, estrapolate da diversi sistemi operativi per un interesse comune. La diretta conseguenza di tutto ciò è che EXPRESS fornisce uno strumento per poter formare un database sicuro, garantendo il completo accesso a questi al di là di ogni possibile specifica applicazione o tecnologia, così che ogni utente possa vedere e validare l'informazione così come collocata all'interno del modello informativo. Secondo le parole dei suoi inventori "il linguaggio EXPRESS è stato sviluppato per fornire tutte le possibilità necessarie a rappresentare dati, indipendentemente dalla particolare tecnologia di implementazione."<sup>6</sup>.

Di particolare EXPRESS ha due funzionalità chiave: un'interfaccia umana, comprensibile dall'utente attraverso un intuitivo layout grafico, chiamata EXPRESS-G, e un'interfaccia processabile dalle macchine, perché in parte indirizzata al software e quindi scritto in linguaggio macchina, attraverso una forma lessicale così come specificata all'interno della ISO 10303-11<sup>7</sup>. Nello sviluppare questo lavoro si farà soprattutto riferimento alla modellazione diagrammatica, poiché è questo che viene utilizzato per la standardizzazione IFC.

Molti dei termini utilizzati coincidono con la terminologia propria di IFC, come si è già potuto vedere, ma è bene non confondere i due argomenti. Lo standard delle *Industry Foundation Classes* di buildingSMART viene sfruttata nell'ambito dell'interoperabilità dell'informazione all'interno del BIM; mentre il linguaggio EXPRESS è la base metodologica con cui viene creata in pratica questa struttura, e vale sia per IFC che per altri modelli di dati strutturati.

---

6 - "The EXPRESS language was developed to provide the full range of capabilities needed to represent data, independent of any particular implementation technology" Howard Mason (BAE Systems), Chairman of ISO TC184/SC4, Industrial Data;

7 - LSC Group, 2010.



Il linguaggio EXPRESS fornisce le capacità per sviluppare i modelli di informazioni, i quali sono poi idonei per realizzare implementazioni per lo scambio, il trasferimento e l'archiviazione dei dati. L'obiettivo finale è l'analisi dell'informazione, che coincide con lo scopo vitale di tutto il processo di sviluppo di un software. Questo processo include altri obiettivi, i quali corrispondono agli elementi dell'*information technology*, l'utilizzo di elaboratori ed attrezzature di telecomunicazione per memorizzare, recuperare, trasmettere e manipolare i dati.

Il perché implementare attraverso EXPRESS è tanto più evidente quanto si considerano i benefici, ottenuti sviluppando un software in maniera sicura e solida, sfruttando modelli standard dell'informazione, identificando precise e corrette opzioni di distribuzione per il linguaggio, sapendo effettivamente sfruttare EXPRESS in tutti i suoi aspetti. Questo perché alla base del processo di sviluppo globale dei software, su cui EXPRESS è stato implementato, vi è l'ingegnerizzazione dei sistemi, che sviluppa l'organizzazione e la strutturazione di sistemi complicati; permette così di includere anticipatamente l'identificazione dei requisiti funzionali corrispondenti alle richieste dell'utente, tra cui lo scambio, il trasferimento e l'archiviazione dei dati.

E ancora, appoggiandosi sugli standard ISO 10303, che definiscono anche EXPRESS, significa sfruttare dei protocolli per applicazioni che definiscono modelli di informazioni, garantendo un'interoperabilità a lungo termine dei dati stessi, evitando ogni possibile re-ingegnerizzazione e riprogettazione di sistemi interi poiché già rispondono a standard di efficienza, i quali considerano anche lo sviluppo.

## EXPRESS-G

Per l'interpretazione dell'utente, EXPRESS prevede una configurazione chiamata EXPRESS-G, che in sostanza riproduce graficamente un modello di informazioni. Per quanto una rappresentazione in EXPRESS-G possa sembrare semplice ad un primo impatto, e in effetti sia molto intuitivo, ricavare in maniera corretta le informazioni rappresentate non è molto immediato e richiede un investimento in analisti addestrati a comprenderne il linguaggio. Una corretta interpretazione dipende anche dall'abilità delle persone che stanno leggendo i modelli e che poi dovranno trasformare e trasmettere le informazioni contenute in caratteristiche da implementare<sup>8</sup>.

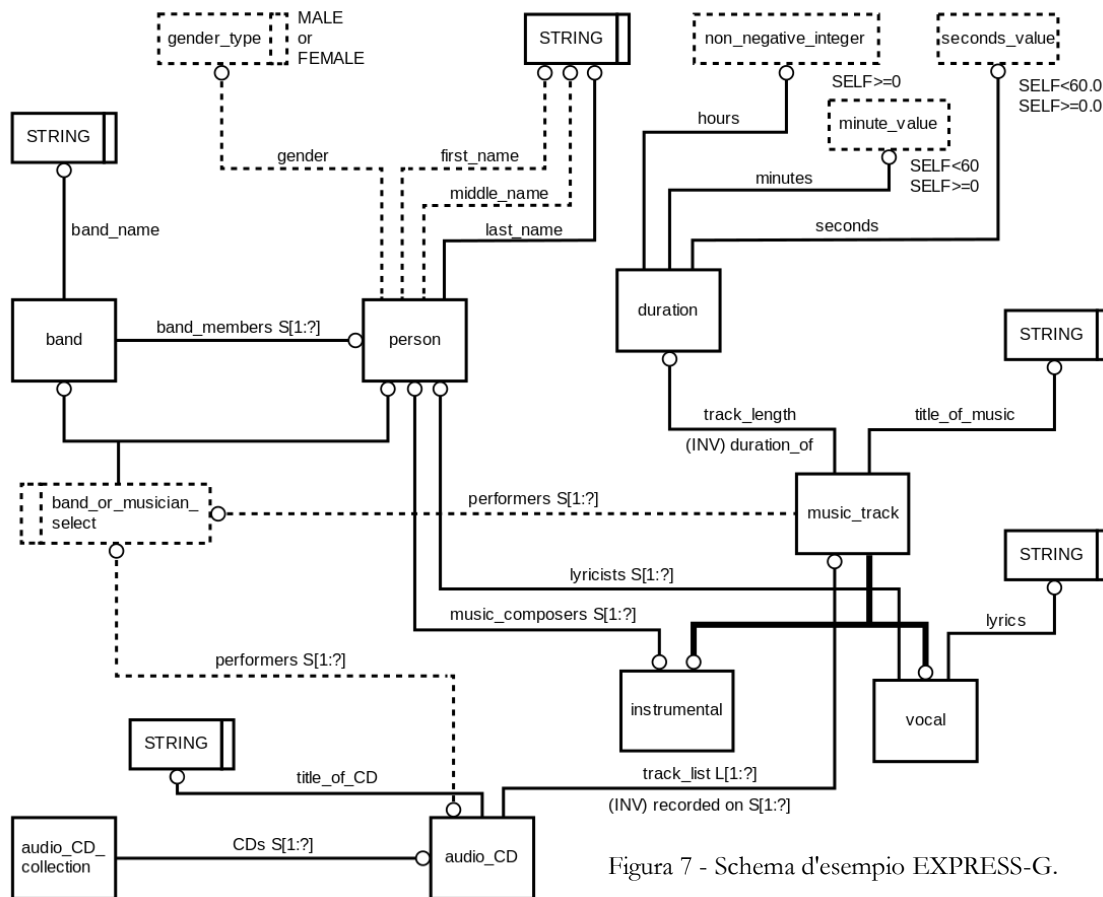


Figura 7 - Schema d'esempio EXPRESS-G.

8 - Luiten et al., "An information reference model for architecture, engineering, and construction. Management of Information Technology for Construction, World Scientific & Global Publication Services".

EXPRESS-G è considerato a tutti gli effetti uno standard grafico<sup>9</sup> usato per rappresentare le definizioni e le relazioni degli oggetti, come le *entities* e i *types*. Viene sempre definito all'interno dell'ISO 10303-11<sup>10</sup> e fa parte di uno dei tanti strumenti a disposizione di EXPRESS. Gli altri sono<sup>11</sup>:

-**EXPRESS-X**<sup>12</sup>: una mappatura del linguaggio in dati strutturati che permette all'utente di convertire i data in maniera conforme da uno schema EXPRESS ad un altro. In altri termini, fornisce la capacità di costruire dei traduttori che permettono lo scambio e il trasferimento delle informazioni;

-**Part 21**<sup>13</sup>: un codificatore di testo per le informazioni che usano uno specifico information model in EXPRESS, permette lo scambio di data sets usando una struttura comprensibile agli utenti ed eseguibile dal computer;

-**Part 28**<sup>14</sup>: un codificatore XML<sup>15</sup>, sempre per le informazioni che usano uno specifico *information model*, che permette lo scambio di *data sets* usando la tecnologia base fornita dai servizi web;

-**Standard Data Access Interface**<sup>16</sup>: spesso abbreviato in "SDAI" fornisce un generico set di operazioni col quale eseguire informazioni in maniera conforme ad un schema di dati sviluppato con EXPRESS. Lo SDAI permette lo sviluppo della capacità di condivisione dei dati, usando linguaggi di programmazione specifici ed obbligatori che sono sempre una parte della norma *STEP*. La norma include vincoli per Java, C, C++ e C#, dando la possibilità di creare *Application Programming Interfaces*, gli API. Queste sono le interfacce di programmazione delle applicazioni;

-**Binary representation**<sup>17</sup>: una rappresentazione binaria in HDF5 (Hierarchical Data Format 5<sup>18</sup>) per fornire forme compatte di file che contengano *data sets* molto grandi, per facilitarne lo scambio;

---

9 - ST-Developer Tools Reference, 2016;

10 - Annex B of Standard ISO, 2004a;

11 - LSC Group, 2010;

12 - Standard ISO, 2005;

13 - Standard ISO, 2016a;

14 - Standard ISO, 2007a;

15 - W3C, 2016;

16 - Standard ISO, 1998a;

17 - Standard ISO, 2011;

18 Il formato HDF5 è un data model, libreria e formato di file per la memorizzazione e la gestione dei dati (HDF Group, 2016)

-**Conformance testing**<sup>19</sup>: dei test di conformità definiti da più parti della norma ISO, che forniscono i significati e le definizioni con i quali assicurare che le implementazioni basate su EXPRESS raggiungano i requisiti necessari.

Un utente, sufficientemente qualificato nell'uso di questo insieme di strumenti, è in grado di creare un modello di informazioni, sia in forma testuale che in quella grafica, fornendo anche diverse capacità di implementazione. Attraverso questo modello sarà possibile archiviare dei dati, accedere e modificare gli stessi dati, accedere a file funzioni che da questi dati derivano.

Questa capacità di implementazione fornirà un particolare beneficio solo con l'interpretazione delle regole nell'*information model* e con l'applicazione di queste regole per far rispettare l'integrità dei *data sets*.

---

19 - Standard ISO, 1994b, 1998b, 2001, 2003.

## TERMINOLOGIA

EXPRESS-G è una notazione grafica per la modellazione in inglese una *graphical modeling notation*<sup>20</sup>, sviluppata con *STEP* ed usata per le definizioni IFC. È usato per identificare *classes*, *data attributes* delle *classes* e le relazioni esistenti tra le *classes*.

Vogliamo quindi definire i simboli, cioè gli elementi, che compongono EXPRESS-G per poter capire sia il programma sia la strutturazione delle *Industry Foundation Classes*. In ogni caso rimane lo scopo di implementare all'interno di queste un modello agli elementi finiti e poter descrivere le azioni dinamiche

I simboli di EXPRESS-G possono essere divisi in due categorie: diagrammi e frecce/collegamenti. I primi sono usati per contenere i *data items* di IFC, mentre i secondi creano le relazioni e definiscono gli *attributes* e le *relationship*.

**Class ed Entity:** con questi termini intendiamo degli oggetti informatici, ognuno definito come *class*; gli oggetti di una stessa *class* sono in qualche maniera simili, avendo caratteristiche, comportamenti comuni. In EXPRESS-G un'*entity* è l'elemento fondamentale e di base del diagramma. Sono rappresentati con un rettangolo a linea continua al cui interno si trova il nome della *class*. Un esempio di *entities* sono i set dei dipendenti di una compagnia o tutti i loro progetti di lavoro; una *class*, rispetto alle *entities*, richiede più informazioni per essere compiutamente descritta. È necessario poter descrivere totalmente un *object* della *class* per poter generalizzare e coprire tutte le possibilità di *objects*. All'interno delle *Industry Foundation Classes*, le due definizioni coincidono perfettamente.



Figura 8 - EXPRESS-G. Entity.

**Subtype:** due o più *entities* da alcune caratteristiche e/o comportamenti comuni possono essere raccolte al di sotto di una *entity* gerarchicamente superiore, un *supertype* o *superentity*, e vengono, così, chiamate *subtype* o *subentity*. Viene così definita una relazione tra differenti classi di tipo padre-figlio in

---

20 - EXPRESS-G è una notazione diagrammatica per modellare (Diagrammatic Modeling Notation DMN) con lo scopo di creare un *object-oriented information modeling*.

cui la sottoclasse contiene ulteriori specificazioni rispetto alla classe parente, la *superentity*; e, di più, eredita tutte le caratteristiche della parente. Ovviamente una *entity* può essere allo stesso momento sia *supertype* che *subtype*. Questo tipo di relazione è indicato con una linea spessa tra le classi e un cerchio dalla parte della classe figlio, che indica il verso.

In EXPRESS è permesso che i *subtype* siano inclusivi, ossia che un elemento padre possa essere allo stesso tempo specificato da due elementi figlio e, nel caso in cui, invece siano esclusivi, viene posto il numero uno sopra il ramo della relazione ad indicare che solo una può essere la classe figlio, tra le possibilità. Questo, però non viene utilizzato in IFC, in cui ogni tipo di *subtyping* è da intendersi esclusivo.



Figura 9 - EXPRESS-G. Abstract entity e subentity.

Nell'esempio in figura, poiché siamo all'interno di IFC, la classe padre a destra, è definita dalla classe figlio, a sinistra, che ne eredita le caratteristiche. In altre parole, possiamo definire proprietà dell'elemento edilizio, sapendo che poi questo sarà un muro. Per il prefisso (ABS) vedere il punto successivo.

**Abstract supertype:** sono un tipo particolare di *supertype*. L'*abstract supertype* non può esistere in se stesso, ma solo in virtù e in relazione con i suoi *subtypes*, venendo, così, completamente definito dalle classi figlio. La loro utilità è nel poter raccogliere gli *attributes* ad un livello superiore rispetto alla definizione dell'*item*, rendendoli, alla fine, ereditabili dalle classi inferiori e permettendo così, uno sviluppo dinamico della struttura, tale per cui ogni nuovo *subtype* eredita gli *attributes* generali che gli sono propri. È sulla base di questo concetto che la gerarchia prima accennata delle *Industry Foundation Classes* si struttura.

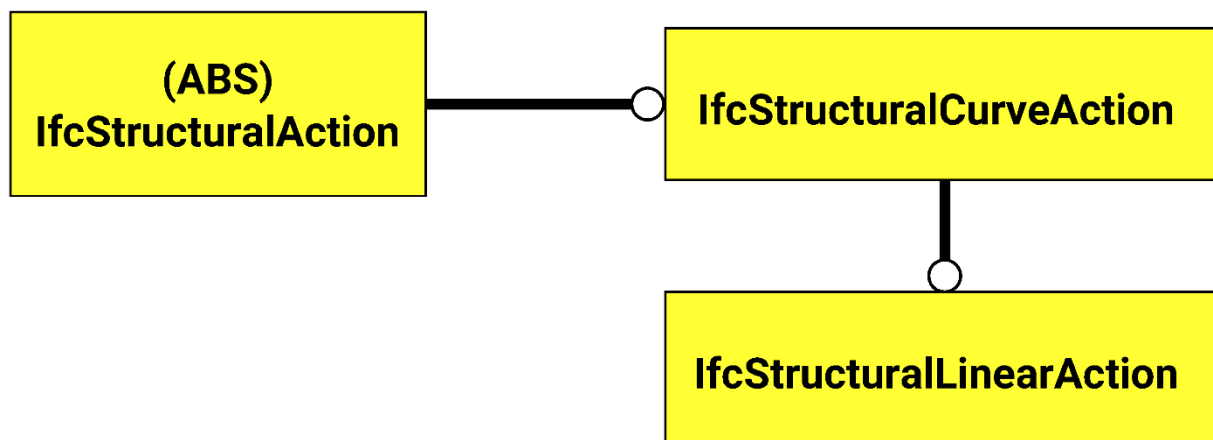


Figura 10 - EXPRESS-G. Abstract superentity, superentity e subentity.

Come si diceva, la classe padre *IfcStructuralAction* è di tipo *abstract* cioè le si possono definire proprietà che la classe figlio, in questo caso più di una, erediterà, ma sarà definita concretamente dalla classe figlio. La classe figlio è a sua volta una classe padre, ma di tipo normale, quindi sarà solo specificata, non definita totalmente, dalla sua classe figlio. Questo schema significa che un'azione strutturale deve necessariamente essere un'azione che insiste su di una curva che, a sua volta, può, ma non necessariamente, essere un'azione di valore lineare.

**Attribute:** con questo termini indichiamo le caratteristiche delle *entities* per supportarne l'uso e la comprensione e si traduce nel fatto che tutto ciò che si collega alla *class* è considerato *attribute* della stessa. Ogni *attribute* può essere obbligatorio oppure opzionale, ossia nel primo caso devono assolutamente avere un valore ogni volta che una *entity* viene definita, mentre nel secondo caso

possono essere assegnati, ma non sono necessari. Graficamente il primo viene reso con una linea continua tra la *class* e l'*attribute*, mentre il secondo con una linea tratteggiata; in ogni caso sopra la linea viene riportato un testo che lo descrive e all'estremità un cerchio mostra la direzione principale della relazione definita.



Figura 11 - EXPRESS-G. Attributes.

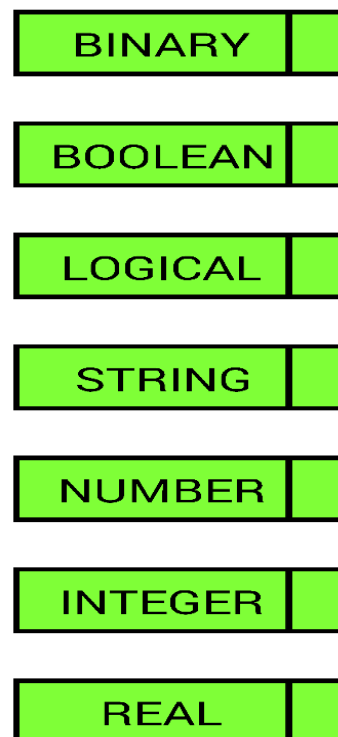
Generalmente, gli oggetti collegati da un *attribute*, appartengono a una delle tre tipologie di *type*, più sotto definiti: i *simple data types*, gli *enumerated types* oppure i *costruted* o *defined types*. Generalmente perché un *attribute* può mettere in relazione anche diverse *classes*: ad esempio, come vedremo meglio in seguito, le *entity IfcStructuralAnalysisModel* e *IfcStructuralLoadGroup* sono messe in relazione da due *attributes*, uno è l'inverso, che esplicitano che il modello di analisi *IsLoadedBy*, cioè è caricato, da un determinato gruppo di carichi e, viceversa, attraverso l'*inverse attributes LoadGroupFor* un gruppo di carichi è riferito a un certo modello per analisi. Così facendo una *entity* ha come *attribute* un valore di un'altra. Sebbene, infine, in EXPRESS sarebbe lecito l'utilizzo del tratto basso come in molte altre risorse STEP, in IFC si richiede che il nome degli *attributes* siano espressi in CamelCase. Rimane valido che non può mai essere utilizzato lo spazio.

**Simple data type:** sono le parti più piccole di EXPRESS ed EXPRESS-G, che non possono essere suddivise ulteriormente. Un *simple data types* è rappresentata graficamente come un rettangolo con una doppia linea verticale sul lato destro al cui interno, come le *class*, è riportato il nome. I *simple data types* consentiti sono:

-*BINARY*: una sequenza di bit, ognuno del quale deve avere valore "0" oppure "1";

-*BOOLEAN*: un valore che vale esclusivamente *TRUE* o *FALSE*;

Figura 12 - EXPRESS-G. Simple Data type.





-*LOGICAL*: un valore che vale *TRUE*, *FALSE* o *UNKOWN*;

-*STRING*: una sequenza di caratteri;

-*NUMBER*: un numero che può arbitrariamente valere come un intero o un reale;

-*INTEGER*: un numero intero che non contiene una frazione o un elemento decimale;

-*REAL*: un numero decimale. il punto decimale deve essere presente anche se un numero dichiarato come *REAL* corrisponde ad un numero intero come ad esempio 2.0).

**Enumerated type**: fornisce un range di possibili valori con i quali l'*attribute* può essere descritto. I valori vengo inseriti all'interno di una lista e solo uno di quelli elencati può essere scelto. Viene reso graficamente da un rettangolo tratteggiato con una seconda linea verticale a destra, prima della fine del box. Nuovamente al suo interno è presente il nome che, all'interno di IFC, deve concludersi con "*TypeEnum*", oltre che iniziare con il prefisso "*Ifc*". In IFC possono essere chiamati anche *declared data*

IfcActionTypeEnum

{PERMANENT\_G; *types* e possono,  
VARIABLE\_Q; opzionalmente,  
EXTRAORDINARY\_A; descrivere un *simple*  
USERDEFINED; *data type*.  
NOTDEFINED}

Figura 13 - EXPRESS-G. Enumerated Type.

Nell'esempio qui mostrato *IfcActionTypeEnum* permette di discriminare, come il nome fa capire, che tipo di azione sia quella a cui ci si riferisce, se si tratta di un carico permanente, variabile, straordinario, oppure definito dall'utente, oppure ancora, non sia stato ancora specificato di che tipo di azione si tratti.

**Defined type:** prende il posto di un *simple data type* ed è usato per rendere il significato del modello più chiaro. Viene rappresentato in EXPRESS-G con un box rettangolare con i bordi tratteggiati. In IFC si fa ampio uso dei *defined types*, spesso per trasmettere un'informazione in modo più completo: ad esempio, sebbene sia sempre utilizzabile un *simple type* di tipo *REAL* o simile, per definire una misura di una forza o di un momento, si utilizza *IfcForceMeasure* o *IfcTorqueMoment* rispettivamente. Di più, possono indirettamente portare con sé l'unità di misura. Oppure, ancora, per esprimere il nome

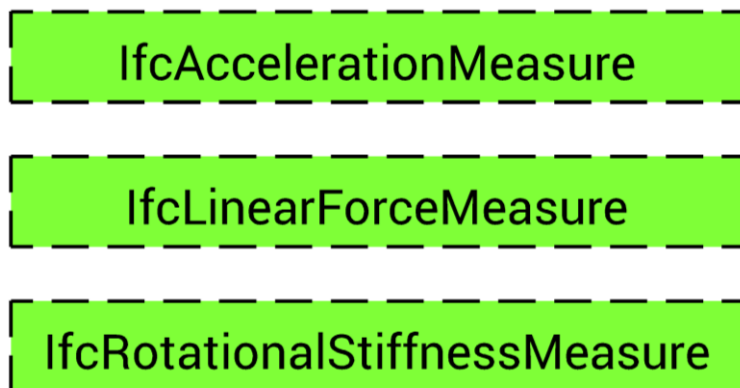


Figura 14 - EXPRESS-G. Defined Type.

di una *class* ci si può riferire al *simple type* *STRING*, ma si preferisce utilizzare *IfcText* che è dedicato alla trasmissione di dati scritti, difatti in una stringa possono comparire formule e numeri indifferentemente. Mentre *IfcText* è implementato per testi scritti veri e propri. In ogni caso il risultato sarebbe lo stesso, sebbene possa essere meno chiaro nel primo caso.

**Select type:** definisce una collezione definita di altri *data types*. Le possibilità di scelta possono essere tra altre *entities* oppure una lista di valori che possono essere *simple data type*; come per l'*enumerated type*, può essere scelto solo un elemento tra quelli elencati. Il nome del *data type* è scritto all'interno di un rettangolo tratteggiato con una seconda linea verticale a sinistra e, similmente all'*enumerated type*, in IFC deve presentare il prefisso "Ifc" e il suffisso "SelectType".

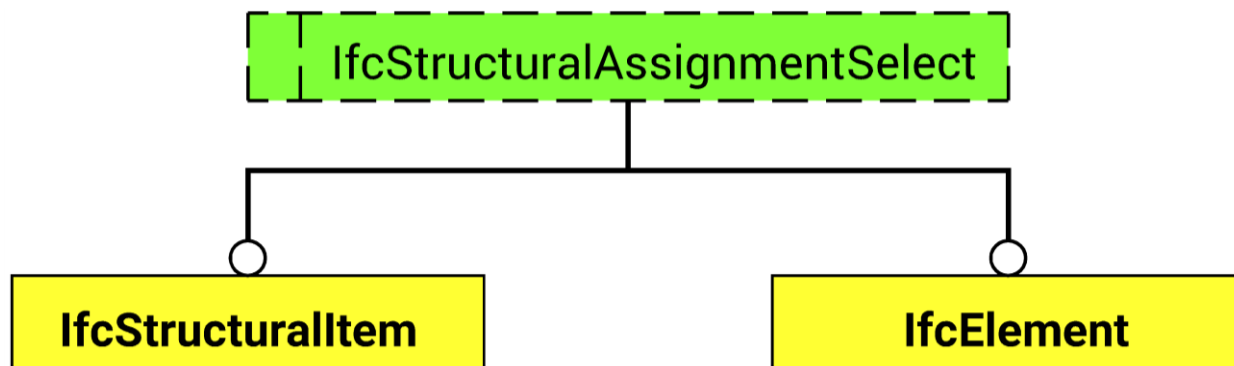


Figura 15 - EXPRESS-G. Select Type.

Ad esempio, sempre nello *Structural Analysis Domain*, il *select type IfcStructuralActivityAssignment.Select* restituisce una *entity* tra *IfcStructuralItem* e *IfcElement* per poter descrivere a quale dei due oggetti, un elemento strutturale o un generico elemento della costruzione, sia riferita una determinata attività strutturale, intesa come azione o reazione.

**Relationship:** esprime una dipendenza o un'interazione, come un *attributes*, tra due *entities*, attraverso una cardinalità, o *cardinality*, cioè che indica il numero di *objects* prese in considerazione di ciascuna delle due *entities* che la *relationship* collega. Esattamente come un *attribute*, la *relationship* è resa graficamente con una linea continua se è obbligatorio che sia specificata, o con una linea tratteggiata nel caso sia opzionale e con un cerchio per indicare la direzione principale; infine il nome è scritto nei pressi della linea e, spesso, è accompagnato da una lettera maiuscola e, fra due parentesi quadre, da due numeri separati dai due punti, oppure da un numero e un punto interrogativo. Nel caso in cui la *relationship* non abbia altro a fianco del nome significa che la *cardinalities* espressa è obbligatoriamente pari a 1. Se, invece, è presente altro serve a specificare se si tratti di un *Sets*, di una *Bags*, di una *Lists* oppure di un *Array*. La lettera maiuscola posta a fianco del nome serve a individuare quale tra questi tipi di *cardinalities* viene implementata, sulla base del fatto che viene utilizzata la lettera iniziale del tipo corrispondente. Sappiamo che i *Sets* sono collezioni disordinate di lunghezza variabile di *items* unici, mentre una *Bags* di *items* sì unici ma non necessari; una *List* è, invece, una collezione ordinata di lunghezza variabile di *items* unici non necessari, in cui sussiste la possibilità di limitare la lista ad una serie di oggetti unici prestabiliti; mentre un *Array* indice una collezione di dimensione fissa di *items* unici non necessari che possono essere accessibili da un indice. Dopo la lettera maiuscola si pone fra parentesi quadre, separati dai due punti, il numero minimo e massimo degli *items* consentiti, ad esempio “L[2:7]” indica una *List* di minimo due oggetti e massimo sette. Qualora il limite massimo non sia specificato, si utilizza il punto interrogativo, ad esempio “B[1:?]” indica una *Bag* di almeno un *item* ma senza un limite massimo; se si trova un solo numero, questo è sia il numero minimo di *items* che il massimo, ossia l'unico accettabile, ossia, “A[5]” indica un *Array* di obbligatoriamente 5 *items*; se, invece, i due numeri non sono accompagnati da parentesi né lettere, si suppone che si stia parlando di un *Set*, come, ad esempio, “[4:8]” indica un *Set* di almeno quattro *items* e massimo otto.

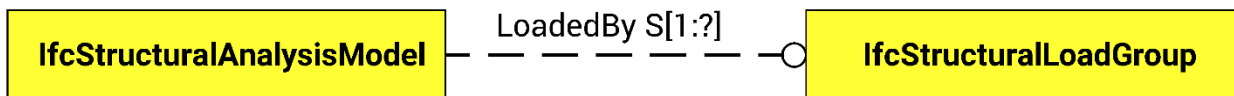
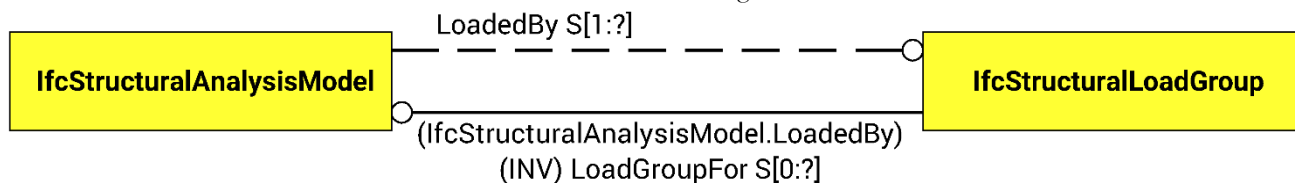


Figura 16 - EXPRESS-G. Attributes con cardinalità.

Nell'esempio qui sopra il modello di analisi strutturali, *IfcStructuralAnalysisModel*, che raccoglie tutti gli oggetti che compongono il modello stesso, attraverso un *Set* è messo in relazione con un gruppo di carichi, per formare così una combinazione di carichi. Dato che la collocazione è già definita e non dipende dalle altre, sono definiti in ordine sparsi, di qui il *Set*. Dato che il numero minimo di carichi necessario per una combinazione di carico è uno, di qui il numero minimo; mentre il massimo non è, giustamente definito perché può essere virtualmente infinito.

**Inverse relationship:** esprime un *relationship* che è necessario descrivere da entrambe le direzioni e, spesso, può essere dedotta dalla *relationship* originale. Viene indicata dal prefisso "(INV)" prima del nome; il tutto preceduto dal nome, se possibile, dell'*entity* a cui si riferisce seguita da un punto e il nome della *relationship* da cui può essere dedotta.

Figura 17 - EXPRESS-G. Attribute e inverse attribute.



Nell'esempio serve a definire a quale modello si riferisce il carico. Difatti per uno stesso edificio possono essere necessari più schematizzazioni strutturali e/o più porzioni strutturali e si dovrà distinguere esattamente a cosa è riferito il carico.

**Page reference:** poiché EXPRESS-G è una rappresentazione grafica pensata per essere compilata all'interno dello spazio di un foglio A4, sussiste il caso in cui il modello necessiti l'uso di più di una pagina. Per eseguire una qualsiasi *relationship*, *attribute* o *supertyping*, tra due definizioni che si trovano su pagine separate, sono stati previsti dei simboli di *page reference*. Dove una relazione attraversa il bordo di una pagina, si introduce un *reference ONTO ANOTHER PAGE*, che si collega ad un secondo *reference ONTO THIS PAGE* posto nella pagina di destinazione. Entrambi questi simboli consistono in un

rettangolo con i lati verticali sostituiti da semicerchi. *ONTO THIS PAGE* presenta, all'interno del simbolo il numero di pagina in cui ci si trova e, separato da una virgola, un indice numerico per catalogare i vari *references*; *ONTO ANOTHER PAGE* presenta invece, il numero della pagina di destinazione separato da una virgola dall'indice numerico del corrispettivo *ONTO*

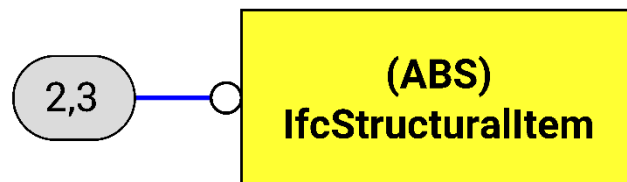


Figura 18 - EXPRESS-G. External reference, ONTO THIS PAGE.

*THIS PAGE* e il nome della *entity* a cui si rimanda. In pratica i due numeri devono coincidere. *ONTO THIS PAGE* è collegato attraverso una linea blu sottile all'oggetto che si vuole richiamare, mentre *ONTO ANOTHER PAGE* si relaziona agli oggetti che lo vorrebbe, a fini di intelligibilità grafica, più vicino, come fosse l'*item* stesso in questione, ovvero con normali linee nere spesse di *subtyping*, o linee sottili nere di *attribute* e di *relationship*. L'utilità del *page reference* è grande per gerarchie di *supertype* con numerosi *subtypes*: ad esempio in IFC l'*entity IfcElement* prevede la scelta tra 25 differenti *subtypes*.

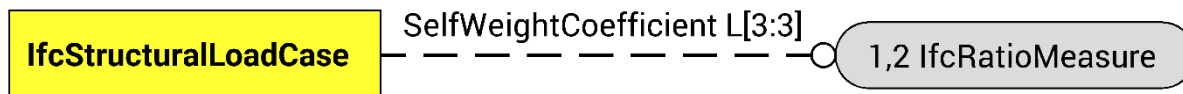


Figura 19 - EXPRESS-G. External reference, ONTO ANOTHER PAGE.

**Schema:** è una collezione, un insieme di elementi in una raccolta logica, per esempio un *Project Planning* che include tutte le definizioni di *entity data types* rilevanti al processo costruttivo di un edificio. Questo è utile anche nella definizione della gerarchia. In pratica sono raccolti tutti gli elementi comuni ad un certo argomento e utili per un certo scopo; sono separati graficamente, in EXPRESS-G, dagli altri, rimanendo, però, perfettamente collegati. Ognuno, forma, all'incirca, un modello di dati a sé stante, per lo più completo, per argomento.

**Schema reference:** dividendo tutto graficamente rimane necessario, alla stessa maniera in cui lo è il *page reference*, poter esplicitare il collegamento tra oggetti posti tra *schemas* differenti, rendendo noto

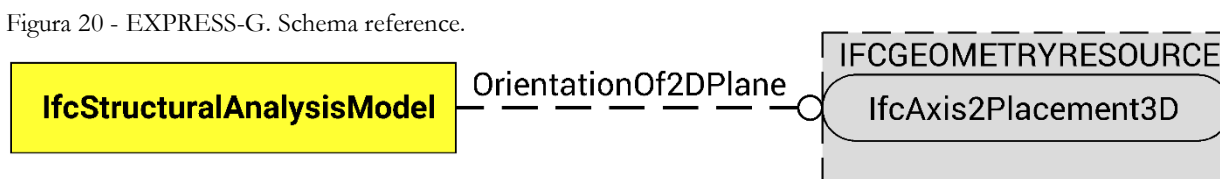


Figura 20 - EXPRESS-G. Schema reference.

dove recepire l'informazione in questione. La rappresentazione grafica è costituita da un rettangolo dai contorni tratteggiati al cui interno si trova un oggetto identico al *page reference*, cioè un rettangolo dai lati verticali sostituiti da semicerchi. All'interno del rettangolo tratteggiato è presente il nome dello *schema* verso cui ci si riferisce, mentre all'interno del rettangolo dai lati tondi viene posto il nome dell'*entity* che si richiama. Di nuovo ci si collega a questo oggetto attraverso le normali relazioni, siano di *subtyping*, *attributes* o *relationship*. Sotto un esempio, che richiama, all'interno dello *Structural Analysis Domain*, un *item* del *Geometry Resource*.

**Domain rule:** l'ultima notazione grafica è quella relativa ai vincoli imponibili agli *attributes*, per i quali è necessario il valore che riportano sia corretto. In EXPRESS, l'applicazione di questa regola è individuata dalla parola *WHERE*. Poiché questo si applica a uno o più *attributes* di una *entity*, quindi ogni *item* deve rispettare questa regola, graficamente questo è riportato con un asterisco all'interno dell'*entity* cui si applica questa regola.

## TERMINI, DEFINIZIONI E ABBREVIAZIONI

Ci sono poi alcuni termini, molti dei quali identici nel nome e nella costruzione, che vengono specificati nel capitolo 3 di IFC, che elenca termini, definizioni ed abbreviazioni che vengono utilizzate all'interno dello standard. Vengono passati brevemente in rassegna quelli utili a questo lavoro di tesi, poiché molti di questi ritorneranno spesso, volendo utilizzare una terminologia adeguata e precisa, necessaria per seguire una modellizzazione di dati standardizzata, con i suoi precisi nomi<sup>21</sup>.

Prima di vedere le definizioni della IFC 4 Addendum 2, il più recente, è meglio specificare altri termini dati per scontati o che dovremo mantenere in lingua inglese perché tradotti possono causare confusione:

-**Data**: traducibile all'incirca con “informazione”, ma più precisamente indica una descrizione elementare codificata di oggetto che porta alla conoscenza di un'informazione;

-**Data type**: indica una classificazione che identifica uno dei diversi tipi di *data* e che ne determina il possibile valore;

-**Data item**: termine generico usato per indicare *types*, *entities*, *rules* e *functions*, cioè tutti quegli elementi che costituiscono l'ossatura di IFC, verranno meglio spiegati in seguito;

-**Reference data**: definisce un set di valori permessi, da essere usati per le altre *data*;

-**Data model**: è invece un modello che organizza gli elementi di un *data* e specificando le sue relazioni con un altro *data* o una *property*.

---

21 - buildingSMART, “IFC4 - Add.2; 3 Terms, definitions and abbreviated terms”

I termini usati per comporre lo schema IFC sono:

-**Actor**: persona indipendente o che agisce per un'organizzazione, può essere riferito anche all'organizzazione stesso;

-**Entity**: parte di informazione definita da attributi e vincoli comuni, così come definito dalla ISO 10303-11; è simile al termine *class* nei comuni linguaggi di programmazione, ma descrive solo strutture di dati, non comportamenti;

-**Type**: informazione basilare di un costrutto definita da una funzione fondamentale, da un'enumerazione o da un gruppo di *entities*. Un'enumerazione è un costrutto che permette che il valore di un *attribute* sia uno tra vari valori predefiniti, ciascuno con un suo proprio nome;

-**Attribute**: unità di un'informazione di una *entity*, definita da un particolare *type* o da un riferimento a una specifica *entity*. Ci sono tre tipi di *attributes*: *direct*, *inverse* e *derived*. Il **direct attribute**, cui spesso ci riferiremo semplicemente come *attribute*, serve per raccogliere collezioni o scale di valori definite nell'ISO 10303-11 e divise in *Set*, *List*, ed *Array*. I primi intende collezioni disordinate di elementi unici, il secondo collezioni ordinate di elementi unici, mentre gli ultimi sono collezioni ordinate di elementi occasionali. Sono possibili, in EXPRESS, anche i *Bags*, collezioni disordinate di elementi occasionali. Gli **inverse attributes** sono parti di informazione che definiscono le *queries* (domande) per ottenere un'informazione relativa e far rispettare l'integrità della fonte. Sono, come da nome, collegati a un *direct attribute* di cui esprimono l'inverso concettuale per rendere precisa l'informazione. Infine i **derived attributes**, poco utilizzati in IFC, che descrivono un aspetto di un'informazione ottenuta da altri attributi usando un'espressione già definita nello schema;

-**Constraint**: restrizioni, vincoli, con lo scopo di limitare i valori di un *attribute*; o, comunque, di rendere corretta un'informazione che può essere eccessivamente ampia;

-**Model**: un insieme di informazioni governate dalla struttura di uno schema sottostante, per poter dare certe informazioni richieste;

-**Model view**: è un sottoinsieme di uno schema, che rappresenta la struttura di dati richiesta per soddisfare le richieste di dati stessi all'interno di uno o più scenari di scambio d'informazioni;



- Classification:** categorizzazione, l'atto di distribuire oggetti in *classes* o categorie dello stesso type;
- Element:** è un prodotto fisico tangibile, può essere descritto con la rappresentazione della sua forma, del materiale o di altre sue *property*;
- Property:** sono parte di un'informazione e sono individuati esattamente da una particolare *instance* di una *entity*;
- Instance:** se riferito ad una *entity* è traducibile con il termine "oggetto", è, cioè, un esemplare unico di una classe, separato da altre *instances* con le quali però comunica. In pratica è un preciso oggetto del tipo descritto dalla *entity* e più o meno compiutamente definito. Generalmente si considerano oggetti tutte quelle *entities* dotate di una propria identità. È un termine molto complesso e spesso è erroneamente tradotto in italiano con istanza;
- Object:** qualsiasi cosa percettibile e immaginabile come una entità distinta, non per forza concreta e tangibile. Un **object occurrence** è una caratteristica di un oggetto come individuale, mentre un **concept type** sono caratteristiche comuni condivise da vari *object occurrences*;
- Schema:** è la definizione di una struttura che organizza lo spazio di archiviazione, lo scambio e la condivisione dei *data*;
- Concept:** regole sull'utilizzo di sotto insiemi di uno *schema*, identificato come un *concept template* per poter sfruttare certe funzionalità all'interno di un *concept root* contenuto in una *model view*. Il **concept template** è la specificazione di sotto struttura di uno *schema* che permette una certa singola funzionalità. **Concept root:** un'*entity* di uno *schema* usata per assegnare i *concepts* che descrivono una certa funzionalità;
- Process:** un object occurrence collocate nel tempo; indica il "quando";
- Product:** oggetto fisico o concettuale che si trova nello spazio;
- Quantity:** misura di una grandezza basata sullo scopo, specialmente lunghezza, area, volume, peso, conteggio o tempo;
- Relationship:** unità dell'informazione descritta dall'interazione tra oggetti o elementi;

-**Representation:** unità dell'informazione che specifica come un object viene mostrato, sia nella forma fisica che nella forma topologica;

-**Resource:** *entity* con una limitata disponibilità, come può essere un materiale o un equipaggiamento. Può esistere solo come informazione di specificazione di altre *entity* indipendenti. Ad esempio una misura non può esistere se non riferita a un qualcosa da misurare: *IfcLengthMeasure* dovrà essere obbligatoriamente definita in relazione a un qualcosa, ad esempio all'*attribute* che specifica altezza o larghezza netta di passaggio di una porta;

-**Library:** catalogo, database o contenitore di dati che è importante per l'informazione all'interno di un set di data.

Tutto ciò è specificato all'interno del capitolo 3 di IFC 4 Addendum 2<sup>22</sup>.

---

22 - buildingSMART, "IFC4 - Add.2; 3 Terms, definitions and abbreviated terms"



## **3B. IFC oggi**

## SCOPO

Tornando ad IFC e continuando, alla voce “scope” della versione IFC4 addendum 2<sup>23</sup>, la più recente, scopriamo che viene dichiarato quanto segue:

«The Industry Foundation Classes, IFC, are an open international standard for Building Information Model (BIM) data that are exchanged and shared among software applications used by the various participants in the construction or facility management industry sector. The standard includes definitions that cover data required for buildings over their life cycle. This release, and upcoming releases, extend the scope to include data definitions for infrastructure assets over their life cycle as well.

The Industry Foundation Classes specify a data schema and an exchange file format structure. The data schema is defined in

EXPRESS data specification language, defined in ISO 10303-11,

XML Schema definition language (XSD), defined in XML Schema W3C Recommendation,

whereas the EXPRESS schema definition is the source and the XML schema definition is generated from the EXPRESS schema according to the mapping rules defined in ISO 10303-28. The exchange file formats for exchanging and sharing data according to the conceptual schema are

Clear text encoding of the exchange structure, defined in ISO 10303-21,

Extensible Markup Language (XML), defined in XML W3C Recommendation.

Alternative exchange file formats may be used if they conform to the data schemas.

---

23 - buildingSMART, “IFC4 - Add.2; 1 Scope”

This release of IFC consists of the data schemas, represented as an EXPRESS schema and an XML schema, and reference data, represented as definitions of property and quantity names, and formal and informative descriptions.

A subset of the data schema and referenced data is referred to as a Model View Definition (MVD). A particular MVD is defined to support one or many recognized workflows in the construction and facility management industry sector. Each workflow identifies data exchange requirements for software applications. Conforming software applications need to identify the model view definition they conform to. »

Ovvero, molto chiaramente, che IFC si pone come uno standard internazionale aperto per le informazioni BIM che vengono scambiate e condivise tra le varie applicazioni utilizzate dalle diverse figure professionali che partecipano alla costruzione o alla gestione di un servizio. Lo standard include definizioni che coprono informazioni richieste per gli edifici in tutto il loro ciclo di vita. Ciò non significa altro che si punta a realizzare una standardizzazione completa delle informazioni necessarie a descrivere un edificio, dal primo progetto di massima con la definizione concettuale di bisogni, alla manutenzione e recupero o smaltimento di un organismo edilizio, comprendendo così l'intero ciclo di vita della struttura. Questo si traduce in requisiti chiari e precisi da trasmettere alle software house per poter sviluppare i loro prodotti, perché così possa essere assicurata l'interoperabilità dell'informazione, per uniformare i flussi di lavoro e facilitare l'automazione.

In pratica, quindi, lo standard IFC specifica uno schema di informazioni definito con il linguaggio di modellazione di dati chiamato EXPRESS, definito nella norma ISO 10303-11; alternativamente può essere definito secondo il metalinguaggio XML.

Alla definizione attuale IFC consiste nei diagrammi di dati, rappresentati secondo diagrammi EXPRESS e XML, nei riferimenti di dati, rappresentati come definizioni di proprietà e quantità, e nelle descrizioni formali e informative.

Sottoinsiemi di questi diagrammi di dati concorrono a definire una *Model View Definition*, abbreviato in MVD. Una particolare MVD viene definita a supporto di uno, o più, flussi di lavoro nella costruzione o nella gestione di un servizio; ogni flusso di lavoro identifica una richiesta di informazioni da

scambiare per certi software, che, se conformi allo standard, necessitano che sia identificata la MVD di riferimento.

Tutto ciò rende chiaro come IFC abbia come unico vero grande obiettivo l'interoperabilità, cioè garantire la correttezza dello scambio di informazioni tra diversi software attraverso un sistema di classificazione e descrizione di ogni elemento del complesso edilizio, dei suoi attributi fisici e astratti. Questo si concretizza in un file di formato “.ifc” che i software conformi sono in grado di aprire e creare, senza perdita di dati collegati possibilmente.

Come garanzia della qualità di IFC, buildingSMART collabora attivamente con l'ente ISO che ha registrato lo standard con la normativa ISO 16739:2013 *“Industry Foundation Classes for data sharing in the construction and facility management industries”*.

In ogni caso, l'associazione elenca tutto ciò che è attualmente previsto e descritto in IFC e tutto ciò che manca ed è, preferibilmente, da implementare.

Le definizioni del formato di interscambio BIM richieste durante le fasi dei cicli di vita degli edifici:

Dimostrazione del bisogno;

Concezione del bisogno;

Abbozzo di fattibilità;

Studio di fattibilità sostanziale e abbozzo di bozza di possibilità finanziaria;

Bozza di design concettuale;

Design concettuale finale;

Design coordinato;

Appalti e possibilità finanziaria completa;

Produzione dell'informazione;

Costruzione;

Operatività e mantenimento.

Le definizioni del formato di interscambio BIM richieste dalle varie discipline coinvolte nelle varie fasi dei cicli di vita:

Architettura;

Manutenzione dell'edificio;

Ingegneria strutturale;

Appalti;

Progettazione edilizia;

Gestione dell'infrastruttura edilizia;

Gestione del progetto;

Gestione delle richieste della committenza;

Autorità edilizia per i permessi e le approvazioni.

La definizione del formato di interscambio BIM include:

Progetto delle strutture;

Componenti fisiche;

Componenti spaziali;

Elementi per le analisi;

Processi;

Risorse;

Controlli;

Attori del processo;



Definizione del contesto.

Rimane, al momento, fuori dall'ultima versione di IFC:

La definizione del formato di scambio al di fuori del dominio delle costruzioni e della gestione dell'infrastruttura;

Il progetto dell'intera struttura e il collasso dei componenti delle strutture al di fuori dell'ingegneria edile, ma fornendo una linea guida per l'ingegneria civile, da estendersi nelle successive versioni;

Aspetti comportamentali di componenti e di altri elementi dell'informazione.

Questo secondo ammissione della stessa organizzazione. Sono verosimilmente molti di più gli aspetti ancora da descrivere.

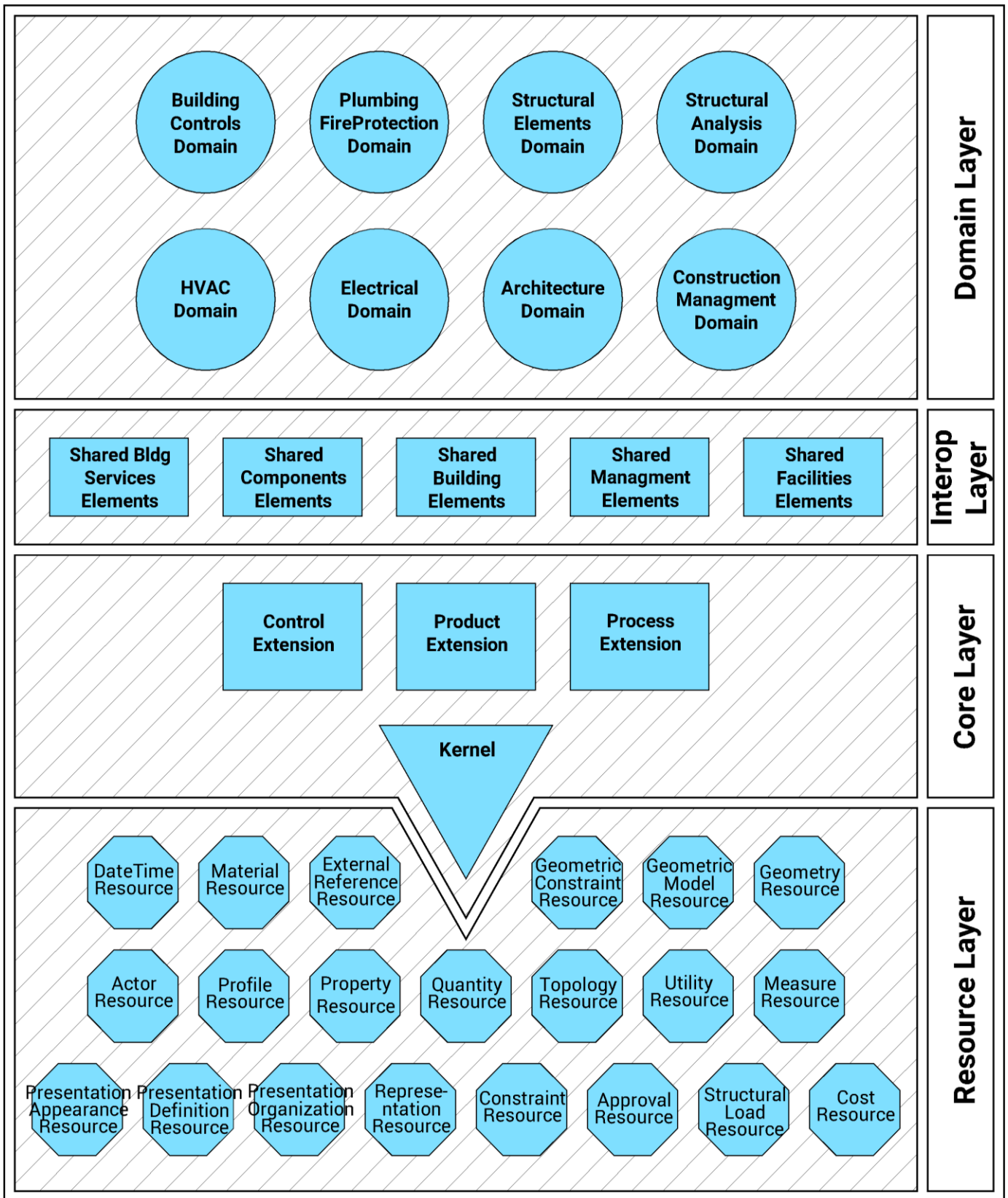


Figura 21 - IFC. Definizione dei livelli gerarchici e dei domini.

## DOMINI IFC<sup>24</sup>

**-Core data schemas:** I *core data schemas* che costituiscono il *Core layer*, traducibile come livello del nucleo, stabiliscono il livello più generale nell'architettura di IFC. Le *entities* definite in questo layer possono essere riferite a e specialistiche di tutte le *entities* gerarchicamente superiori. Il *Core Layer* fornisce strutture base, relazioni fondamentali e concetti comuni per tutti i successivi perfezionamenti in specifici aspetti dei vari modelli. Tutte le *entities* definite in questo livello derivano da *IfcRoot*, con un'identificazione ID unica, un nome, una descrizione e informazioni sullo storico dei proprietari.

Presenta quattro *data schemas*, il principale è il *Kernel schema*, seguito dal *Control Extension schema*, il *Process Extension schema* e il *Product Extension schema*. Sono chiamati *IfcKernel*, *IfcControlExtension*, *IfcProcessExtension* e *IfcProductExtension*.

**-Shared element data schemas:** I *shared data schemas* che costituiscono l'*Interoperability layer*, circa livello di interoperabilità, descrivono tutte le *entities* specifiche per un prodotto generale, per un process o per un *resource specialization* che vengono utilizzati tra diverse discipline. Questo livello è più alto del precedente. In pratica questi *schemas* contengono specializzazioni intermedie di *entities* e forniscono *objects* più specialistici e *relationships* più specialistici condivisi tra più *domains*. Al momento sono presenti cinque *data schemas*: lo *Shared Building Elements schema*, lo *Shared Building Service Elements schema*, lo *Shared Component Elements schema*, lo *Shared Facilities Elements schema* e lo *Shared Management Elements schema*. Sono chiamati, rispettivamente, *IfcSharedBldgElements*, *IfcSharedBldgServiceElements*, *IfcSharedComponentElements*, *IfcSharedFacilitiesElements*, *IfcSharedMgmtElements*.

**-Domain specific data schemas:** I *shared data schemas* all'interno del *Domain layer*, traducibile come livello dei domini contengono le specificazioni finali delle *entities*, precipuamente appartenenti ad una certa disciplina, siano esse descrittive prodotti, processi o risorse. È il livello più alto di tutti e tutto ciò che appartiene a questo livello non può essere preso come riferimento da nessun altro livello, proprio perché questo organizza le definizioni secondo le varie discipline. Oggi come oggi sono

---

24 - buildingSMART, "IFC4 - Add.2; Introduction"

presenti otto *data schemas*: l'*Architecture Domain data schema*, il *Building Controls Domain data schema*, il *Construction Management Domain data schema*, l'*Electrical Domain data schema*, il *HVAC Domain data schema*, il *Plumbing Fire Protection Domain data schema*, lo *Structural Analysis Domain data schema* e lo *Structural Elements Domain data schema*. Sono, rispettivamente *IfcArchitectureDomain*, *IfcBuildingControlsDomain*, *IfcConstructionMgmtDomain*, *IfcElectricalDomain*, *IfcHvacDomain*, *IfcPlumbingFireProtectionDomain*, *IfcStructuralAnalysisDomain*, *IfcStructuralElementsDomain*.

**-Resource definition data schemas:** I *resource definition data schemas*, che costituiscono il livello più basso tra tutti, quello del *Resource layer*, traducibile come livello delle risorse, consistono in strutture di dati di supporto. Infatti *entities* e *types* di questo layer possono essere fatti riferire a ogni *entity* nei layer superiori; anzi è più corretto dire che questi oggetti debbono far riferimento ad altri livelli, infatti le strutture di dati delle risorse non possono esistere indipendentemente, ma possono esistere solo se fatti riferire, direttamente o indirettamente, a una o più *entities* derivanti da *IfcRoot*; in pratica devono essere collegati con il livello del nucleo. Proprio perché si tratta di risorse, non esiste per questo layer il concetto di identità esclusiva, difatti più oggetti possono essere riferiti ad una stessa *instance* ma senza, necessariamente, che questo implichi una *relationship* tra le parti. L'esempio che buildingSMART porta è quello, elementare delle polilinee, descritte con *IfcPolyline*, che possono condividere geometricamente uno stesso punto, necessariamente descritto con *IfcCartesianPoint*, ma che possono arbitrariamente fare riferimento o meno a due diverse *instances* per descrivere questo punto; si suggerisce, in ogni caso, per minimizzare la dimensione del file, di condividere le *instances* del *resource layer*. Alla data attuale si contano ventisei definizioni generiche per descrivere ogni tipo di prodotto e le sue proprietà base, come la geometria, i materiali o i costi.

Questi *data schemas* sono l'*Actor Resource data schema*, l'*Approval Resource data schema*, il *Constraint Resource data schema*, il *Cost Resource data schema*, il *Data Time Resource data schema*, l'*External Reference Resource data schema*, il *Geometric Constraint Resource data schema*, il *Geometric Model Resource data schema*, il *Geometry Resource data schema*, il *Material Resource data schema*, il *Measure Resource data schema*, il *Presentation Appearance Resource data schema*, il *Presentation Definition Resource data schema*, il *Presentation Organization Resource data schema*, il *Profile Resource data schema*, il *Property Resource data schema*, il *Quantity Resource data schema*, il *Representation Resource data schema*, lo *Structural Load Resource data schema*, il *Topology Resource data schema* e l'*Utility Resource*

*data schema*. Questi rispettivamente, sono chiamati *IfcActorResource*, *IfcApprovalResource*, *IfcConstraintResource*, *IfcCostResource*, *IfcDateTimeResource*, *IfcExternalReferenceResource*, *IfcGeometricConstraintResource*, *IfcGeometricModelResource*, *IfcGeometryResource*, *IfcMaterialResource*, *IfcMeasureResource*, *IfcPresentationAppearanceResource*, *IfcPresentationDefinitionResource*, *IfcPresentationOrganizationResource*, *IfcProfileResource*, *IfcPropertyResource*, *IfcQuantityResource*, *IfcRepresentationResource*, *IfcStructuralLoadResource*, *IfcTopologyResource* e, infine, *IfcUtilityResource*.

# **3C. structural analysis domain**

## PREMESSA

Il *data schema* dedicato alle analisi strutturali è, appunto lo *Structural Analysis Domain*<sup>25</sup> che descrive il modello analitico della struttura, con classi che individuano connessioni tra elementi, elementi mono o bidimensionali, o azioni e reazioni della struttura, attraverso *absolute supertypes* come *IfcStructuralConnection*, *IfcStructuralMember* e *IfcStructuralActivity*, il quale contiene *IfcStructuralAction* e *IfcStructuralReaction*. Lo *schema* invece dello *Structural Load Resource* definisce dati di supporto per le entità principali, contenute nello *Structural Analysis Domain*; in pratica è uno schema ausiliario. Infatti presenta la possibilità di descrivere i tipi di carico, le loro combinazioni, i risultati del sistema strutturale, o condizioni al contorno, come le specificazioni sulle connessioni, attraverso, ad esempio, *IfcFailureConnectionCondition*, che contiene le condizioni di rottura di una giunzione qualunque tra due elementi strutturali.

Nella premessa, lo *Structural Analysis Domain* è così descritto:

«The *IfcStructuralAnalysisDomain* describes the structural analysis model in order to tightly integrate the structural engineering domain. It reuses the existing building element and spatial structure element definition and associates the structural assumptions to it. The focus is to ensure that structural engineering information is captured and made visible to other related domains.

The following features are in scope:

Defining planar and/or spatial structural analysis models which can be used by structural analysis applications.

Included are, basically:

Straight or curved structural curve elements, planar or curved structural surface elements.

Point, curve, and surface connections and supports.

---

25 - buildingSMART, “IFC4 - Add.2; 7.7 IfcStructuralAnalysisDomain”.

Specification of loadings including point, curve, surface loads, temperature loads, their assignment to load groups, load cases and load combinations.

Specification of different structural analysis models in order to describe different aspects or parts of the building. Furthermore, dependencies between these models can be stored in the model for further use.

Analysis results defined by forces and displacements.

Currently not in scope are:

Dynamic analysis

Description of prestressed loads

Finite element topology

Detailed results in finite element meshes as well as stresses and strains in the structural elements. »

Ossia si dice che lo schema di dominio strutturale si propone di descrivere, appunto, il modello dell'analisi strutturale, per poter assolutamente completare la disciplina dell'ingegneria delle strutture. Si appoggia, appunto, a elementi dell'edificio e a strutture spaziali esistenti nel modello e vi associa delle ipotesi con l'obiettivo di assicurare che le informazioni pertinenti alla disciplina siano presenti e visibili anche agli altri domini collegati.

La stessa buildingSMART passa poi in rassegna le caratteristiche già presenti e quelle assenti nello standard IFC. Ossia come sia possibile definire e utilizzare in appositi software modelli per analisi strutturali nel piano e nello spazio; siano essi basati su linee e superfici, curve o meno; collegati e vincolati in punti, curve o superfici; sostenenti carichi puntuali, distribuiti su curve o su superfici, azioni date dalle temperature, nonché i loro gruppi, i loro casi e le loro combinazioni. Si possono descrivere compiutamente diversi modelli per analisi, cosicché sia possibile descrivere diversi aspetti o parti dell'edificio, considerando che anche le dipendenze tra questi modelli possono essere immagazzinate per eventuali scopi. Infine sono, ovviamente, inclusi i risultati delle analisi, siano espresse in forze e tensioni o in spostamento.



Vengono, appunto, anche riportate quali siano le mancanze dello stesso dominio, sottintendendo come questi siano i prossimi punti che si intendono implementare. Viene riferito che non sono ancora previste all'interno dello standard le analisi dinamiche e tutto ciò che vi si collega; i carichi dovuti alla precompressione; la possibilità di descrivere un elemento finito e, infine, i risultati dettagliati di una mesh di elementi finiti, siano essi espressi in sforzi o in deformazioni in un elemento strutturale.

Per poter capire come effettivamente risolvere le mancanze di cui sopra, ovvero le analisi sismiche e la definizione completa di una mesh agli elementi finiti, provvediamo a passare, quindi, in rassegna l'intero *data schema* per poter capire quali *types* ed *entities* siano effettivamente presenti e cosa descrivono, per confrontarci con la necessità informativa che i problemi che vogliamo affrontare richiedono.

Lo *Structural Analysis Domain* è organizzato concettualmente in varie "classi" atte a descrivere gli oggetti del modello analitico, partendo da *IfcStructuralItem*, oppure azioni e reazioni della struttura, generate a partire da *IfcStructuralActivity*, o ancora altri dettagli del modello di analisi, come condizioni e combinazioni di carico e gruppi di risultati attraverso *IfcStructuralAnalysisModel*. È infine presente una classe di relazioni atte a collegare un elemento alla corrispondente attività.

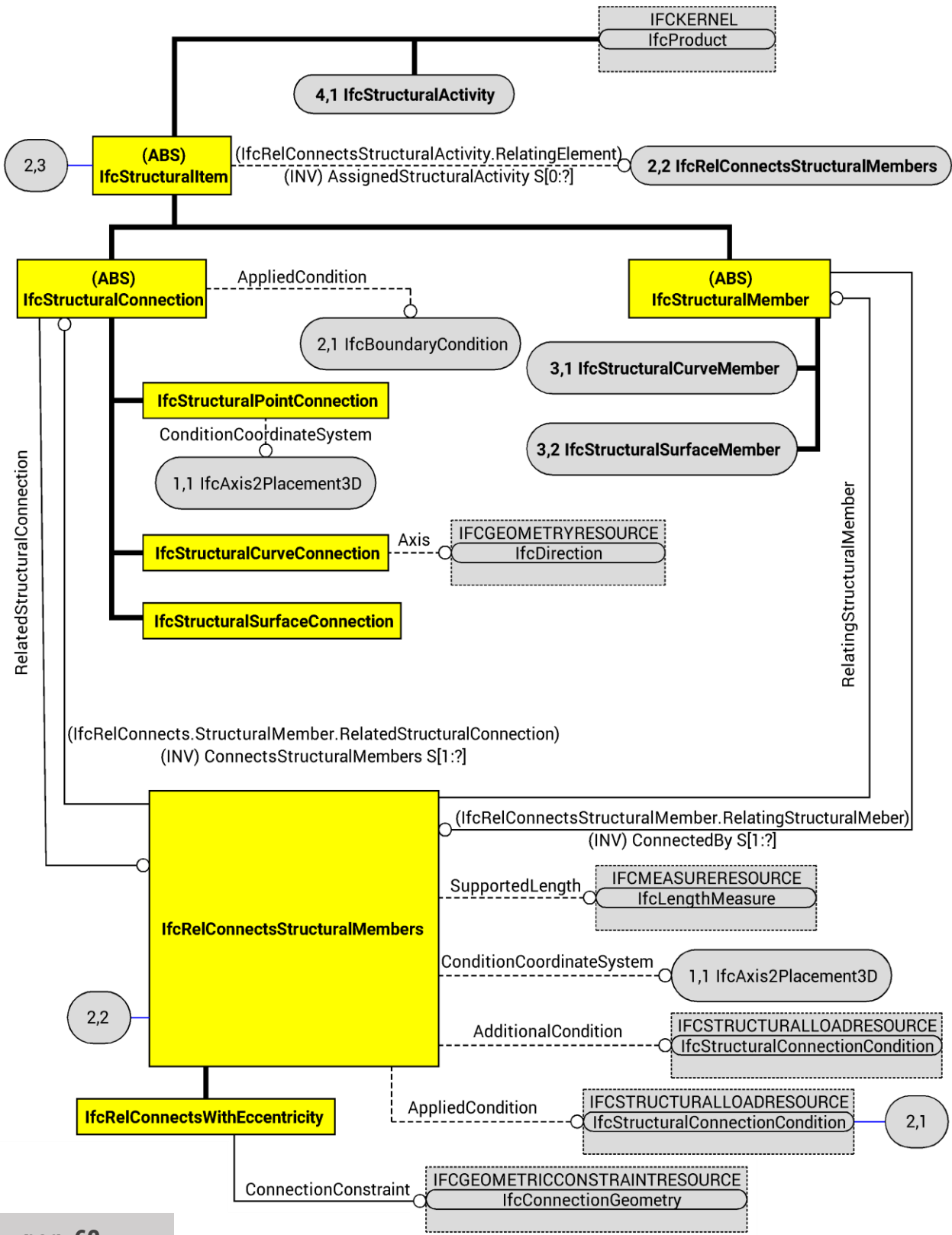


Figura 22 - IFC SAD. Elementi e connessioni strutturali e loro relazioni.

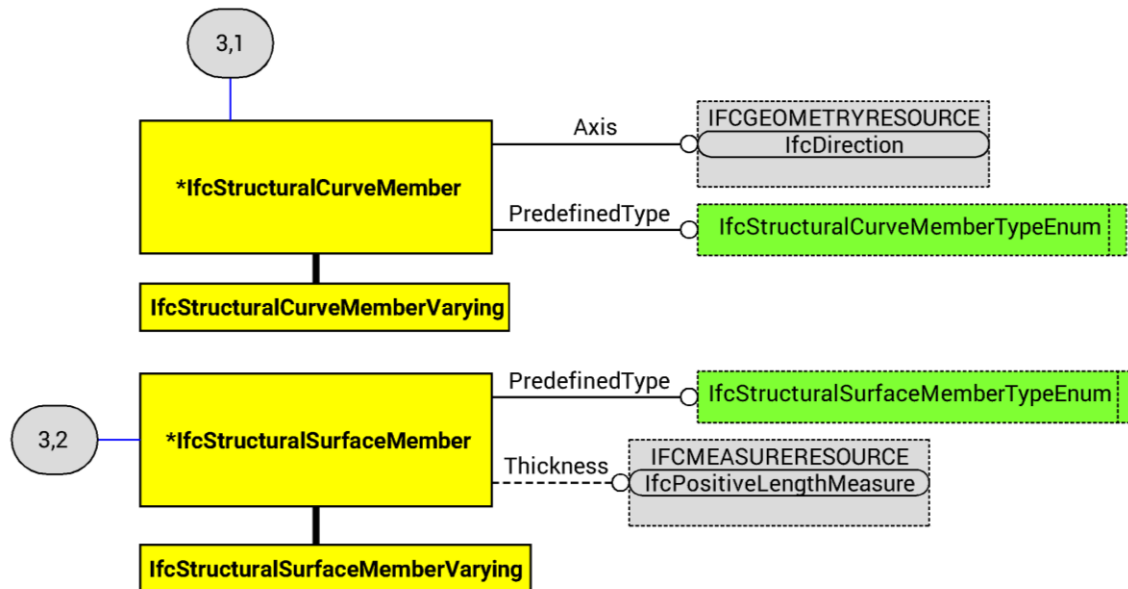


Figura 23 - IFC SAD. Elementi strutturali mono e bidimensionali.

## IFC STRUCTURAL ITEM

-La *absolute entity* chiamata **IfcStructuralItem**<sup>26</sup> è una generalizzazione di un elemento o una connessione strutturali, dimodoché possa contenere ogni schematizzazione degli elementi del modello e, poi, essere collegata ad essi attraverso la *instance IfcRelAssignsToProduct*. Viene anche messa in relazione alle attività strutturali, azioni e reazioni, attraverso *IfcRelConnectsStructuralActivity*.

In ogni caso, gli elementi strutturali e gli oggetti di connessioni strutturali sono messi in relazione attraverso *IfcRelConnectsStructuralMember* che definisce le proprietà necessarie, come condizioni di vincoli. Per fare ciò si appoggia anche allo *StructuralLoadResource*.

È importante sottolineare che queste *entities* hanno due diversi sistemi di coordinate, il primo con le cosiddette coordinate globali, il secondo con quello locali. Il sistema di coordinate globali è il sistema di coordinate condiviso da tutti gli *items* e le attività raggruppati nel comune *IfcStructuralAnalysisModel*. Questo sistema di coordinate è stabilito automaticamente, attraverso *IfcProductObjectPlacement*. Infine,

26 - buildingSMART, "IFC4 - Add.2; 7.7.3.13 IfcStructuralItem".

questo sistema non è necessariamente lo stesso presente in *IfcProject*. Invece, il sistema di coordinate locali è un sistema di coordinate per un *item* strutturale, sia esso una connessione o un membro. Questo sistema è stabilito da una *attribute* di *Representation*, ereditato da *IfcProduct*, in unione con ulteriori specifiche d'uso e *attributes* dei *subtypes* di *IfcStructuralItem*. Gli *item* di *Representation* nella rappresentazione topologica sono sempre dati all'interno di *ObjectPlacement* cioè con le cosiddette coordinate globali, nel rispetto di quelle date da *IfcStructuralAnalysisDomain*, al quale appartiene. Per quanto riguarda le coordinate locali e la loro specificazione, si rimanda ai singoli *subtypes*.

Le *instances* classificate con *IfcStructuralItem* devono avere una rappresentazione topologica che includa il posizionamento e la rappresentazione del prodotto. Questa, deve essere assegnata con *IfcProductRepresentation*, appartenente al *data schema* *IfcRepresentationResource* attraverso un'*attribute* di *Representation*, al *type* *IfcTopologyRepresentation*.

Il posizionamento locale è definito nel *supertype* *IfcProduct* ed è definito da *IfcLocalPlacement*, che stabilisce un sistema di coordinate globale comune a tutti gli *item* e le *activities* di *IfcStructuralAnalysisModel*.

La rappresentazione topologica viene assegnata con una *instance* del *subtype* *IfcTopologicalRepresentationItem*, singolo *item* di *IfcTopologyRepresentationItem*. Questa viene definita in base alla *subentity* a cui si riferisce, per cui si rimanda al tipo di connessione e membro.

Il riferimento di punti, curve e superfici definisce, più o meno completamente, il sistema di coordinate locali dell'oggetto strutturale rappresentato, secondo le regole definite di volta in volta per l'*item* in questione. In tutti i casi le coordinate x, y e z seguono una disposizione che segue la regola della mano destra.

Si impone che l'*ObjectPlacements* di tutti gli oggetti strutturale raggruppati nella stessa *instance* di *IfcStructuralAnalysisModel* si riferiscano alla stessa *instance* di *IfcObjectPlacement*. Questo per ottenere una rappresentazione topologica coerente, poiché questa si intende condivida vertici e spigoli, dovendo così avere lo stesso posizionamento. Un *item* strutturale potrebbe essere raggruppato in più di un modello di analisi, in questo caso tutti questi modelli devono usare la stessa *instance* di *IfcObjectPlacement*.

-**IfcStructuralMember** è un' *abstract entity* che può essere una sola tra due *subentities* atte a descrivere una curva o una superficie strutturale, secondo il modello analitico. Attraverso *IfcRelConnectsStructuralMember* che stabilisce anche una *inverse relationship* con la classe le connessioni strutturali definite per il relativo membro strutturale, si esplica, appunto, la relazione tra i membri e le rispettive connessioni.

-La prima *entity* in questione, che può descrivere un elemento strutturale è **IfcStructuralCurveMember** su cui è stata imposta una regola di *Domain Rule*, la regola WHERE. In altre parole è stato assegnato un vincolo e la nostra *entity* non può non avere un valore che le arriva dal *type IfcStructuralCurveMemberTypeEnum* che ne descrive la tipologia di elemento strutturale. Rimane in generale capace di descrivere le schematizzazioni di elementi monodimensionali del modello analitico, siano essi travi, colonne, bielle, e altro; di più può descriverli sia che essi siano curvi, sia che siano rettilinei e sia che profilo e materiale siano costanti o meno lungo l'asse. Viene inoltre fornita una regola secondo la quale assegnare gli assi x, y e z dell'elemento. Che, cioè, il sistema di coordinate locali sia stabilito dalla curva in questione e dalla sua rappresentazione topologica, che porta con sé l'*attribute Axis*. L'asse locale x è parallelo alla tangente alla curva, mentre l'asse locale z è posizionato sulla superficie che si crea estendendo l'asse lungo la curva di riferimento ed è diretto secondo l'asse stesso. L'asse y è, infine, definito in modo da formare una tripletta di assi cartesiani che seguono la regola della mano destra. Si deve preferire ed è di solito possibile che più *instances* di *IfcStructuralCurveConnection* e di *IfcStructuralCurveMember* condividano la *instance IfcDirection* e l'*attribute Axis*.

È a questa *entity* che si applica il materiale. Ossia, attraverso *IfcMaterialProfileSetUsage* e *IfcRelAssociatesMaterialRelatingMaterial*, viene descritto il materiale dell'elemento strutturale; viene definita inoltre un' *inverse relationship*. Per rappresentare un profilo composto ci si può riferire a vari *IfcMaterialProfiles* all'interno di *IfcMaterialProfileSet*, che è referenziato da *IfcMaterialProfileSetUsage*; per elementi rastremati viene utilizzato il *subtype IfcMaterialProfileSetUsageDual*, poiché specifica separatamente all'inizio e alla fine dell'elemento rastremato *IfcMaterialProfileSet*. Il materiale in ogni set è specificato al minimo da un nome che corrisponde con un solo materiale strutturale specifico all'interno di un elenco standardizzato. Alternativamente si può dare significato alle proprietà del

materiale attraverso *IfcMechanicalMaterialProperties* e attraverso *IfcExtendedMaterialProperties*. Similmente anche il profilo, descritto da *IfcProfileDef*, è scelto all'interno di un elenco di standardizzato di profili strutturali, è scelto attraverso il set *IfcMaterialProfile*. Deve, però essere prevista una *reference* esterna alla risorsa per poter provvedere alla designazione del profilo; alternativamente o addizionalmente, si può provvedere una esplicita geometria del profilo usando i rispettivi *subtypes* di *IfcProfileDef* e alla proprietà strutturali di questo profilo attraverso i *subtypes* di *IfcProfileProperties*. Un *IfcProfileDef* è un oggetto geometria bidimensionale con un sistema di coordinate  $x_p, y_p$ , la cui origine va a inserirsi sulla curva di riferimento dell'elemento, in modo che questo sistema di coordinate si ritrovi parallelo e nella stessa direzione del sistema di riferimento locale  $y,z$ ; così facendo si può inserire il profilo sulla curva. Sarà necessario, se il profilo deve essere inserito in modo che il centro delle proprietà meccanica coincida con l'asse dell'elemento, che l'origine  $x_p, y_p$  coincida con il baricentro della sezione e, nel caso in cui questo avvenga si deve impostare l'attribute *CardinalPoint* a 10, altrimenti il profilo sarà inserito eccentricamente ad un diverso punto cardinale. Sarà comunque possibile inserire un'altra eccentricità nel modello. Nel caso in cui venga utilizzato *IfcParameterizedProfileDef*, che cioè il profilo abbia un solo asse di simmetria o sia asimmetrico, si richiede un'esplicita traslazione dell'origine  $x_p, y_p$  attraverso l'attribute *Position*. Un'altra eccentricità del modello è disponibile, indipendentemente dall'eccentricità del profilo, in quanto la curva di riferimento dell'elemento può essere posizionata eccentricamente rispetto ai punti di connessione specificati con *IfcStructuralPointConnection*. La relazione di connessione è, allora, stabilita attraverso *IfcRelConnectsWithEccentricity*. Se una delle due o entrambe le eccentricità viene sfruttata si richiede che vengano rispettati gli obblighi di informazioni e gli accordi locali.

Si ricorda inoltre che, data la diversa convenzione in meccanica delle strutture, la nomenclatura degli assi di *IfcStructuralCurveMember* differirà da quella degli assi degli elementi del modello dell'edificio, come, ad esempio, in *IfcBeamStandardCase*: nel primo caso l'asse principale sarà chiamato  $x$ , nel secondo caso  $z$ ; quindi a  $x, y$  e  $z$  corrisponderanno  $z, x$  e  $y$ .

La curva di riferimento dell'elemento sarà rappresentata attraverso l'instance *IfcEdge* o un suo *Subtype*. Difatti, sia gli elementi che le connessioni monodimensionali devono essere rappresentate da o *IfcOrientedEdge*, o *IfcEdgeCurve*, oppure *IfcEdge*. La curva a cui queste *entities* si riferiscono è la curva di

riferimento dell'*item* strutturale descritto nel sistema di coordinate globali; i vertici iniziali e finali devono essere descritti da *IfcVertexPoint*, e quindi dal sottostante *IfcCartesianPoint*. Per la rappresentazione topologica in *IfcTopologyRepresentation* si utilizzano due stringhe collegate come *attribute*: la prima chiamata *Reference*, la seconda *Edge*, la prima stabilirà un collegamento identificativo, *RepresentationIdentifier*, la seconda tipologico, *RepresentationType*. Mentre *IfcEdge* e *IfcOrientedEdge* non prevedono una geometria curva esplicita sottostante, possono essere utilizzate per sottintendere una sottostante linea dritta come curva di riferimento con l'origine al vertice iniziale. Per ogni *item* strutturale monodimensionale, il parametro *u*, origine della curva di riferimento nella rappresentazione dell'*item* stesso, è l'origine anche del sistema di coordinate locali dello stesso *item*. L'asse locale *x* è, come già detto, parallelo alla tangente della curva e diretto nella stessa direzione del parametro *u*, mentre gli assi *y* e *z* sono orientati secondo la definizione dei rispettivi *subtypes* di *IfcStructuralItem*. In pratica in parametro *u* coincide con la rappresentazione dell'asse della stessa curva di riferimento.

-A questa *entity* si collegano due elementi, il primo è una *subentity*, mentre il secondo un *enumeration type*. Il *type* **IfcStructuralCurveMemberVarying** serve a descrivere le variazioni di proprietà della sezione dell'elemento monodimensionale considerato. Ogni *instance* deve essere composta da due o più *instance* *IfcStructuralCurveMember* con diverse proprietà del profilo, questi elementi subordinati si relazionano attraverso la *entities* *IfcRelAggregates* appartenente a *IfcKernel*. Si suggerisce di modellare anche gli elementi rastremati con una singola *instance* del *supertype* *IfcStructuralCurveMember*.

È raccomandato che le attività strutturali non siano connessi a questa *entity* ma alla superiore, se non nel caso di aggregazione; questo per evitare difficoltà interpretative degli assi locali. Questo è definito come in *IfcStructuralItem* e in *IfcStructuralCurveMember*. La lunghezza lungo l'asse *x* locale di un aggregato dipende dalla continuità e dalla lunghezza delle parti, si suggerisce che l'asse dell'aggregato sia sempre lo stesso, una volta definito l'asse iniziale.

Infine solamente le parti individuali, *instances* diretti di *IfcStructuralCurveMember*, possono portare informazioni sul materiale e il profilo.

-**IfcStructuralCurveMemberTypeEnum** invece è un *enumerated type* che serve a distinguere diversi tipi di elementi strutturali, come ad esempio, cavi. I valori che può assumere sono considerati in base al comportamento analitico che hanno nei confronti dei carichi che portano:

*RIGID\_JOINED\_MEMBER*, cioè un elemento, come una trave, capace di portare carichi trasversali e assiali. I suoi vincoli possono essere incastri o cerniere ed è tipicamente utilizzato nei telai a nodi rigidi;

*PIN\_JOINED\_MEMBER*, ossia un elemento capace di portare solo carichi assiali. Tipicamente usato nelle reticolari;

*CABLE*. Questo valore serve a descrivere un elemento a trazione che sopporta carichi trasversali solamente nel caso in cui sia molto deformato;

*TENSION\_MEMBER*. Simile al valore precedente, ma questo è semplicemente privo di rigidità a compressione;

*COMPRESSION\_MEMBER*, un elemento diametralmente opposto al precedente, ovvero privo di rigidità a trazione;

*USERDEFINED*, un elemento definito specificatamente;

*NOTDEFINED*, un elemento che non viene meglio definito.

-La seconda *entity* che descrive un elemento strutturale, in questo caso bidimensionale, è **IfcStructuralSurfaceMember**. Anche in questo caso è imposta la regola WHERE ed è obbligatorio il valore che le arriva dal *type IfcStructuralSurfaceMemberTypeEnum* che ne descrive la tipologia di elemento strutturale bidimensionale, schematizzando solai, muri e *shell* generici, indipendente dal fatto che la superficie sia planare o curva. Anche per questo elemento è stabilita una regola per il posizionamento degli assi di coordinate locale, in base alla rappresentazione e ricordando che si assume che le *instances* siano locate al centro della superficie. Si sottolinea che il materiale è definito dalla *entity IfcMaterial* che fa riferimento a materiali standardizzati, o, alternativamente da *IfcMechanicalMaterialProperties* o *IfcExtendedMaterialProperties*, e riferito all'elemento attraverso *IfcRelAssociatesMaterial*. È anche definita la



proprietà opzionale *Thickness* che, come da nome, permette di descrivere lo spessore della superficie attraverso *IfcPositiveLengthMeasure*.

Le *instances* dirette di *IfcStructuralSurfaceMember* devono avere una rappresentazione topologica che consiste di una superficie di riferimento, descritta con *IfcFaceSurface*; il sistema di coordinate locali è stabilito dalla superficie di riferimento. In pratica l'origine dei parametri u e v della superficie di riferimento corrisponde con l'origine del sistema di coordinate locali dell'oggetto strutturale, mentre gli assi locali x e y seguono le tangenti alla superficie e sono, di conseguenza, paralleli e nella stessa direzione di u e v rispettivamente; l'asse z, infine, è parallelo nella stessa direzione della normale alla superficie.

La superficie strutturale può essere connessa attraverso curve o superficie di collegamento, descritte rispettivamente con *IfcStructuralCurveConnection* o con *IfcStructuralSurfaceConnection*, a seconda che la connessione con i contorni della superficie strutturale stessa avvenga con elementi mono o bidimensionali.

**-IfcStructuralSurfaceMemberVarying:** questa *entity* descrive il variare delle proprietà delle sezioni dei membri della superficie. Queste proprietà acquistano significato grazie al *property set* *Pset\_StructuralSurfaceMemberVaryingThickness*, oppure grazie a *IfcRelDefinesByProperties*, appartenente ad *IfcKernel*, o tramite composizione. Infatti l'*instance* può essere composta da due o più *instances* *IfcStructuralSurfaceMember* con diverse proprietà. Questi elementi subordinati sono comunque messi in relazione attraverso *IfcRelAggregates*.

Come per l'elemento bidimensionale si raccomanda che l'attività strutturale, cioè azioni e reazioni siano connesse con *IfcStructuralSurfaceMember* anche nel caso di aggregazione per evitare difficoltà interpretative. Le coordinate locali, sempre nel caso di aggregazione, sono genericamente lasciate indefinite dal momento che non è possibile garantire la continuità tra le coordinate locali delle varie parti.

Per la definizione dei materiali e dello spessore, ancora nel caso di aggregazione, solo le singole parti portano con sé l'informazione, anziché l'*instance* *IfcStructuralSurfaceMember*. In tutti gli altri casi rimangono informazioni esclusive dell'*entity* superiore.

**-Pset\_StructuralSurfaceMemberVaryingThickness:** attraverso questo *property set* si definisce il parametro spessore di un membro di una superficie con spessore variabile; in particolare anche con spessore linearmente variabile. Lo spessore è interpolato o estrapolato da tre punti e la localizzazione di questi può essere data o in coordinate locali x e y o in coordinate globali X, Y e Z; in ogni caso questi punti devono posti all'interno o sui bordi della faccia dell'elemento della superficie e non devono essere allineati. Si raccomanda di utilizzare la stessa convenzione, o coordinate locali o globali, all'interno di un *property set*.

I parametri compilabili sono:

*Thickness1:* cioè il primo parametro di spessore di un elemento di una superficie a spessore variabile;

*Location1 Local:* le coordinate locali x e y del punto a cui è collegato il primo parametro di spessore di un elemento di una superficie a spessore variabile;

*Location1 Global:* le coordinate globali X, Y e Z del punto a cui è collegato il primo parametro di spessore di un elemento di una superficie a spessore variabile;

*Thickness2:* il secondo parametro di spessore di un elemento;

*Location2 Local:* le coordinate locali x e y del punto a cui è collegato il secondo parametro di spessore;

*Location2 Global:* le coordinate globali X, Y e Z del punto a cui è collegato il secondo parametro di spessore;

*Thickness3:* il terzo parametro di spessore di un elemento;

*Location3 Local:* le coordinate locali x e y del punto a cui è collegato il terzo parametro di spessore;

*Location3 Global:* le coordinate globali X, Y e Z del punto a cui è collegato il secondo parametro di spessore.

**-IfcStructuralSurfaceMemberTypeEnum** è l'*enumerated type* che serve a distinguere che tipo effettivo di elemento strutturale sia la superficie. I valori possibili sono:

*BENDING\_ELEMENT*, cioè un elemento capace di portare carichi fuori da piano, cioè un *plate*;

*MEMBRANE\_ELEMENT*, cioè un elemento capace di portare carichi nel piano, come un setto di controventamento;

*SHELL*, ossia un elemento che combina i due precedenti, capace quindi di portare carichi sia nel piano che fuori piano;

*USERDEFINED* ossia un elemento specificatamente definito;

*NOTDEFINED* cioè un elemento non ancora meglio definito.

-Gli oggetti catalogati attraverso *IfcStructuralItem* possono essere, oltre che schematizzazioni mono e bidimensionali della struttura, anche vincoli interni o esterni di questa. Per fare ciò buildingSMART utilizza l'*abstract superentity* *IfcStructuralConnection* che rappresenta, appunto, una generica schematizzazione di una connessione strutturale interna oppure di un vincolo esterno.

Si riferisce anch'esso allo *StructuralLoadResource* attraverso *IfcBoundaryCondition*. Infatti un vincolo interno della struttura può supportare anche la condizione di vincolo esterno. Nel caso in cui questa informazione non venga specificata, si assume che l'oggetto non abbia altro che un vincolo con i membri. Il riferimento tra elemento e connessione avviene con una *relationship* con *IfcRelConnects.StructuralMembers* attraverso la quale i membri strutturali possono essere associati alle connessioni.

Si distingue, in ogni caso, tre tipi di connessioni strutturali, a seconda della loro geometria. Quindi vengono definiti tre *subtypes*.

**-IfcStructuralPointConnection:** quest'*entity* descrive un nodo strutturale o un vincolo puntuale, anche inclinato. Il suo sistema di coordinate locale, usato per la descrizione del sistema delle condizioni di supporto, è stabilito da un punto di riferimento dato dalla rappresentazione topologica e dall'*attribute ConditionCoordinateSystem* che, definisce un sistema di coordinate, specificate in relazione al sistema di coordinate globali, attraverso l'attribuzione di *IfcAxis2Placement3D*. Se non meglio specificato il posizionamento di questo *attribute* è sottinteso che *x*, *y* e *z* siano le coordinate di punto di riferimento di questa *instance* e che gli assi di default siano paralleli e orientati con gli assi globali.

Le connessioni puntuali, infatti, devono essere rappresentate da un *IfcVertexPoint* e il sottostante *IfcCartesianPoint*. Questo è il punto di riferimento della connessione nel cosiddetto sistema di coordinate globale. Si usano poi due stringhe di testo per la rappresentazione topologica, che vengono attribuite a *IfcTopologyRepresentation*, la prima identificativa e chiamata *Reference*, la seconda tipologica e chiamata *Vertex*. La prima sarà un *attribute* di *RepresentationIdentifier*, la seconda di *RepresentationType*.

Il punto di riferimento è comunque l'origine del sistema di coordinate locali, gli assi del sistema di coordinate locali sono o paralleli e nella stessa direzione degli assi del sistema di coordinate globali, oppure sono orientati secondo le definizioni dei rispettivi *subtypes* di *IfcStructuralItem*.

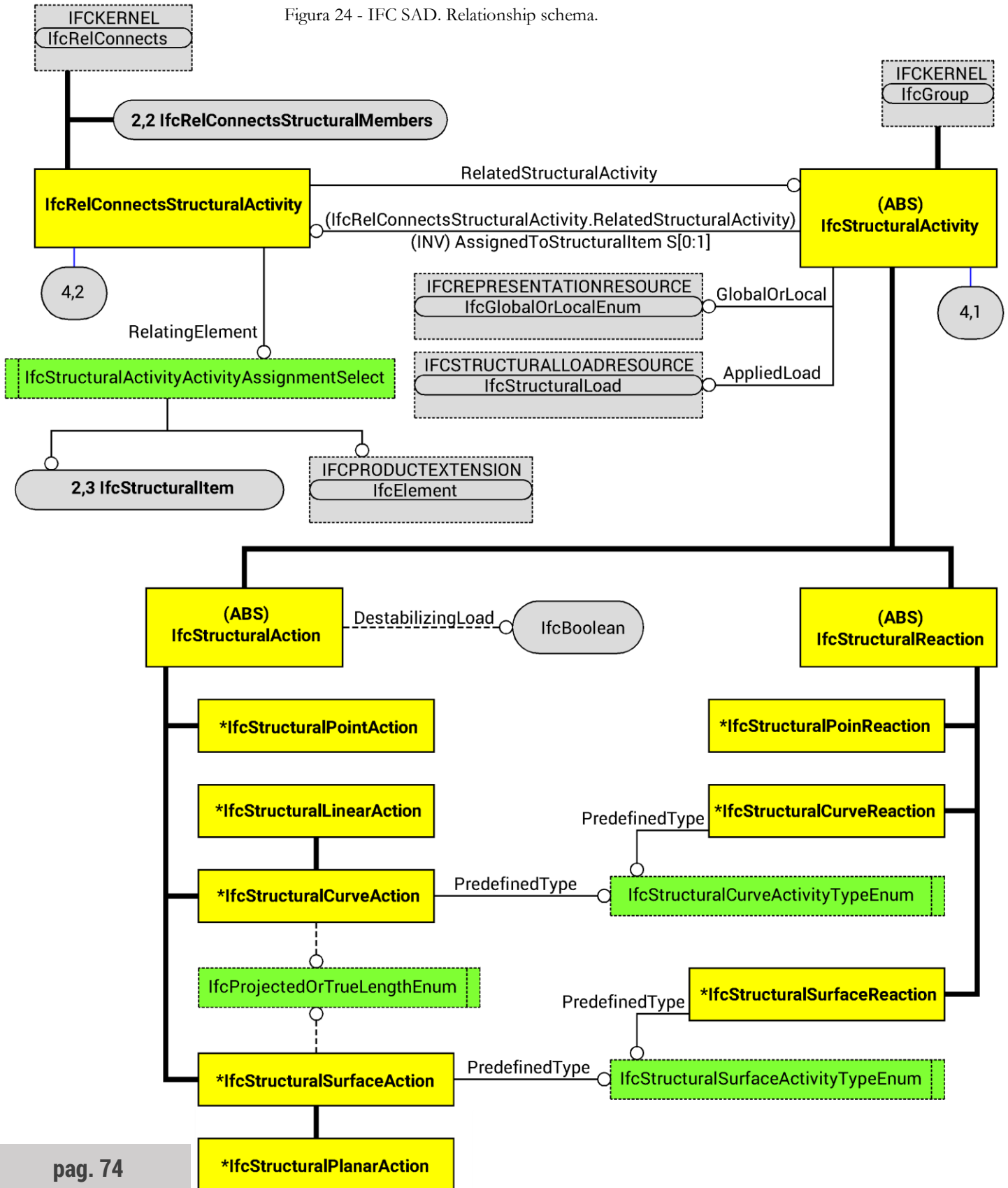
**-IfcStructuralCurveConnection:** questa *instance* descrive i nodi dei margini, cioè dove i lembi di due o più superfici sono uniti, oppure un vincolo esterno monodimensionale. In entrambi i casi si può trattare di elementi rettilinei o curvilinei, anche con l'asse inclinato. Il sistema di coordinate è stabilito dalla curva di riferimento, quindi dalla sua rappresentazione topologica e dal suo *attribute* *Axis*. L'asse locale x è parallelo alla tangente alla curva di riferimento ed è diretto come l'asse della curva di riferimento, mentre l'asse locale y è diretto in modo tale che la terna x, y e z sia levogira. L'asse locale z sarà definito attraverso *IfcDirection* con l'*attribute* *Axis*, in modo che sia specificato relativamente al sistema di coordinate globali. È preferibile e possibile che più *instances* di *IfcStructuralCurveConnection* e *IfcStructuralCurveMember* condividano la stessa *instance* *IfcDirection* come *attribute* *Axis*.

Come per gli elementi strutturali monodimensionali, anche i collegamenti monodimensionali devono essere rappresentati attraverso *IfcOrientedEdge*, oppure *IfcEdgeCurve* o attraverso *IfcEdge*, che descrivono la curva di riferimento nel sistema di coordinate globali. I vertici iniziali e finali saranno descritti attraverso *IfcVertexPoint* e due stringhe di testo, *Reference* ed *Edge*, saranno *attributes* di *IfcTopologyRepresentation* per poter supportare l'identificazione topologica.

**-IfcStructuralSurfaceConnection** è invece una *entity* che descrive i “nodi” delle facce, cioè le porzioni di superfici in cui due o più elementi strutturali bidimensionali sono unite, oppure i vincoli esterni delle facce. Più semplicemente si tratta di una superficie che stabilisce un vincolo, interno o esterno, tra due o più superfici, nel caso di vincolo interno, o di una o più superfici, nel caso di vincolo esterno. In ogni caso le superfici possono essere piane oppure curve.

Il sistema di coordinate locali è stabilito dalla rappresentazione topologica data dalla superficie di riferimento. Queste *instance* devono, di conseguenza, avere una rappresentazione topologica che consiste di una *IfcFaceSurface*, che rappresenta la superficie di riferimento della superficie di connessione, come nel caso si voglia descrivere un elemento strutturale bidimensionale. La sottostante superficie definisce la superficie di riferimento dell'*item* bidimensionale strutturale nel sistema di coordinate globali. Vengono usate due stringhe di testo per la rappresentazione topologica, ossia saranno *attribute* di *IfcTopologyRepresentation* e la prima sarà chiamata *Reference* e sarà identificativa, mentre la seconda sarà tipologica e si chiamerà *Face*; gli *attributes* che le collegheranno saranno *RepresentationIdentifier* e *RepresentationType*. Poiché i vincoli bidimensionali rimangono *item* strutturali bidimensionali la definizione degli assi locali sarà la stessa che per gli elementi strutturali bidimensionali, cioè i parametri  $u$  e  $v$ , l'origine della superficie di riferimento, saranno l'origine del sistema di coordinate locali dell'*item* strutturale; gli assi locali  $x$  e  $y$  seguiranno la tangente alla superficie e saranno paralleli e nella stessa direzione di  $u$  e  $v$  rispettivamente. Infine l'asse  $z$  sarà parallelo e nella stessa direzione della normale alla superficie.

Figura 24 - IFC SAD. Relationship schema.



## RELATIONSHIP

-**IfcRelConnectsStructuralMember**<sup>27</sup> è la prima grande *entity* utilizzata per definire tutte le proprietà necessarie a descrivere il collegamento tra elementi strutturali e oggetti di connessione strutturale, siano essi nodi o vincoli esterni. Viene utilizzata anche per la connessione tra *item* e attività strutturali.

*Instances* di questa *entity* vengono utilizzate per descrivere il collegamento tra una *IfcStructuralPointConnection* e una instance del tipo *IfcStructuralCurveMember* oppure del tipo *IfcStructuralSurfaceMember*. L'attribute *RelatedStructuralConnection* deve, ovviamente, riferirsi a *IfcStructuralPointConnection*. Possono descrivere un collegamento tra una instance del tipo *IfcStructuralCurveConnection* e una instance o del tipo *IfcStructuralCurveMember* oppure del tipo *IfcStructuralSurfaceMember*. Questa volta sarà necessario che l'attribute si riferisca a *IfcStructuralCurveConnection*. Si può, infine, descrivere il collegamento tra un *IfcStructuralSurfaceConnection* e un *IfcStructuralSurfaceMember*, l'attribute punterà a un *IfcStructuralSurfaceConnection*.

L'attribute *RelatingStructuralMember* si riferisce a una *instance IfcStructuralMember* da mettere in collegamento con la specifica connessione strutturale. Questa sarà descritta da una *instance IfcStructuralConnection* e collegata attraverso l'attribute *RelatedStructuralConnection*. L'attribute *AppliedCondition* definisce le proprietà delle connessioni, collegandosi ad *IfcBoundaryCondition* dello *Structural Load Resource* e permette di descrivere tutte le connessioni di qualunque dimensionalità, siano esse rigide, a cerniera, elastiche o quant'altro. Attraverso *AdditionalCondition* si descrivono ulteriori proprietà delle connessioni, cioè condizioni di cedimento e spostamento di attivazione, collegandosi ad *IfcStructuralConnectionCondition*, *entity* dello *Structural Load Resource*. Questi ultimi due *attributes* sono dati in riferimento al sistema di coordinate assegnato dall'attribute *ConditionCoordinateSystem*, che è definito relativo al sistema di coordinate locali dell'elemento strutturale stesso. Con l'attribute *ConditionCoordinateSystem*, attraverso *IfcAxis2Placement3D* del *Geometry Resource*, si definisce un sistema di coordinate utilizzato poi per la descrizione delle proprietà del collegamento in *ConnectionCondition* relative al sistema di coordinate locali di *RelatingStructuralMember*. Se non meglio specificato, il

---

27 - buildingSMART, "IFC4 - Add.2; 7.7.3.2 IfcRelConnectsStructuralMember".

posizionamento di *IfcAxis2Placement3D* si assume che  $x$ ,  $y$  e  $z$  siano le stesse coordinate locali di *IfcStructuralMember* dove avviene la connessione. Infine, l'ultimo *attribute*, *SupportedLength* definisce la lunghezza, o ampiezza, di supporto della connessione fisica lungo la curva di collegamento strutturale, come spiegato nella figura che segue.

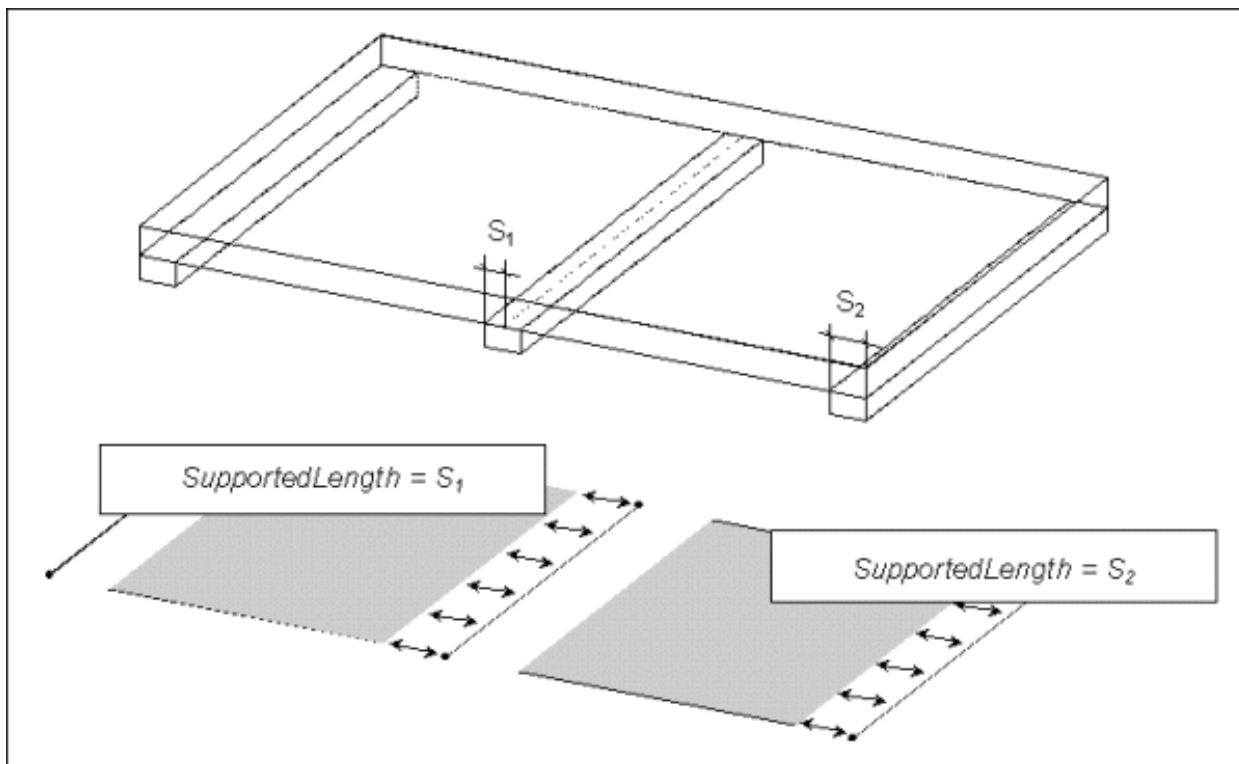


Figura 25 – Spiegazione lunghezza della connessione. Cfr. buildingSMART, “IFC4 - Add.2; 7.7.3.2 IfcRelConnectsStructuralMember”.

È, in conclusione, presente una *subentity*.

-**IfcRelConnectsWithEccentricity** che aggiunge la definizione dell'eccentricità del collegamento tra l'elemento strutturale e la relativa connessione, sia che si tratti di un vincolo esterno sia che si tratti di uno interno. L'altra eccentricità specificabile nel modello è indipendente da questa, infatti il profilo di sezione di una curva può essere inserito eccentricamente, rispettando la curva di riferimento dell'elemento. Possono, queste due eccentricità, essere utilizzate indipendentemente l'una dall'altra, purché rispettino le rispettive regole.



È presente un *attribute*, *ConnectionConstraint*, che esplicita l'eccentricità tra due oggetti topologici, sia due vertici, due spigoli o due facce. Se deve riferire ad *IfcConnectionPointGeometry* e a due *instances* dello stesso tipo, due *IfcVertexPoint*, nel caso di una connessione puntuale, due *IfcEdge*, nel caso di una connessione monodimensionale, due *IfcFaceSurface*, nel caso di connessione bidimensionale.

-**IfcRelConnectsStructuralActivity**<sup>28</sup> è la seconda grande *entity* di relazione e viene utilizzata per collegare le attività strutturali, indipendentemente dal fatto che si tratti di azioni o di reazioni, all'*item* strutturale, sia esso un elemento o un collegamento, o al generico oggetto a cui si riferiscono.

Per assolvere a questa funzione si appoggia a due *attribute*, *RelatingElement* e *RelatedStructuralActivity*. Il primo, riferendosi ad *IfcStructuralActivityAssignmentSelect*, permette di specificare a quale *item* strutturale o elemento generico l'attività considerata si applica. Il secondo, attraverso *IfcStructuralActivity*, specifica quale sia l'effettiva attività strutturale agente.

-**IfcStructuralActivityAssignmentSelect** è, appunto, l'*enumerated type* che permette di distinguere all'*instance* a cui l'attività strutturale si riferisce, sia essa un elemento del sistema edilizio o un *item* propriamente strutturale, permettendo così di stabilire la connessione tra le attività e l'elemento su cui questa insiste.

I due *type* possibili sono, infatti *IfcStructuralItem* e *IfcElement*.

---

28 - buildingSMART, "IFC4 - Add.2; 7.7.3.1 IfcRelConnectsStructuralActivity".

## IFC STRUCTURAL ACTIVITY

-**IfcStructuralActivity**<sup>29</sup> è un'*abstract entity* che combina la definizione di azioni, come forze o spostamenti, e di reazioni, come reazione vincolari, forze interne o deformazioni, le quali vengono specificate usando le definizioni base dei carichi dell'*IfcStructuralLoadResource*. La differenziazione tra azioni e reazioni è realizzata con la formazione di due *instances* attraverso *IfcStructuralAction* o *IfcStructuralReaction*, rispettivamente. Questi *subtypes* ereditano gli *attributes* necessari da *IfcStructuralActivity*, in particolare le *relationship* che mettono in relazione le azioni o le reazioni con gli elementi o i collegamenti strutturali o con elementi in generale, descritti attraverso, quindi *IfcStructuralItem* oppure, nell'ultimo caso, con *IfcElement*. Nel caso in cui un'attività strutturale sia collegata ad un elemento non strutturale saranno previsti accordi al di fuori di questo ambito; la semantica di un'attività strutturale è pienamente definita se una *instance* di attività è collegata a soltanto un *IfcStructuralItem*; l'*inverse attribute AssignedToStructuralItem* può essere solo vuoto solo in modelli incompleti o concettuali, quindi non pronti o utilizzabili per le analisi.

Sono nuovamente previsti due sistemi di coordinate, quelle globali e quelle locali. Il sistema di coordinate globali è il sistema di coordinate condiviso da tutti gli *items* e le *activities* raggruppate nel comune *IfcStructuralAnalysisModel*. Questo sistema di coordinate è stabilito da un *ObjectPlacement*. Non è necessario che il sistema di coordinate sia lo stesso sistema di coordinate mondiali previsto in *IfcProject*. Il sistema di coordinate locali, invece, è il sistema di coordinate riferito all'oggetto strutturale, sia esso una connessione o un elemento strutturale. Questo sistema è stabilito da un *attribute* di *Representation*, unitamente a un'ulteriore specificazione data da definizioni e *attributes* dei *subtypes* di *IfcStructuralItem*. La rappresentazione matematica è sempre data all'interno dell'*attribute ObjectPlacement*, cioè nel cosiddetto sistema di coordinate globali riferito al modello delle analisi strutturali.

---

29 - buildingSMART, "IFC4 - Add.2; 7.7.3.5 IfcStructuralActivity".

Inoltre, l'applicazione dei carichi, che avviene con l'attribute *AppliedLoad*, se descritto con il type *IfcStructuralLoadConfiguration*, è sempre dato in coordinate locali; la direzione dei carichi invece è definita locale o global a seconda dell'attribute *GlobalOrLocal*.

Per il posizionamento e la rappresentazione topologiche, attraverso *ObjectPlacement* e *Representation*, sono a volte non specificati esplicitamente, potendo essere impliciti come nei casi sottostanti; i sistemi di coordinate globali e locali sono determinati alla stessa maniera anche se posizione e rappresentazione sono impliciti. Le *instances* di *IfcStructuralActivity* collegate con un *item* strutturale della stessa dimensionalità non devono avere né posizionamento né rappresentazione, poiché sono implicitamente considerati gli stessi dell'*item* strutturale di riferimento; si intende il caso di un'azione o una reazione puntuale collegata con un *item* puntuale, cioè *IfcStructuralPointConnection*; o il caso di un'azione o reazione che si sviluppa in lunghezza collegata con un *item* strutturale monodimensionale, cioè *IfcStructuralCurveConnection* oppure *IfcStructuralCurveMember*; oppure, infine, il caso di un'azione o reazione che si sviluppa in piano collegata a un *item* strutturale bidimensionale, cioè *IfcStructuralSurfaceConnection* o *IfcStructuralSurfaceMember*, e che agiscono sull'intera superficie dell'oggetto e non è specificato un carico costante per curve. Mentre le *instances* di *IfcStructuralActivity* che sono collegate con un *item* strutturale monodimensionale, descritto con *IfcStructuralCurveConnection* o con *IfcStructuralCurveMember*, e che agiscono su un punto dell'*item*, così come se collegate con un *item* strutturale bidimensionale, descritto con *IfcStructuralSurfaceConnection* o con *IfcStructuralSurfaceMember*, e agenti su un punto, su una curva o su una porzione della superficie dell'*item*, devono invece avere una rappresentazione topologica ben precisa, più sotto specificata, che includa il posizionamento e la rappresentazione del *product*. Va utilizzato *IfcProductRepresentation* che deve essere dato da *IfcTopologyRepresentation* attraverso l'attribute *Representation*. Infine, le *instances* di *IfcStructuralActivity* collegate con *item* strutturali bidimensionali e che sono specificati carichi per profili di carico costante, devono avere una rappresentazione per forma ben precisa, anch'essa specificata dopo, che includa posizionamento e rappresentazione del *product*. Va utilizzato *IfcProductRepresentation* che deve essere data da *IfcShapeRepresentation* attraverso l'attribute *Representation*; le rappresentazioni per forma e topologiche possono essere combinate.

Il posizionamento locale è definito da *IfcStructuralActivity* è definito dal suo *supertype* *IfcProduct*, cioè è definito da *IfcLocalPlacement* che stabilisce un sistema di coordinate globali che deve essere comune a tutti gli *item* e le *activities* in un *IfcStructuralAnalysisModel*.

La rappresentazione topologica invece si distingue in base alla dimensionalità dell'attività strutturale considerata. Ossia, *instances* che agiscono su un *item* strutturale bidimensionale devono avere una rappresentazione topologica data dalla sottostante superficie geometrica, descritta da *IfcFaceSurface*, la quale dovrebbe essere un singolo *item* di *IfcTopologyRepresentation*. La superficie stabilisce, così, un sistema di coordinate locali per l'attività, in cui l'origine dei parametri u e v della superficie è l'origine del sistema di coordinate locali, in cui gli assi x e y sono orientati sulle tangenti alla superficie, quindi paralleli e direzionati secondo u e v rispettivamente, mentre l'asse z si ritrova parallelo e diretto secondo la normale alla superficie; il tutto forma una tripletta di assi cartesiani che segue la regola della mano destra. L'identificazione della rappresentazione è per riferimento e deve essere, tipologicamente, una superficie.

Mentre *instances* che insistono su di una curva posta su di un oggetto bidimensionale devono avere una rappresentazione topologica data da un bordo, attraverso *IfcEdge* o un *subtype*, che dovrebbe essere, anche in questo caso come nel precedente, un singolo *item* di *IfcTopologyRepresentation*. La geometria deve essere compatibile, ovviamente con la superficie geometrica sull'oggetto collegato. In unione con questa superficie la curva stabilisce un sistema di coordinate locali dell'attività, dove l'origine del parametro del parametro u è l'origine del sistema di coordinate locali, l'asse x segue la tangente alla curva ed è direzionato secondo u, l'asse z è parallelo e diretto secondo la normale alla superficie dell'*item* di riferimento collegato; infine l'asse y è definito sapendo che la tripletta x, y e z forma una tripletta cartesiana che segue la regola della mano destra. L'identificazione della rappresentazione è per riferimento e deve essere, tipologicamente, una curva.

Mentre *IfcEdge* non provvede un'esplicita geometria curve, può essere utilizzata un'implicita linea dritta come curve di riferimento con l'origine del parametro della curva come punto di vertice iniziale.

Infine, *instances* che insistono su di un singolo punto posto su di un *item* mono o bidimensionale, devono avere una rappresentazione topologica data da un *IfcVertexPoint*, che dovrebbe essere un

singolo *item IfcTopologyRepresentation*. La geometria del punto deve essere compatibile con la geometria della curva o della superficie di riferimento; il sistema di coordinate locali dell'attività è orientato della geometria a cui è collegata, secondo le stesse regole di cui sopra. L'identificazione della rappresentazione è per riferimento e deve essere, tipologicamente, un vertice.

Per la rappresentazione, si impone che *instances* che agiscono su elementi bidimensionali e sono specificate da curve di livello debbano avere una rappresentazione della geometria data da un set di curve su di una superficie, attraverso *IfcPCurve*. Invece, la superficie di base deve aderire, o preferibilmente essere la stessa, alla superficie dell'*item* strutturale a cui l'attività è collegata. L'identificazione della rappresentazione è per livelli e deve essere, tipologicamente, un insieme di curve geometriche.

Vengono definiti tre *attributes*, il primo è *AppliedLoad* e si riferisce ad *IfcStructuralLoad* che descrive le risorse, ovvero le specificazioni, dei carichi o dei risultati che definiscono il tipo di carico, la direzione e i valori. Nel caso di attività strutturali dalla distribuzione variabile su curve o superficie si utilizza *IfcStructuralLoadConfiguration* per listare i vari valori di carico e la loro posizione nella distribuzione del carico, misurata in coordinate locali dell'elemento strutturale, mono o bidimensionale, su cui l'attività corrispondente agisce. I contenuti di questa distribuzione di carico o di risultato può essere ulteriormente ristretta per definizioni a oggetti sottostanti a *IfcStructuralActivity* stessa.

Il secondo *attribute*, *IfcGlobalOrLocalEnum*, indica se le direzioni del carico si riferiscano al sistema di coordinate globali del modello delle analisi, cioè al sistema stabilito dall'*attribute SharedPlacement* di *IfcStructuralAnalysisModel*, oppure al sistema di coordinate locali, rispetto all'attività o all'*item* collegato, così come stabilito esplicitamente o implicitamente dalla rappresentazione e dai parametri spaziali previsti. Questo *attribute* in sé non distingue il tipo di sistema di coordinate, benché questa distinzione sia necessaria nel modello di analisi, ma, anche qui, il sistema di coordinate globali può benissimo essere diverso tra il modello di analisi e il modello dell'intero sistema edilizio. Scegliendo, poi, *GLOBAL\_COORDS* è perciò da intendersi non come sistema di coordinate del sistema edilizio, ma come sistema di coordinate globali del modello di analisi. Per contro, *LOCAL\_COORDS* deve essere

considerato come coordinate locali dei singoli elementi o connessioni strutturali, o attività strutturali, come stabilito dalle definizioni geometriche d'uso delle specifiche *entities* considerate.

Il terzo e ultimo attribute è *AssignedToStructuralItem*, e il suo inverse *RelatedStructuralActivity*, collega *IfcRelConnectsStructuralActivity* e riferenzia la relationship *IfcRelConnectsStructuralActivity* con cui le attività sono collegata con gli *items* strutturali.

-**IfcStructuralAction** è una dei due *entities* di *IfcStructuralActivity* e serve a descrivere un'azione strutturale che agisce su di un *item* strutturale o un elemento del progetto. Si tratta di un'*abstract entity* che distingue, attraverso *subtypes*, le azioni in puntuali, lineari o superficiale. Le azioni stesse sono, poi, raggruppate o con *IfcStructuralLoadGroup* o, più spesso, con *IfcStructuralLoadCase* e sono anche riferite al modello di analisi strutturale a cui i rispettivi carichi fanno riferimento.

Di più è possibile stabilire una *relationship* tra le azioni in un modello di analisi e le reazioni di un altro modello di analisi in modo che questo funzioni come origine delle azioni del primo. Per esempio una reazione vincolare di un sistema strutturale può essere considerato come carico su di un altro sistema strutturale che supporta il primo.

È, infine, presente per questa *entity*, un *attribute* ad *IfcBoolean* che permette di indicare se l'azione può causare un problema di stabilità, attraverso l'assegnazione della variabile TRUE oppure FALSE. Nel caso in cui sia FALSE non è necessaria alcuna indagine riguardo problemi di stabilità.

-**IfcStructuralPointAction** è il primo *subtype* di *IfcStructuralAction* e definisce un'*entity* che rappresenta un'azione puntuale. Così si vuole descrivere o una singola forza o un singolo spostamento. Un'azione puntuale è tipicamente riferita a un collegamento puntuale, ma può essere collegata anche a un elemento strutturale o a un collegamento sia monodimensionale che bidimensionale.

Nel caso standard in cui l'azione puntuale sia collegata ad un *item* puntuale, l'*instance* in questione non deve avere ulteriori specificazioni di rappresentazione o posizionamento poiché si intende che queste informazioni siano le stesse presenti nell'*item* strutturale di riferimento. Se invece l'azione puntuale si riferisce ad un *item* strutturale dimensionalmente maggiore o nel caso in cui non si riferisca ad alcun *item* strutturale, cosa che avviene per modelli incompleti o concettuali, è invece necessario specificare

posizionamento e rappresentazione, attraverso l'attribute *ObjectPlacement* e l'entity *IfcVertexPoint*. Per meglio modellare azioni concentrate su di una curva o una superficie, è preferibile utilizzare *IfcStructuralCurveAction* o *IfcStructuralSurfaceAction* assegnando attraverso il collegato *enumerated type* la variabile *DISCRETE* perché così non è richiesto un'ulteriore rappresentazione topologica; in ogni caso *IfcStructuralPointAction* dovrebbe essere utilizzato per azioni concentrate su di una curva o una superficie solo quando si desidera rappresentare esplicitamente un vertice.

**-IfcStructuralCurveAction:** è il secondo *subtypes* utile a descrivere un'azione strutturale, questa volta un'azione distribuita lungo una curva. Quest'azione può essere connessa con un elemento o una connessione strutturali monodimensionali, oppure con un elemento o una connessione strutturali bidimensionali.

*IfcStructuralCurveAction* è meglio definita da due *enumerated types* che descrivono, il primo se il valore di carico è dato per lunghezza effettiva della curva di riferimento o per lunghezza della proiezione della curva nella direzione del carico – quest'ultimo caso è applicabile ai soli carichi definiti nel sistema globale di coordinate –, il secondo l'orientamento e la proiezione lungo gli assi globali o locali e il tipo di azione in base alla sua distribuzione di valori di carico lungo la curva di riferimento.

Per quando riguarda la definizione topologica, se l'azione si collega ad un *item* con la stessa dimensionalità, non si deve provvedere a definire la curva, in quanto si implica che posizionamento e rappresentazione sia la stessa dell'*item* strutturale, definito con *IfcStructuralItem*. Se invece l'azione si collega a un *item* bidimensionale, dalla dimensionalità maggiore, oppure nel caso in cui non si colleghi ad alcun *item* strutturale, cosa che avviene in un modello incompleto o concettuale, è necessario, allora, fornire sia il posizionamento, attraverso l'attribute *ObjectPlacement* e la rappresentazione, attraverso l'entity *IfcEdgeCurve*.

Si prevedono una serie di proposizioni informali, a seconda del valore che restituisce l'*enumerated type* che descrive l'andamento del carico. Se l'azione è costante, con andamento sinusoidale o parabolico allora l'attribute *AppliedLoad* che collega la *superentity* *IfcStructuralActivity*, contenente *IfcStructuralCurveAction* non può restituire, dallo *StructuralLoadResource*, un *item* *IfcStructuralLoadConfiguration*. Di più, se il carico ha andamento sinusoidale o parabolico, l'*item* di carico definisce il massimo del carico al centro della

distribuzione del carico stesso. Se, invece l'azione è di tipo linear, l'*attribute AppliedLoad* che collega la *superentity IfcStructuralActivity*, deve restituire *IfcStructuralLoadConfiguration* e contenere due *item*. Se invece, il carico ha andamento poligonale, gli *items* collegati devono essere due o più, come sopra. Nel caso di andamento lineare, poligonale o discreto, devono un *attribute Locations* monodimensionale che definisce il posizionamento in coordinate locali dei valori del carico lungo la curva d'azione. Gli *items* di carichi devono essere dati in ordini crescente secondo la loro posizione, il primo e l'ultimo definiscono l'estensione della distribuzione del carico. Si deve modellare azioni puntuali come carichi discreti, così contengono due o più carichi puntuali, mentre singoli carichi puntuali sono modellati con *IfcStructuralPointAction*.

Le proposizioni formali, invece, prevedono che attraverso l'*attribute ProjectedIsGlobal* un carico possa insistere sulla proiezione di una lunghezza se è stato specificato in coordinate globali, cioè nelle coordinate del modello di analisi; infatti se il carico viene specificato in coordinate locali può essere riferito solamente ad una lunghezza reale. Una seconda preposizione prevede che l'*attribute HasObjectType* debba essere specificato se viene utilizzato, all'interno di *IfcStructuralCurveActivityTypeEnum*, il set *USERDEFINED*. Infine, si prevede che non si possa non sia possibile descrivere una distribuzione di carico del tipo *EQUIDISTANT* per un'azione su di un elemento strutturale monodimensionale, la sua utilità è nel descrivere, infatti, una reazione.

**-IfcStructuralLinearAction** è l'*unico subtype* possibile, al momento, per *IfcStructuralCurveAction* e definisce un'azione con valore costante distribuito su di una curva. Sono definite due *rules*, la prima, *SuitableLoadType*, fa sì che l'azione in questione si riferisca a una forza linear oppure a un carico di temperatura, la seconda, *ConstPredefinedType*, fa sì che l'*enumerated subtype* possa restituire solamente una distribuzione di carico costante sul suo dominio.

**-IfcStructuralCurveActivityTypeEnum** è un *enumerated type* atto a definire la distribuzione del valore del carico in un'azione distribuita su di una curva. I valori che possono essere assegnati sono:

*CONST*: descrive un carico dal valore costante per la sua intera estensione;

*LINEAR*: un carico dal valore distribuito linearmente nella sua estensione;



*POLYGONAL*: il carico consiste di varie sezioni consecutive in cui il carico varia linearmente, cioè vari segmenti di carico lineare di tipo *LINEAR*;

*EQUIDISTANT*: il carico consiste in N sezioni consecutive della stessa lunghezza ed è specificato un numero N+1 di valori di carico. L'interpolazione sui segmenti non è definita qui, ma può essere qualificata attraverso l'*attribute ObjectType*, previ ulteriori specificazioni.

*SINUS*: il valore del carico è distribuito come una semilunghezza d'onda di una sinusoide;

*PARABOLA*: il valore del carico è distribuito come una mezza onda descritta da una parabola quadratica e simmetrica;

*DISCRETE*: il carico è definito come una serie di almeno due separati carichi puntuali;

*USERDEFINED*: la distribuzione del carico è definito dall'utente;

*NOTDEFINED*: la distribuzione del carico non è definita.

**-IfcProjectedOrTrueLengthEnum** è un *enumerated type* necessario per la definizione del carico ed è utilizzato solamente se i valori dei carichi sono dati come azione globale e se definiscono carichi mono o bidimensionali.

Per l'interpretazione della definizione di un carico dipendentemente da *IfcGlobalOrLocalEnum* e da *IfcProjectedOrTrueLengthEnum*.

Sono presenti due valori possibili, che non vengono ulteriormente descritti, ma il cui nome rimane abbastanza chiarificante:

*PROJECTED\_LENGTH*: cioè il caso in cui si debba considerare il carico agente sulla proiezione;

*TRUE\_LENGTH*: il caso, invece, in cui sia da considerare la lunghezza effettiva, con il suo andamento, come linea d'azione del carico.

-**IfcStructuralSurfaceAction**: è il terzo e ultimo *subtypes* utile a descrivere un'azione strutturale, in questo caso definisce un'azione distribuita su di una superficie. Un'azione superficiale dovrebbe essere collegata a un *item* strutturale bidimensionale.

Vi si collegano due *enumerated types* atti a descrivere se il carico sia orientato sugli assi locali o globali e che distribuzione abbia il carico sulla superficie di riferimento.

Normalmente *IfcStructuralSurfaceAction* è intesa collegata a un intero *item* bidimensionale e, in questo caso, non necessita di ulteriori specificazioni né di posizionamento né di rappresentazione, poiché si sottintende che queste informazioni siano le stesse dell'*item* strutturale di riferimento. Se, invece si riferisce a una parte di *item* strutturale bidimensionale o non si riferisce ad alcunché, che avviene solo per modelli incompleti o concettuali, la superficie deve essere descritta sia nel posizionamento, attraverso l'*attribute ObjectPlacement*, che nella rappresentazione, attraverso *IfcFaceSurface* che topologicamente deve descrivere la parte della *item* superficiale su cui insiste il carico.

Anche in questo sono previste delle proposizioni informali, simili al caso di *IfcStructuralCurveAction*. Se la distribuzione è costante, l'*attribute AppliedLoad* di *IfcStructuralActivity* non deve restituire *IfcStructuralLoadConfiguration*. Se invece l'andamento è bilineare lo stesso *attribute* deve restituire *IfcStructuralLoadConfiguration* e deve contenere tre *items* con *attribute Locations* bidimensionali, che definiscono il posizionamento dei valori del carico in coordinate locali sulla superficie di riferimento. Se l'andamento è a punti, lo stesso *attribute* deve restituire lo stesso *IfcStructuralLoadConfiguration* e deve contenere due o più *items* bidimensionali per descrivere la posizione dei carichi puntuali. I carichi puntuali vanno modellato, appunto, attraverso un carico superficiale a punti, se ci sono due o più punti di carico, altrimenti vanno modellati con *IfcStructuralLoadPoint*.

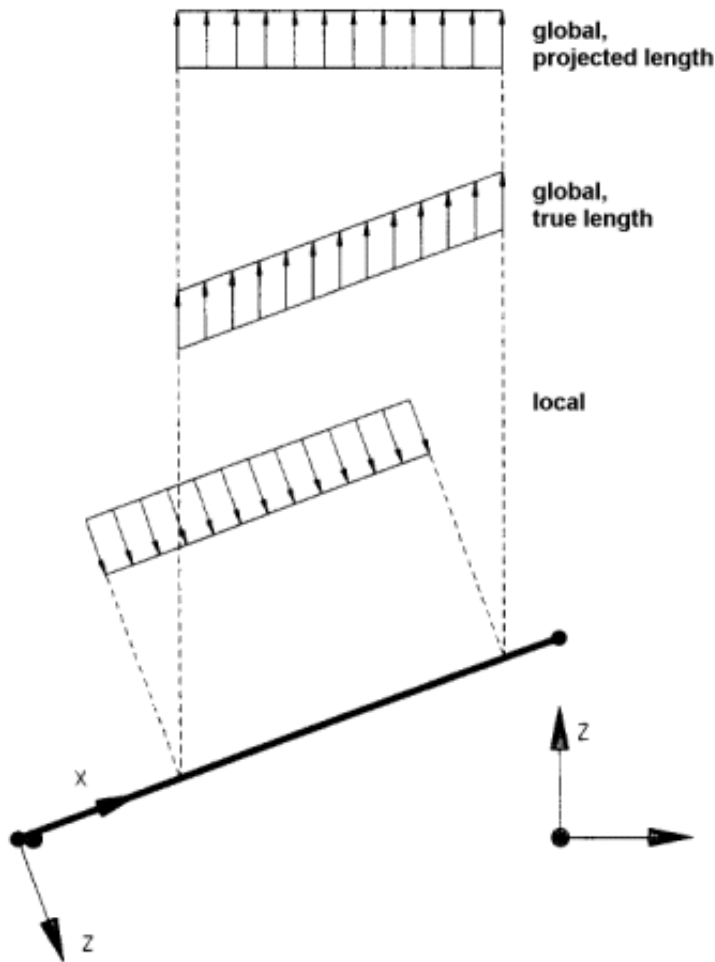


Figura 26 - IFC SAD. Definizione della direzione dei carichi.

Attraverso due *attributes*, *ProjectedOrTrue* e *PredefinedType*, che collegano rispettivamente *IfcProjectedOrTrueLengthEnum* e *IfcStructuralSurfaceActivityTypeEnum*, è possibile descrivere, con il primo *attribute*, se il valore del carico è definito per la lunghezza effettiva della superficie su cui agisce, oppure se agisca sulla proiezione di questa oppure, con il secondo *attribute*, l'andamento della distribuzione dei valori di carico.

Le proposizioni formali prevedono, come si dice parlando dell'*attribute ProjectedOrTrue*, che, attraverso la regola *ProjectedIsGlobal*, un carico può essere riferito alla lunghezza proiettata solo se specificato in coordinate globali, cioè quelle del modello di analisi; se il carico,

infatti, è specificato in coordinate locali, può riferirsi solamente alla lunghezza effettiva della superficie. Inoltre, l'*attribute ObjectType* deve essere specificato, per la regola *HasObjectType*, ogni volta che si specifica che l'andamento del carico è definito dall'utente attraverso *USERDEFINED*.

-**IfcStructuralPlanarAction** è l'*entity* che definisce un'azione con valore costante su di una superficie. Vi sono due *rules*, la prima *SuitableLoadType*, definisce che l'azione debba descrivere una forza bidimensionale oppure un carico di temperatura, la seconda, *ConstPredefinedType*, invece che il *subtype* possa restituire solamente un valore costante di carico distribuito sul dominio.

-*IfcStructuralSurfaceActivityTypeEnum* è un *enumerated type* che permette di descrivere l'andamento e la distribuzione dei valori di carico in un'azione o reazione distribuita su di una superficie. I valori che possono essere assegnato sono:

*CONST*: il carico ha valore costante per tutta la sua estensione;

*BILINEAR*: il valore del carico è distribuito bilinearmente sulla sua estensione;

*DISCRETE*: il carico è specificato come una serie di almeno due separati carichi puntuali;

*ISOCONTOUR*: il carico è definito da una serie di iso-curves, cioè curve di livello sulle quali il valore del carico è costante; queste curve sono perpendicolari al gradiente del carico;

*USERDEFINED*: ossia la distribuzione del carico è definita dall'utente;

*NOTDEFINED*: cioè la distribuzione del carico non è definita.

-**IfcStructuralReaction**: è il secondo di due *subtypes* di *IfcStructuralActivity*. Questo viene utilizzato per descrivere l'attività strutturale che risulta da un'azione strutturale imposta su un *item* strutturale o su un elemento del modello. Possono essere descritti sia reazioni vincolari che sforzi interni che deformazioni. Vengono poi raggruppate all'interno di *IfcStructuralResultGroups*, attraverso l'*inverse relationship HasAssignments* e attraverso una *entity* del tipo *IfcRelAssignsToGroups*. Infine attraverso l'*attribute ResultGroupFor* si mette in relazione *IfcStructuralResultGroups* in modo che i risultati siano riferiti al modello di analisi strutturali corrispondente.

Si può, di più, come già detto per le azioni, stabilire una relazione tra le reazioni di un modello di analisi e le azioni che queste possono causare in un secondo modello. Per esempio le reazioni vincolari di un sistema strutturale possono fungere da carichi su di un altro sistema strutturale; questo è espresso tramite l'*inverse relationship HasAssignment* della reazione e attraverso *IfcRelAssignsToProduct*, a cui il nome è impostato come causa, scrivendo "Cause" e l'*attribute RelatingProduct* si riferisce ad un'*instance* del *subtype* di *IfcStructuralAction*.

-**IfcStructuralPointReaction** è la prima di tre *subentity* di *IfcStructuralReaction* che descrive una reazione che si verifica in un punto. Questa reazione esprime o una singola forza o un singolo spostamento.

Tipicamente è collegata a una connessione puntuale, ma può essere anche collegata a un qualsiasi *item* strutturale descritto con *IfcStructuralItem*. Nel caso standard in cui si colleghi a *item* puntuali, non deve esserci ulteriore specificazione di posizione e rappresentazione, poiché si sottintende che queste informazioni coincidano con quelle dell'*item* a cui si riferiscono. Quando invece questa *entity* fa riferimento a un *item* dalla dimensionalità maggiore è obbligatorio definirne la rappresentazione topologica, nonostante sia meglio modellare azioni puntuali su di una curva o una superficie attraverso il *type DISCRETE* poiché non richiede ulteriori specificazioni di sorta; questa modellazione è utile per rappresentare reazioni concentrate solamente quando è necessario rappresentare esplicitamente un vertice. Si suggerisce di specificare la rappresentazione topologica quando l'*instance* in questione non si riferisce ad alcun oggetto strutturale, come nel caso di modelli incompleti o concettuali.

**-IfcStructuralCurveReaction** è la seconda di tre *subentity* di *IfcStructuralReaction* e descrive una reazione che si verifica su di una curva. Può essere collegata a un elemento o a una connessione strutturale mono o bidimensionali. Normalmente collegata a un *item* monodimensionale, questa *instance* non deve essere ulteriormente specificata topologicamente, poiché si assume che le sue proprietà siano le stesse della curva di riferimento. Nel caso in cui sia collegata a un *item* bidimensionale è obbligatorio, invece, un'ulteriore specificazione.

Similmente ad *IfcStructuralCurveAction* sono definite delle identiche proposizioni informali. Rispetto però, alle azioni, si differenziano perché, se la definizione degli *attributes* rimane identica, cambiano le proposizioni formali. Infatti *PredefinedType* riferito a *IfcStructuralCurveActivityTypeEnum* prevede una serie di tipologie di distribuzione dei valori di carico tra cui scegliere, ma attraverso la regola *SuitablePredefinedType* non sono utilizzabili distribuzioni di carico sinusoidali o paraboliche, attraverso *SINUS* e *PARABOLA* perché servono a descrivere solamente azioni sull'elemento strutturale monodimensionale. Infine, la regola *HasObjectType* prevede che *attribute ObjectType* debba essere obbligatoriamente definito quando si utilizza il set predefinito per descrivere una distribuzione di valori di carico qualunque, come deciso dall'utente attraverso *USERDEFINED*.

**-IfcStructuralSurfaceReaction** è, infine, l'ultima di tre *subentity* di *IfcStructuralReaction* e serve a descrivere una reazione che avviene su di una superficie, può essere collegata a un *item* strutturale della stessa dimensionalità.

Identicamente ad *IfcStructuralSurfaceAction* sono definite delle proposizioni informali. Ossia che la reazione della superficie è costante, cioè descritta attraverso *CONST*, l'*attribute AppliedLoad* non deve essere dato con *IfcStructuralLoadConfiguration*. Questo deve essere utilizzato, invece, quando è di tipo bilineare, attraverso, cioè, *BILINEAR* e devono essere tre *items* con *attribute Location* bidimensionali per specificare, in coordinate locali, il posizionamento dei valori di reazione sulla superficie. Se, invece, la reazione è definita per curva di valore costante, attraverso *ISOCONTOUR*, l'*attribute AppliedLoad* dato con *IfcStructuralLoadConfiguration* deve contenere il numero di *items* specificati. Ogni *item* di *IfcStructuralLoadConfiguration* deve contenere posizionamenti bidimensionali che definiscono la localizzazione dei valori di carico, in coordinate locali della superficie, sulla stessa; infine, ogni *item* deve essere riferito a esattamente una curva di valore di reazione costante. Il set per la rappresentazione degli *items* non è ordinata, qui i posizionamenti del risultato sono richiesti da correlare ai valori di risultati e alle curve di valore costante. Queste sono rappresentate con *IfcPCurve* che è definito in termini di parametri della superficie *u* e *v*, mentre la posizione dei risultati è data in coordinate locali della superficie, cioè in *x* e *y*; quindi si raccomanda fortemente che la parametrizzazione di *u* e *v* sia in scala 1:1 per evitare diverse scale tra *u*, *v* e *x*, *y*. Così facendo i due sistemi di coordinate coincidono. Poiché la *superentity* *IfcStructuralActivity* è collegata attraverso l'*attribute AppliedLoad* a *IfcStructuralLoadConfiguration* appartenente allo *Structural Load Resource*, è necessario che i valori listati attraverso l'*attribute Value* siano tutti dello stesso tipo di *entity*, ovvero tutte forze puntuali, ad esempio, oppure singoli spostamenti.

La definizione degli *attribute* prevede solamente *PredefinedType* verso *IfcStructuralSurfaceActivityTypeEnum* per descrivere il tipo d'andamento dei valori di reazione sulla superficie. Infine l'unica proposizione formale, la regola *HasPredefinedType*, impone che l'*attribute ObjectType* debba essere specificato obbligatoriamente se la distribuzione dei valori di reazione è specificato dall'utente attraverso *USERDEFINED*.

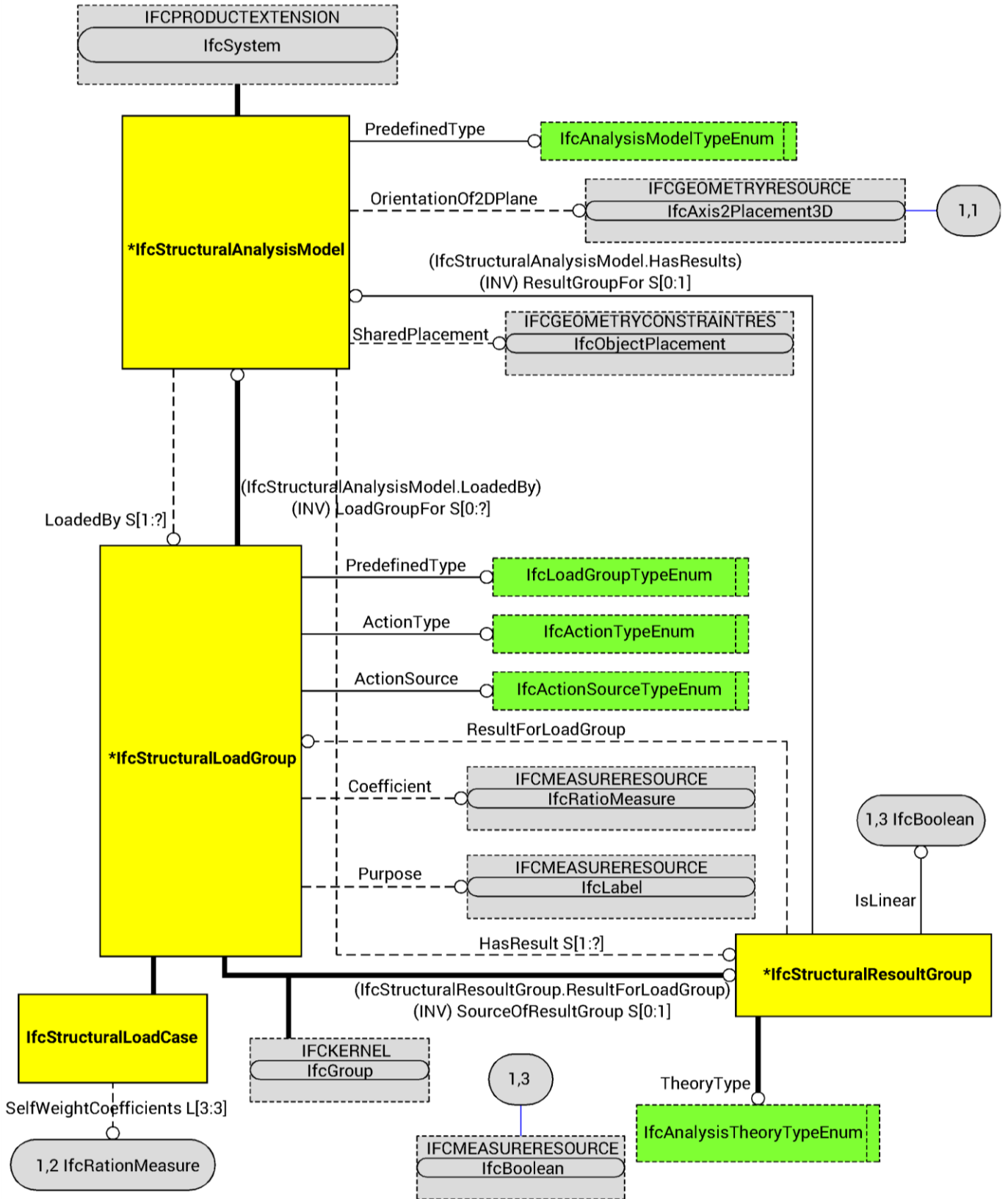


Figura 27 - IFC SAD. Modello analitico, carichi e combinazioni di carico, risultati.

## IFC STRUCTURAL ANALYSIS MODEL

-**IfcStructuralAnalysisModel**<sup>30</sup> è una *entity* utilizzata per assemblare le informazioni necessarie a rappresentare un modello di analisi strutturali; comprende alcune proprietà generali, come il tipo di analisi, fa riferimento a tutti gli *item* strutturali, a tutti i carichi e tutti i rispettivi risultati.

Funzionalità importanti per la descrizione del modello di analisi sono derivate da altre *entities* di IFC già esistenti:

Da *IfcSystem* viene ereditata la possibilità di accoppiare, se necessario, un modello di analisi a più di un *IfcBuilding* attraverso *IfcRelServicesBuilding*;

Da *IfcGroups* viene ereditata l'*inverse attribute IsGroupedBy* che punta a *IfcRelAssignsToGroups*. Questo Permette di raggruppare gli elementi e le connessioni strutturali, che appartengono a un specifico modello di analisi. I carichi non possono essere inclusi, ma sono assegnati attraverso l'*attribute IsLoadedBy*, permettendo di usare i gruppi di carico come meccanismi di raggruppamento. Solamente combinazioni di carico, se ne esiste più di una, vanno raggruppati attraverso *LoadedBy*, oppure dai casi di carico, se non esistono combinazioni di carico nel modello per le analisi. I risultati vanno collegati attraverso l'*attribute HasResults* usando i risultati di gruppi come meccanismo di raggruppamento.

Da *IfcObject* eredita l'*inverse attribute IsDecomposedBy* che punta alla *class* di relazioni *IfcRelNests* che provvede alla gerarchia tra i parziali e separati modelli di analisi.

Si prevede poi, informalmente, che se uno o più *item* strutturali siano raggruppati nella stessa *instance IfcStructuralAnalysisModel*, deve essere fornito di valore l'*attribute SharedPlacement*. Mentre l'*attribute ObjectPlacement* di tutti gli *item* strutturali raggruppati nella stessa *instance*, deve riferirsi alla stessa *instance IfcObjectPlacement*. Quest'ultima regola è necessaria per avere una rappresentazione topologica coerente, infatti gli *item* strutturali di un modello si intende debbano condividere vertici e bordi e devono, quindi,

---

30 - buildingSMART, "IFC4 - Add.2; 7.7.3.6 IfcStructuralAnalysisModel".



avere lo stesso posizionamento. Se un *item* è raggruppato in più di un modello di analisi è necessario che tutti i modelli devono sfruttare la stessa *instance IfcObjectPlacement*.

Per il funzionamento di questa *entity* sono previsti cinque *attributes*. Il primo è un *attribute* di *PredefinedType* verso *IfcAnalysisModelTypeEnum*, che è un *enumerated type* che permetta di definire il modello di analisi strutturale. Il secondo è *OrientationOf2DPlane* verso *IfcAxis2Placement3D* che, se il *type* selezionato descrive un sistema bidimensionale, definisce il piano delle analisi e la normale a questo; ciò perché *item* e attività strutturali sono sempre definite in spazio tridimensionale anche se si intende analizzarla in ambiente bidimensionale. Se, poi viene scelto il *type IN\_PLANE\_LOADING\_2D* le analisi verranno effettuate sulle proiezioni del piano definito; mentre se viene scelto il *type OUT\_PLANE\_LOADING\_2D* verrà analizzato solamente la componente dei carichi e i suoi effetti nella normale al piano, per analisi, perlopiù, di griglie di travi e di solai. Nel caso, infine, di scelta del *type LOADING\_3D* l'*attribute OrientationOf2dPlane* viene omissa. Viene poi definito l'*attribute LoadedBy* riferito a *IfcStructuralLoadGroup* che collega tutti i gruppi di carico da analizzare, mentre l'*attribute HasResults* verso *IfcStructuralResultGroup* collega tutti i risultati di gruppi disponibile per il definito modello di analisi strutturali. Infine l'*attribute SharedPlacement* verso *IfcObjectPlacement*, che deve essere comune a tutti gli *items* e le attività strutturali che sono collegata alla stessa *instance* di *IfcStructuralAnalysisModel*, stabilisce un sistema di coordinate che viene considerato come globale nelle definizioni delle varie classi di *items* e attività strutturali. È comune, ma non necessario che questo *attribute* sia *IfcLocalPlacement* il cui asse z è parallelo all'asse z del sistema di coordinate globali dato in *IfcProject* e diretto similmente “verso l'alto” oppure in direzione opposta, “verso il basso”. Per le proposizioni informali precedenti, sebbene sia formalmente un *attribute* non necessario, deve essere definito ogni volta che almeno un *IfcStructuralItem* è raggruppato nell'*instance IfcStructuralAnalysisModel*.

Infine, una proposizione formale prevede che l'*attribute ObjectType* debba essere dato se è viene utilizzato il *type USERDEFINED*.

-**IfcAnalysisModelTypeEnum** è un *enumerated type* utilizzato per distinguere tra diverse tipologie di modelli di analisi strutturali, differenziati sulla base della loro dimensionalità.

*IN\_PLANE\_LOADING\_2D*: per le analisi che vengono svolte all'interno del piano di riferimento specificato;

*OUT\_PLANE\_LOADING\_2D*: per le analisi che vengono svolte al di fuori, cioè sulla normale, del piano di riferimento;

*LOADING\_3D*: per le analisi che vengono svolte nello spazio tridimensionale;

*USERDEFINED*: per le analisi che avvengono in uno qualsiasi dei piani, come definito dell'utente;

*NOTDEFINED*: qualora non vengano fornite ulteriori informazioni sulla dimensionalità del modello di analisi considerato.

-**IfcStructuralLoadGroup** è l'*entity* utilizzato per organizzare gli effetti fisici. Utilizzando le funzioni ereditate da *IfcGroup*, *instances* di *IfcStructuralAction*, o sottoposte, e di *IfcStructuralLoadGroup*, può essere utilizzato per definire gruppi, casi e combinazioni di carico, utili alla descrizione a descrivere un sistema portante. Questo perché *IfcGroup* può, attraverso *IfcRelAssignsToGroup*, fornire il meccanismo di raggruppamento necessario; in questo modo le *instances* di *IfcStructuralAction*, appartenenti a uno specifico gruppo di carico, possono essere determinate senza ambiguità. La relazione stabilità con *IfcRelAssignsToGroupByFactor* viene utilizzato per raggruppare i casi di carichi all'interno delle combinazioni di carico, il fattore fornito è applicato assieme all'*optional attribut Coefficient*; ma, a differenza di questo che incide sul gruppo di carico, è specifico per la coppia caso di carico-combinazione di carico. Le *instances* di *IfcRelAssignsToGroupByFactor* vengono utilizzate all'interno di una combinazione di carico in quanto vi sono diversi *attribute Factor* da applicare ai casi di carico nella combinazione di carico. D'altra parte, un caso di carico può comparire in più di una combinazione di carico e avere diversi *Factor* in ogni diversa assegnazione.

Le proposizioni informali prevedono che i gruppi di carichi definiti con *LOAD\_GROUP* debbano contenere solamente *instances IfcStructuralAction*, mentre quelli definiti con *LOAD\_CASE* debbano essere definite con *instances* di *IfcStructuralLoadCase*, non direttamente dallo stesso e generico *IfcStructuralLoadGroup*. Le *instances IfcStructuralLoadCase* devono contenere solo *instances*

*IfcStructuralAction* e/o *IfcStructuralLoadGroup* definite con *LOAD\_GROUP*. I gruppi di carico definiti con *LOAD\_COMBINATION* devono contenere solamente *IfcStructuralLoadCase*.

Sono presenti sette *attributes*. Il primo è *PredefinedType* che si collega a *IfcLoadGroupTypeEnum* permettendo di definire un tipo predefinito di gruppo di carico, può essere differenziato tra gruppi di carico, casi di carico, combinazioni di carico o livelli di raggruppamento definiti dall'utente. Il secondo, *ActionType*, collegato *IfcActionTypeEnum*, permette di definire il tipo di azioni nel gruppo, è normalmente necessario se il primo *attribute* specifica *LOAD\_CASE*. Il terzo *attribute* è *ActionSource* e collega *IfcActionSourceTypeEnum*, specificando la fonte delle azioni nel gruppo, come il secondo *attribute* è normalmente impiegato quando il primo specifica *LOAD\_CASE*. L'*attribute Coefficient* mettendo in relazione *IfcRatioMeasure* specifica un fattore di carico che, se omesso è considerato non ancora conosciuto o non specificato, questo perché, se questo è necessario sia pari a 1.0 deve essere esplicitato in quanto tale. Il quinto *attribute*, *Purpose*, collega *IfcLabel* in modo che sia possibile inserire una descrizione dello scopo di questa *instance*, tra tutti è possibile specificare gli stati limiti considerati. Il sesto *attribute* è l'*inverse attribute SourceOfResultGroup* collegato a *IfcStructuralResultGroup* serve a listare quali risultati sono stati computati usando questo gruppo di carico. Similmente, l'ultimo *attribute*, anch'esso un *inverse attribute*, *LoadGroupFor* collega *IfcStructuralAnalysisModel* lista quali sono i modelli di analisi in cui il gruppo di carico è utilizzato.

La proposizione formale, la regola *HasObjectType*, specifica che l'*attribute ObjectType* deve essere dato se il valore dato a uno dei primi tre *attribute*, cioè *PredefinedType*, *ActionType* oppure *ActionSource* è definito dall'utente attraverso *USERDEFINED*.

**-IfcLoadGroupTypeEnum** è un *enumerated type* utilizzato per distinguere tra diversi livelli di raggruppamenti di carico. Permette di differenziare tra gruppi di carico, casi di carico e combinazioni di carico. I valori presenti sono:

*LOAD\_GROUP*: raggruppa le *instances* sottostanti ad *IfcStructuralAction* e deve essere utilizzato come contenitore per carichi raggruppati assieme per specifici scopi, come carichi che sono parte di speciali modelli di carico;

*LOAD\_CASE*: raggruppa i gruppi di carico specificati con *LOAD\_GROUP* e le *instances* sottostanti ad *IfcStructuralAction*; dovrebbe essere utilizzato come contenitore per carichi con la stessa origine;

*LOAD\_COMBINATION*; un livello intermedio tra i due precedenti, è obsoleto e disapprovato; prima dell'introduzione di *IfcRelAssignsToGroupByFactor* provvedeva un fattore con cui uno o più casi di carico, descritti con *LOAD\_CASE*, si verificano in una combinazione di carico, descritta con *LOAD\_COMBINATION*;

*USERDEFINED* è invece un livello di raggruppamento che non segue la gerarchia standard dei tipi di carico;

*NOTDEFINED* viene utilizzato quando ancora il livello di raggruppamento è ancora incognito.

-**IfcActionTypeEnum** è un *enumerated type* utilizzato per distinguere tra i possibili tipi di azione ad alto livello. Può essere utilizzato per una definizione automatica di combinazioni di carico e per dimensionamenti. Le possibilità di selezione e i loro acronimi sono adottati dagli Eurocodici:

PERMANENT\_G;

VARIABLE\_Q;

EXTRAORDINARY\_A;

USERDEFINED;

NOTDEFINED;

-**IfcActionSourceTypeEnum** è un *enumerated type* che contiene possibili fonti di azione:

DEAD\_LOAD\_G;

COMPLETION\_G1;

LIVE\_LOAD\_Q;

SNOW\_S;

WIND\_W;

PRESTRESSING\_P;  
SETTLEMENT\_U;  
TEMPERATURE\_T;  
EARTHQUAKE\_E;  
FIRE;  
IMPULSE;  
IMPACT;  
TRANSPORT;  
ERECTION;  
PROPPING;  
SYSTEM\_IMPERFECTION;  
SHRINKAGE;  
CREEP;  
LACK\_OF\_FIT;  
BUOYANCY;  
ICE;  
CURRENT;  
WAVE;  
RAIN;  
BRAKES;  
USERDEFINED;

NOTDEFINED.

-**IfcStructuralLoadCase** questa *instance* è comunemente utilizzata per raggruppare in un gruppo di carico, carichi dalla stessa fonte.

È previsto, poi, un *attribute*: *SelfWeightCoefficient*, che specifica un vettore a tre, obbligatoriamente tre, valori per cui i carichi dovuti al peso, cioè i pesi propri degli elementi, devono essere inclusi nel gruppo di carico. Questi carichi non sono esplicitamente modellati attraverso *IfcStructuralAction*, perché devono essere calcolati sulla base della geometria, della sezione e dei materiali di ogni elemento strutturale. Le tre componenti del vettore del peso proprio corrispondono al direzionamento del peso secondo le direzioni x, y e z delle cosiddette coordinate globali, del modello di analisi. Per esempio, se il posizionamento definisce un asse z diretto verso l'alto e concorde in direzione al sistema di coordinate globali, allora i coefficienti del peso proprio saranno (0,0,-1) all'interno del gruppo di carichi. Il coefficiente complessivo ereditato dall'*attribute Coefficient* non deve essere applicato a questo *attribute* della stessa *instance* di *IfcStructuralLoadCase*, perché si applica soltanto ad azioni e gruppi di carico raggruppati sotto la condizione di carico, non al peso proprio calcolato della condizione di carico stessa. La proposizione formale *IsLoadCasePredefinedType* viene utilizzata per dire che una *instance* di *IfcStructuralLoadCase* non può essere di altro tipo che una condizione di carichi.

-**IfcStructuralResultGroup** è una *instance* di *IfcStructuralResultGroup* utilizzata per raggruppare i risultati dei calcoli delle analisi strutturali e per trattenere i collegamenti tra i sottostanti gruppi strutturali basici. Le funzionalità base per il raggruppamento ereditate da *IfcGroup* sono utilizzate per raccogliere le *instances* da *IfcStructuralReaction* o dai suoi sottoposti.

Sono presenti quattro *attribute*: il primo è *TheoryType* che, collegando *IfcAnalysisTheoryTypeEnum*, permette di specificare la teoria di analisi utilizzata per ottenere i rispettivi risultati; il secondo è *ResultForLoadGroup* che riferisce per quale *instance* di *IfcStructuralLoadGroup* sono ottenuti i risultati qui contenuti e, per fare ciò, si appoggia ad *IfcStructuralLoadGroup*; il terzo *attribute* è *IsLinear*, che, con una funzione booleana *IfcBoolean*, permette riconoscere facilmente se è stata svolta un'analisi lineare e se è, quindi, possibile sovrapporre semplicemente i risultati; il quarto e ultimo *attribute* è *ResultGroupFor* e

l'*inverse HasResults* che collega un'*instance IfcStructuralAnalysisModel* per riportare a quale modello di analisi strutturale sono relativi questi risultati.

Infine, la proposizione formale *HasObjectType* prevede che l'*attribute ObjectType* debba essere dato se viene utilizzata una teoria di analisi personalizzata e definita dall'utente, attraverso l'utilizzo del set *USERDEFINED*.

**-IfcAnalysisTheoryTypeEnum** è un *enumerated type* utilizzato per distinguere tra diversi tipi di metodi di analisi strutturale, incluse la teoria del primo, secondo o terzo ordine, a seconda dell'ordine di grandezza delle deformazioni, la teoria pienamente non lineare. Le enumerazioni presenti sono:

-*FIRST\_ORDER\_THEORY*: per descrivere il metodo d'analisi che considera il sistema strutturale dalle deformazioni nulle o assimilabili a tali;

-*SECOND\_ORDER\_THEORY*: per descrivere il metodo d'analisi che considera il sistema strutturale dalle deformazioni piccole, che mettono in gioco forze di un ordine di grandezza inferiore rispetto alle forze direttamente applicate, ma che non sono approssimabili a zero;

-*THIRD\_ORDER\_THEORY*: per descrivere il metodo d'analisi che considera il sistema strutturale dalle deformazioni grandi, che mettono in gioco forze dello stesso ordine di grandezza rispetto alle forze direttamente applicate;

-*FULL\_NONLINEAR\_THEORY*: per descrivere il metodo d'analisi che considera la non linearità della geometria o dei materiali, come ad esempio il comportamento plastico o elasto-plastico;

-*USERDEFINED*: per descrivere un metodo d'analisi non previsto dalla standardizzazione IFC ma descritto dall'utente;

-*NOTDEFINED*: per descrivere un metodo d'analisi ancora non meglio specificato.

# **3D. structural load resource**



## PREMESSA

Per la completa descrizione di un sistema strutturale non è però sufficiente il *data schema Structural Analysis Domain*<sup>31</sup>. Questo è quindi affiancato e supportato da una specifica risorsa strutturale e il suo proprio *data schema*: lo *Structural Load Resource*.

Nella premessa, lo *Structural Load Resource* è così descritto:

«The *IfcStructuralLoadResource* holds the definitions of structural loads, results, boundary conditions, and connection conditions. These definitions specify the force, moment, stiffness, displacement and other factors.

The load resource definitions are used in other parts of the IFC model, particularly by the definition of structural actions and reactions that places the load definitions into 2D or 3D space.

NOTE: The current version of IFC does not define dynamic loads, it is restricted to static loads.»

Ossia che la risorsa in questione contiene le definizioni di carichi strutturali, i risultati le condizioni di vincolo e le condizioni di connessioni, questi ultimi due casi, in altre parole, descrivono le condizioni di vincolo interne ed esterne. Queste definizioni, tra le altre cose specificazione forze, momenti, rigidità, e spostamenti. Le definizioni di questa risorsa, che come le altre risorse non è assolutamente autosufficiente, sono utilizzate in altre parti del modello IFC, in particolare per la definizione di azioni e reazioni che posizionano la definizione del carico nello spazio a due oppure a tre dimensioni. In pratica questa risorsa mette a disposizione dello *Structural Analysis Domain*, una serie di entità per descrivere concretamente i vari tipi di carico, di forze, di risultati, di vincoli e di condizioni al contorno.

Come per lo *Structural Analysis Domain* viene riportata l'incompletezza, allo stato attuale, dello standard per quanto riguarda il mondo delle strutture e delle relative analisi: difatti ad oggi è compiutamente descritto solamente ciò che riguarda i carichi statici, mancando completamente ciò che riguarda i carichi dinamici. Non sarà necessario implementare solamente i carichi dinamici, ma non rientra in

---

31 - buildingSMART, "IFC4 - Add.2; 8.19 IfcStructuralLoadResource".

questo *data schema* la definizione di proprietà dinamiche e quanto è già stato sottolineato come mancante nella premessa dello *Structural Analysis Domain*.

Anche lo *Structural Load Resource* è concettualmente organizzato in “macro-classi”, per meglio supportare il dominio strutturale. Queste tre classi servono a descrivere carichi o risultati, condizioni di vincolo esterne rigide o cedevoli e condizioni di congruenza per i nodi. Questo si concretizza in tre grandi *abstract entities*: *IfcStructuralLoad*, *IfcBoundaryCondition* e *IfcStructuralConnectionCondition* rispettivamente.

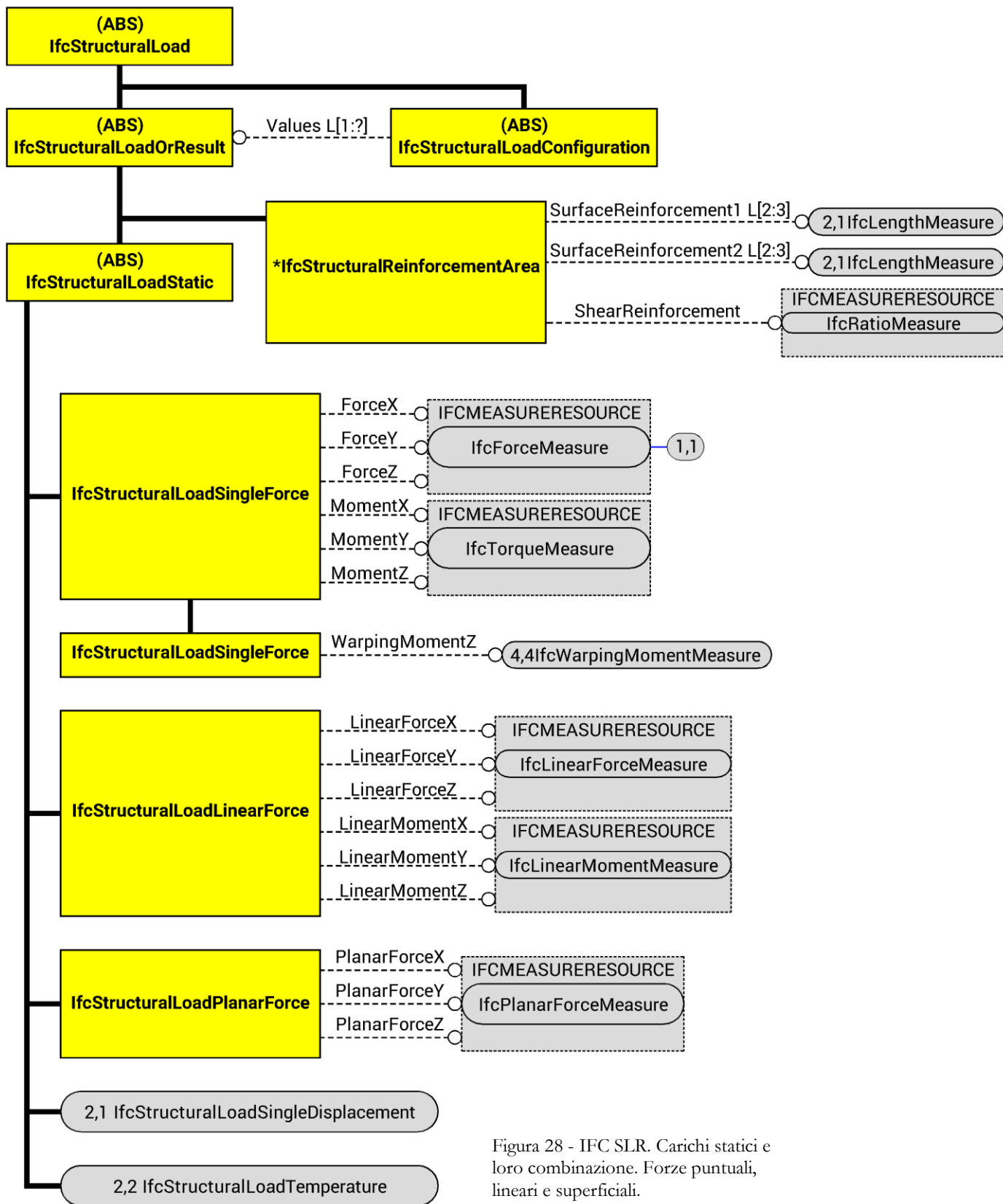


Figura 28 - IFC SLR. Carichi statici e loro combinazione. Forze puntuali, lineari e superficiali.

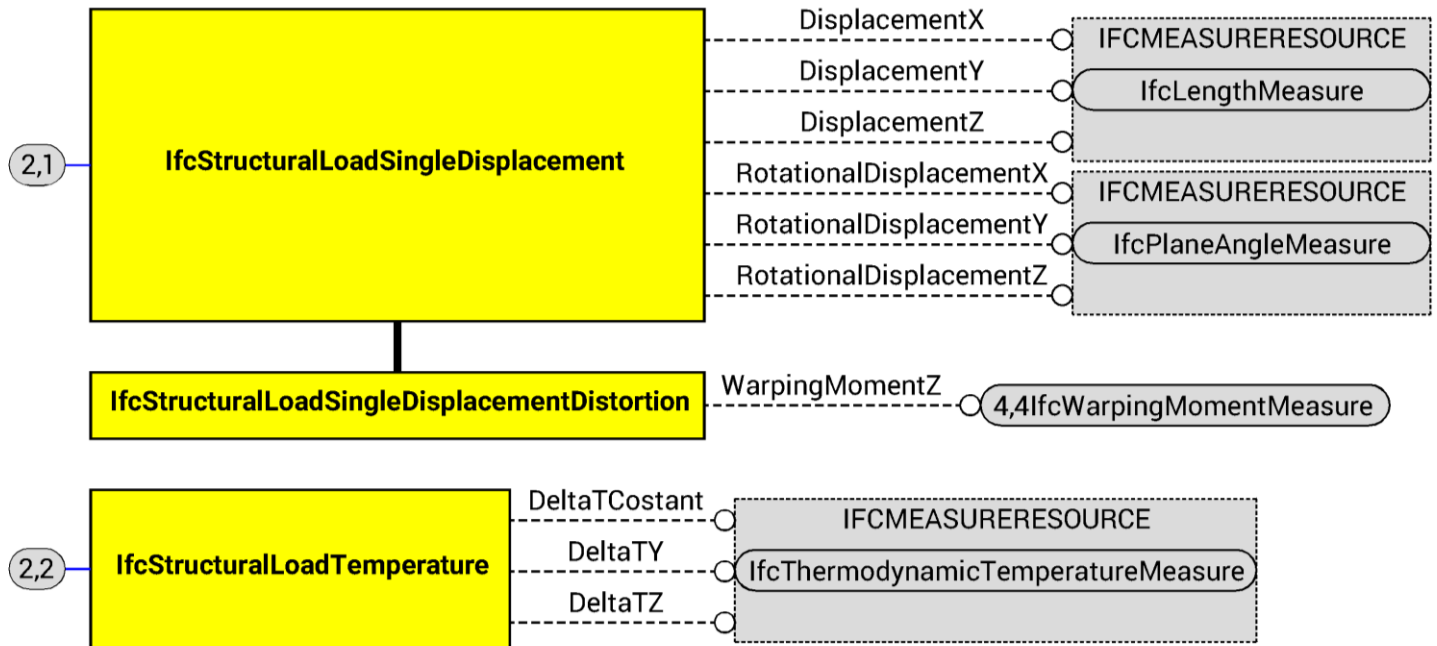


Figura 29 - IFC SLR. Carichi statici. Spostamenti impressi e carichi di temperatura.

## IFC STRUCTURAL LOAD

-La prima *abstract entity* che consideriamo è **IfcStructuralLoad**<sup>32</sup>. È un *supertype*, in pratica, contenente o tutti i carichi di azione e di reazione o alcuni requisiti risultanti dalle analisi strutturali o alcuni prerequisiti che influenzano le analisi strutturali. È possibile definire una stringa di testo come nome per ogni singola *entity* che viene utilizzata. Sono presenti due *subentities*.

-**IfcStructuralLoadConfiguration** è la prima delle due *subentities* che fanno riferimento a *IfcStructuralLoad* ed è una classe che serve a combinare uno o più valori di carico o di risultato in una configurazione mono o bidimensionale. Tutti gli *item* listati attraverso l'*attribute Value* devono essere dello stesso tipo. Se i carichi o i risultati comprendono l'attività strutturale di una curva, il posizionamento monodimensionale deve essere dato, misurato localmente lungo la curva e non deve eccedere i limiti dell'attività della curva. I campioni di carico e le corrispettive posizioni devono essere

32 - buildingSMART, "IFC4 - Add.2; 8.19.3.9 IfcStructuralLoad".

date in ordine crescente. Se i carichi o i risultati comprendono, invece l'attività di una curva, il posizionamento bidimensionale deve essere dato e misurato localmente sulla superficie, senza eccederne i limiti, attraverso gli assi locali x e y.

Come già accennato, a descrivere queste combinazioni di carico o di risultati provvede l'*attribute Value*, che fisicamente fa una lista di valori. Mentre l'*attribute Locations* restituisce la posizione dei modelli di carico o di risultato, dati in coordinate locali definite dall'*instance* presa come riferimento. Questo *attribute* è facoltativo per le configurazioni nelle quali le posizioni sono conosciuti implicitamente da livelli di definizioni superiori. Qualora però siano indicate le posizioni, devono esserci tanti *items* di *Locations* quanti sono gli *items Value*.

-**IfcStructuralLoadOrResult** è la seconda delle due *subentities* che fanno riferimento a *IfcStructuralLoad* ed è a sua volta un'*abstract entity* che serve a descrivere semplici classi di carico o di risultato. È collegata a due *subentities* a sua volta.

-**IfcSurfaceReinforcementArea** è la prima delle due *subentities* che fanno riferimento a *IfcStructuralLoadOrResult* e descrive le aree di armatura richieste o fornite a un elemento bidimensionale. Si ricorda che i parametri di disegno, come copriferro, profondità effettiva, orientamento della mesh, barre nelle due o anche tre direzioni eccetera, non vengono specificate nella risorsa, in *IfcStructuralLoadResource*, ma nel dominio corrispondente ed è lì che devono essere specificate, all'interno dell'elemento strutturale.

L'armatura in questione viene specificata attraverso tre *attributes*: il primo descrive l'armatura alla faccia del membro strutturale posizionato in corrispondenza della direzione positiva dell'asse locale z e specifica, appunto un'area per unità di lunghezza e, per questo, la sua unità di misura è espressa in metri. È necessario specificare l'area di armatura per almeno due e massimo tre direzioni delle barre. Il secondo *attribute*, similmente al primo, descrive l'armatura alla faccia del membro strutturale posizionato, però, in corrispondenza della direzione negativa dell'asse z e specifica, nuovamente, un'area di armatura per unità di lunghezza, esprimendola in metri. Di nuovo, è richiesto che l'area di ferro sia specificata per almeno due e massimo tre direzioni delle barre d'armatura. L'ultimo *attribute* serve, invece, a descrivere l'armatura a taglio, specificandola come un'area di armatura per area di

sezione trasversale della superficie ed esprimendola, conseguentemente, in metri. È necessario che almeno uno degli *attributes* delle aree di ferro vengano specificato e che nessuno dei tre sia, invece, minore di zero.

-**IfcStructuralLoadStatic** è la seconda *entities* che si riferisce a *IfcStructuralLoadOrResult* ed è una grande *abstract superentity* per tutte le azioni o reazioni statiche che possono essere definite. Sono previste azioni o reazioni puntuali, cioè forze e momenti concentrati, lineari, cioè forze e momenti dimensionalmente distribuiti su di una curva, e planari, cioè forze distribuiti bidimensionalmente; sono, inoltre previsti azioni, o reazioni, di spostamento e carichi dati dalla temperatura.

-**IfcStructuralLoadSingleForce** è una delle cinque *entity* che definiscono *IfcStructuralLoadStatic* ed è un'*instance* che deve essere utilizzata per definire le forze e i momenti di un'azione strutturale agente su di un singolo punto. Presenta sei *attributes*, tre per descrivere il valore della forza lungo gli assi locali x, y e z e tre per descrivere il valore del momento attorno agli assi locali x, y e z. Questi *attributes* sono chiamati *ForceX*, *ForceY*, *ForceZ*, *MomentX*, *MomentY*, e *MomentZ*. Presenta, infine, una *subentity*.

-**IfcStructuralLoadSingleForceWarping** è un'*entity* che fa riferimento a *IfcStructuralLoadSingleForce* e che deve essere usata per definire un'azione che agisce su un singolo punto. In più va definito un valore di resistenza torsionale secondaria, anche detta resistenza a ingobbamento contrastato.

-**IfcStructuralLoadLinearForce** è una delle cinque *entity* che definiscono *IfcStructuralLoadStatic* ed è un'*instance* che viene utilizzata per definire azioni distribuite su di una curva. Presenta sei *attributes* tre per descrivere il valore della forza lineare lungo gli assi locali x, y e z e tre per descrivere il valore del momento lineare attorno gli assi locali x, y e z. Sono chiamati *ForceLinearX*, *ForceLinearY*, *ForceLinearZ*, *MomentLinearX*, *MomentLinearY*, e *MomentLinearZ*.

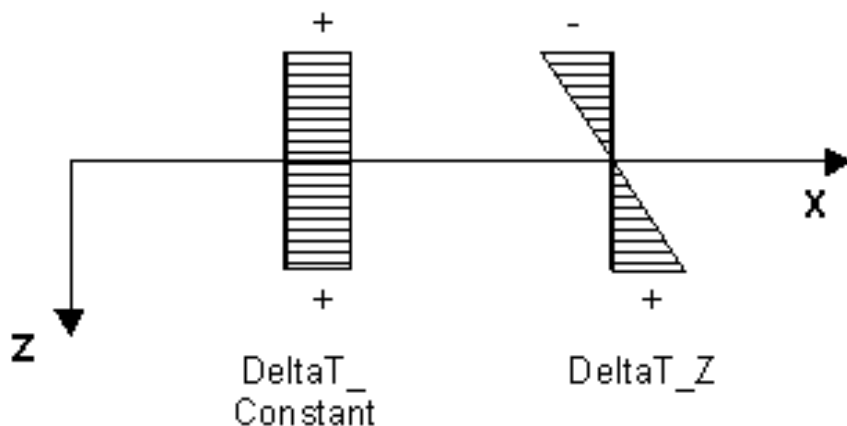
-**IfcStructuralLoadPlanarForce** è un'*entity*, sottoposta di *IfcStructuralLoadStatic*, che viene utilizzata per definire azioni su superfici. Presenta tre *attributes* che definiscono il valore della forza lungo gli assi locali x, y e z; questi *attributes* sono chiamati *PlanarForceX*, *PlanarForceY* e *PlanarForceZ*.

-**IfcStructuralLoadSingleDisplacement** è la quarta *subentity* di *IfcStructuralLoadStatic* che definisce delle deformazioni, siano traslazionali e rotazionali. Per fare questo sfrutta sei *attributes*, tre per

descrivere lo spostamento lungo i tre assi locali  $x$ ,  $y$  e  $z$ , tre per descrivere la rotazione attorno ai tre assi locali. I nomi che hanno sono *DisplacementX*, *DisplacementY*, *DisplacementZ*, *RotationalDisplacementX*, *RotationalDisplacementY*, e *RotationalDisplacementZ*. Presenta una *subentity*.

-**IfcStructuralLoadSingleDisplacementDistortion** è la *subentity* di cui si accennava prima e serve a definire, una deformazione fuori piano della sezione trasversale, cioè un'ingobbamento. Per questo sfrutta un'attribute chiamato *Distortion*.

-**IfcStructuralLoadTemperature** è l'ultima *subentity* di *IfcStructuralLoadStatic* e viene utilizzata per definire azioni causate da cambi di temperatura. Come in figura si spiega, il cambio di temperatura è dato con un valore costante che è applicato all'intera sezione e valori per le differenze di temperatura tra le fibre più esterne della sezione. Il carico della temperatura è definito con tre *attributes* il primo chiamato *DeltaTConstant* cioè una variazione di temperatura che ha effetto sull'intera sezione dell'*item* strutturale oppure su una porzione uniforme di un non uniforme cambio di temperatura, per convenzione un valore positivo descrive un aumento della temperatura, cioè un valore positivo causa un allungamento oppure una compressione nell'elemento strutturale a seconda dei vincoli presenti. Con il secondo *attributes*, *DeltaTY*, viene specificata una variazione non uniforme di temperatura attraverso una differenza di temperatura tra le fibre più esterne nella direzione positiva dell'asse locale  $y$  e le fibre opposte, cioè quelle più esterne nella direzione della  $y$  negativa. Un valore positivo del



questo *attribute* induce una curvatura negativa dell'elemento attorno all'asse locale  $z$ , oppure una flessione positiva attorno a  $z$ , a seconda dei vincoli. Il terzo e ultimo *attribute* descrive invece un cambio di temperatura non uniforme simile a quello appena descritto, ma nell'altra

Figura 30 - IFC SLR. Carico di temperatura. Specificazione variazioni di temperatura generanti azioni.

direzione: ovvero il caso in cui sia data una differenza di temperatura tra le fibre più estreme della sezione in direzione positiva e negativa dell'asse locale  $z$ . Un valore positivo di questa differenza di temperatura induce una curvatura positiva attorno all'asse locale  $y$  oppure una flessione negativa attorno all'asse  $y$ , a seconda dei vincoli dell'elemento strutturale.



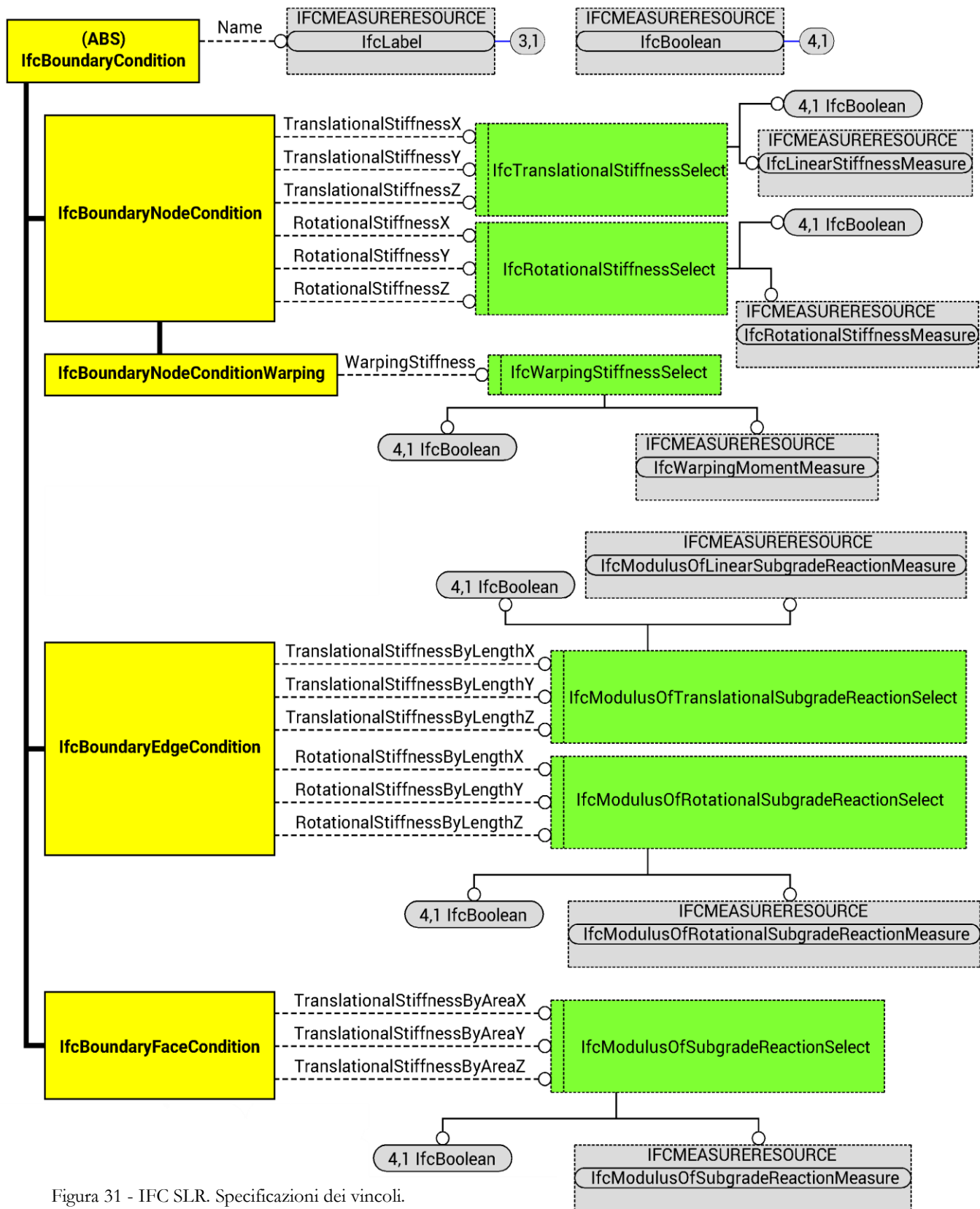


Figura 31 - IFC SLR. Specificazioni dei vincoli.

## IFC BOUNDARY CONDITION

-La seconda *abstract entity* che consideriamo è **IfcBoundaryCondition**<sup>33</sup>, *supertype* che può essere applicato alle definizioni di connessione strutturali, sia direttamente alla connessione, come un giunto, sia alla relazione tra l'elemento strutturale e la connessione; definisce a tutti gli effetti un vincolo esterno. È presente una stringa di testa come *attribute* per poter definire un nome per la condizione di vincolo. Per poter descrivere il tutto, sono presenti tre *subtypes* che descrivono i vincoli in base alla loro dimensionalità. Si parlerà di rigidezza alla traslazione lungo gli assi e alla rotazione attorno agli assi: si dovrà intendere, ovviamente, intendere come rigidezza a compressione e trazione, cioè come modulo di Young, la rigidezza alla traslazione lungo l'asse principale, mentre come rigidezza tagliante, cioè come modulo a taglio, le altre due; come rigidezza torsionale la rigidezza rotazionale attorno all'asse principale e come rigidezza flessionale le altre due. Sembra quindi ovvio, e da implementare, che un vincolo possa avere, comportamento, oltre che lineare elastico, anche bilineare, magari con diverse tensioni di snervamento tra compressione e trazione. Questa mancanza non è tra quelle esplicitate nella premessa del *data schema*.

-**IfcBoundaryNodeCondition** è la prima *entity* per descrivere una condizione di vincolo esterna con un comportamento linearmente elastico. Può essere applicata a supporti e connessioni puntuali. Per svolgere il suo compito presenta sei *attributes* che fanno riferimento a due *enumerated types* e una *subentity*. Gli *attributes* sono *TraslationalStiffnessX*, *TraslationalStiffnessY*, *TraslationalStiffnessZ*, *RotationalStiffnessX*, *RotationalStiffnessY*, e *RotationalStiffnessZ*. I due *types* sono *IfcTraslationalStiffnessSelect* e *IfcRotationalStiffnessSelect*. Questo permette di descrivere la rigidezza rotazionale e traslazionale lungo e attorno i tre assi locali x, y e z definiti dall'*instance* presa come riferimento. La *subentity* permette di descrivere l'eventuale rigidezza torsionale secondaria, quella relativa all'ingobbamento contrastato.

-**IfcTraslationalStiffnessSelect** misura la rigidezza lineare traslazionale: è possibile assegnare una rigidezza infinita per poter descrivere un nodo rigido alla traslazione, oppure una rigidezza nulla per descrivere un vincolo a doppia biella, oppure ancora una rigidezza alla traslazione finita; questo è

---

33 - buildingSMART, "IFC4 - Add.2; 8.19.3.1 IfcStructuralBoundaryCondition".

possibile attraverso la selezione di una semplice preposizione di logica booleana, dove TRUE denota l'infinita rigidità, mentre FALSE la rigidità nulla, mentre con una seconda selezione possibile si può scrivere un valore numerico di una rigidità lineare elastica finita. Sono quindi concretamente possibili due sole scelte, la prima sarà *IfcBoolean*, con cui descrivere i casi di rigidità infinita o nulla, la seconda *IfcLinearStiffnessMeasure*, per descrivere la rigidità finita.

-**IfcRotationalStiffnessSelect** è un *type* per misurare, invece, la rigidità rotazionale. Come nel caso precedente, con scegliendo *IfcBoolean* si potrà assegnare la proprietà TRUE e descrivere un nodo rigido alla rotazione, dall'infinita rigidità rotazionale, mentre con FALSE, una cerniera, dalla rigidità rotazionale nulla. Oppure, scegliendo *IfcRotationalStiffnessMeasure* si potrà assegnare una rigidità rotazionale finita al vincolo.

-**IfcBoundaryNodeConditionWarping** è la *subentity* che descrive una condizione di vincolo esterno linearmente elastico, compresa una rigidità torsionale secondaria elastica. Per fare ciò presenta un *attribute* all'*enumerated type* *IfcWarpingStiffnessSelect* che permette di stabilire questo valore.

-**IfcWarpingStiffnessSelect** permette appunto di meglio specificare la rigidità torsionale secondaria attraverso due possibili scelte, similmente agli *enumerated types* definiti per descrivere la rigidità alla traslazione e alla rotazione. Scegliendo *IfcBoolean* si può assegnare TRUE per descrivere una rigidità infinita, oppure si può assegnare FALSE, per descrivere il caso in cui sia nulla. Oppure si può scegliere *IfcWarpingMomentMeasure* per assegnare un valore numerico che denoti una rigidità finita lineare elastica alla rotazione torsionale secondaria.

-**IfcBoundaryEdgeCondition** è la seconda *entity* per descrivere una condizione di vincolo esterna con un comportamento linearmente elastico. Può essere applicata a supporti e connessioni monodimensionali. Presenta sei *attributes* con due *enumerated types*: gli *attributes* si chiamano *TranslationalStiffnessByLengthX*, *TranslationalStiffnessByLengthY*, *TranslationalStiffnessByLengthZ*, *RotationalStiffnessByLengthX*, *RotationalStiffnessByLengthY*, e *RotationalStiffnessByLengthZ*. I primi tre si riferiscono all'*enumerated type* *IfcModulusOfTranslationalSudgradeReactionSelect* per poter descrivere la rigidità alla traslazione lungo gli assi locali x, y e z definiti dall'*instance* monodimensionale di

riferimento, mentre gli ultimi tre si riferiscono a *IfcModulusOfRotationalSubgradeReactionSelect* per poter descrivere la rigidezza alla rotazione attorno agli assi locali x, y e z.

**-IfcModulusOfTraslationalSubgradeReactionSelect** è l'*enumerated type* utilizzato per definire il valore della rigidezza alla traslazione del sottofondo (subgrade) che esprime la rigidezza traslazionale di strato per unità di lunghezza di un *item* strutturale monodimensionale. Se si sceglie il valore booleano si può esprimere una rigidezza alla traslazione infinita attraverso l'assegnazione della variabile TRUE, oppure una rigidezza nulla, quindi la possibilità di traslare, con la variabile FALSE. Altrimenti, scegliendo il *type IfcModulusOfLinearSubgradeReactionMeasure* è possibile una rigidezza finita lineare elastica.

**-IfcModulusOfRotationalSubgradeReactionSelect** è l'*enumerated type* che permette di definire il modulo di sostrato rotazionale che esprime rigidezza rotazionale del sostrato per unità di lunghezza di un elemento strutturale monodimensionale. Se si sceglie il valore booleano si può esprimere una rigidezza alla rotazione infinita attraverso l'assegnazione della variabile TRUE, oppure una rigidezza nulla, quindi la possibilità di ruotare, con la variabile FALSE. Altrimenti, scegliendo il *type IfcModulusOfRotationalSubgradeReactionMeasure* si può assegnare una rigidezza finita lineare elastica.

**-IfcBoundaryFaceCondition** è la terza e ultima *entity* per descrivere una condizione di vincolo esterna o interna con un comportamento linearmente elastico. Può essere applicata a supporti e connessioni bidimensionali. Presenta, a differenza delle *entities* simili ma di diversa dimensionalità, solamente tre *attributes* utili a descrivere la rigidezza alla traslazione lungo i tre assi locali x, y e z definite dalla *instance* presa come riferimento; questi sono *TranslationalStiffnessByAreaX*, *TranslationalStiffnessByAreaY* e *TranslationalStiffnessByAreaZ* e fanno riferimento a un unico *enumerated type*.

**-IfcModulusOfSubgradeReactionSelect** è l'*enumerated type* che esprime BEDDING per unità di area di un *item* strutturale bidimensionale. Assegnando *IfcBoolean* e assegnando la variabile TRUE è possibile definire una rigidezza infinita, mentre assegnando la variabile FALSE denota una rigidezza nulla, cioè la possibilità di traslare liberamente. Assegnando invece *IfcModulusOfSubgradeReactionMeasure* si può assegnare un valore numerico che denota una rigidezza finita lineare elastica.

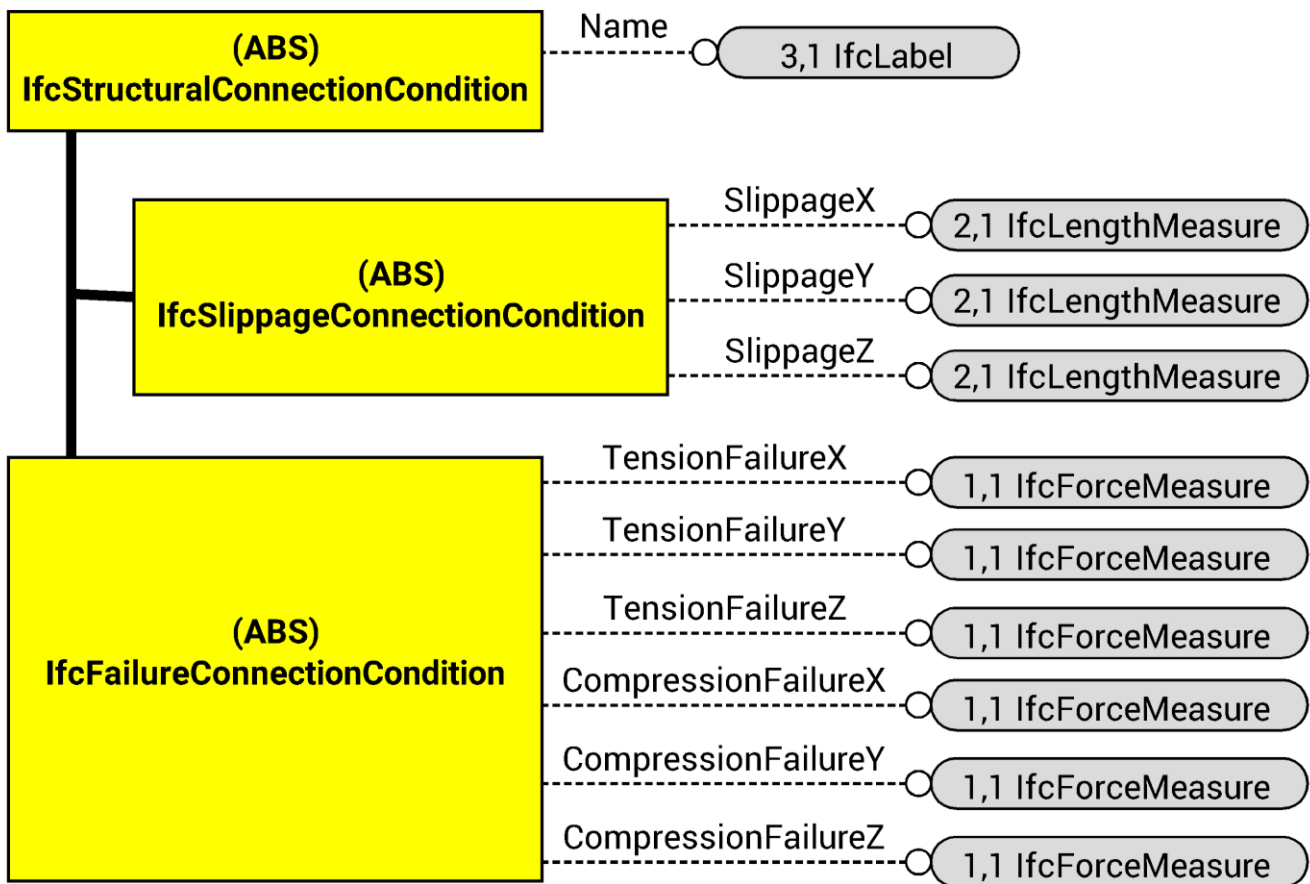


Figura 32 - IFC SLR. Condizione di slittamento e di rottura dei vincoli.

## IFC STRUCTURAL CONNECTION CONDITION

La terza e ultima *abstract entity* che consideriamo è **IfcStructuralConnectionCondition**<sup>34</sup>, *entity* il cui unico *attribute* è una stringa di testo attraverso la quale dare un nome alla condizione del vincolo, in quanto, attraverso le due *subentities* è possibile descrivere, quando necessario, utili proprietà accessorie dei vincoli, quali la resistenza ultima e lo slittamento.

34 - buildingSMART, "IFC4 - Add.2; 8.19.3.8 IfcStructuralConnectionCondition".

**-IfcSlippageConnectionCondition** questa *entity*, infatti, descrive lo slittamento di una condizione di vincolo interna o esterna. Ossia descrive quello spostamento relativo che può avvenire in un vincolo interno o esterno prima che le reazioni che questo esercita siano esplicitate. Può essere applicata a ogni tipo di connessione e vincolo, sia puntuale, sia monodimensionale, sia bidimensionale. Presenta tre *attributes* che descrivono lo slittamento lungo i tre assi locali x, y e z, sono *SlippageX*, *SlippageY* e *SlippageZ*.

**-IfcFailureConnectionCondition** è una *entity* che descrive le forze per le quali un vincolo interno o esterno cede. Questa *entity* può essere applicata solamente a elementi di connessione e a vincoli puntuali. Presenta sei *attribute* che descrivono le forze, di trazione o di compressione, lungo gli assi locali di riferimento x, y e z per le quali il vincolo cede, sono *TensionFailureX*, *TensionFailureY*, *TensionFailureZ*, *CompressionFailureX*, *CompressionFailureY*, e *CompressionFailureZ*.



## **3E. a completare**



## SHARED BUILDING ELEMENTS DOMAIN<sup>35</sup>

Per completare il quadro dell'informazione strutturale in ambito BIM così come standardizzata oggi da IFC, diremo che lo *Shared Building Elements* descrive gli elementi di una costruzione, come pilastri, travi, solai, eccetera e fa parte dell'*Interoperability Layer*. Presenta infatti al suo interno *entity* come *IfcBeam*, *IfcColumn* e *IfcSlab*, e *IfcWall*. Infatti questi elementi sono condivisi tra le professioni coinvolte in un progetto a causa della loro dimensione fisica e della loro funzione strutturale. In concreto il dominio è costruito come due grandi estensioni delle *superentity* dell'*IfcProductExtension Domain*: *IfcBuildingElement* e *IfcBuildingElementType*. Cioè sono definite molte *subentity* atte a descrivere, come da nome, l'elemento della costruzione e il tipo di elemento. Ad esempio è possibile descrivere un elemento strutturale e architettonico come una colonna sfruttando *IfcColumn* come *subentity* di *IfcBuildingElement*; nel caso in cui rispetti certe regole e sia piuttosto semplice, è presente una *subentity* chiamata *IfcColumnStandardCase* atta a descriverla. *IfcColumn* ha determinate regole riguardo i materiali che lo costituiscono, riguardo la rappresentazione geometrica degli assi, riguardo la costruzione del solido geometrico che lo costituisce. Presenta, poi, un corrispettivo in *IfcColumnType*, *subentity* di *IfcBuildingElementType*, che, collegata attraverso l'*attribute PredefinedType* ad *IfcColumn* stesso, serve a descrivere il tipo di elemento attraverso l'*enumerated type IfcColumnTypeEnum*; in questo caso o una colonna, con *COLUMN*, che resiste a compressione e, eventualmente, anche ad azioni trasversali, oppure un pilastro, attraverso *PILASTER*, inserito in un muro che, a differenza della colonna, può anche avere funzione estetica, oppure, ancora un generico elemento lineare, al solito definito dall'utente attraverso *USERDEFINED*, o, infine, non ancora specificato con *NOTDEFINED*.

Con questa impostazione sono descritti elementi di tutto il costruito: facciate continue, finestre, porte, comignoli, scossaline, coperture in genere, frangisole, tetti, corrimani, scale, rampe, solai, colonne, travi, muri o generici elementi mono e bidimensionali non meglio specificati. Tutto ciò fa il paio con i corrispetti *item* che ne descrivono la tipologia.

---

35 - buildingSMART, "IFC4 - Add.2; 6.1 IfcSharedBldgElements".

Ogni *entity* presenta *attributes* atti a descriverne le caratteristiche particolari.

Per terminare la disanima di questo dominio, affrontiamo le *relationship class*, che servono per lo più a mettere in relazione gli ambienti e i locali, del modello: *IfcRelConnectsPathElement*, *IfcRelCoversSpaces*, *IfcRelCoversBldgElements*. La prima relazione due ambienti, la seconda definisce quale sia l'elemento che copre uno spazio, come un tetto o un solaio, l'ultima quale elemento copra un altro elemento, come ad esempio le scossaline.

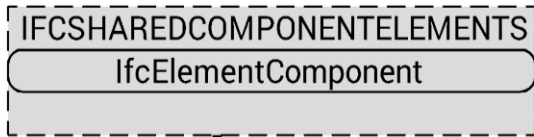
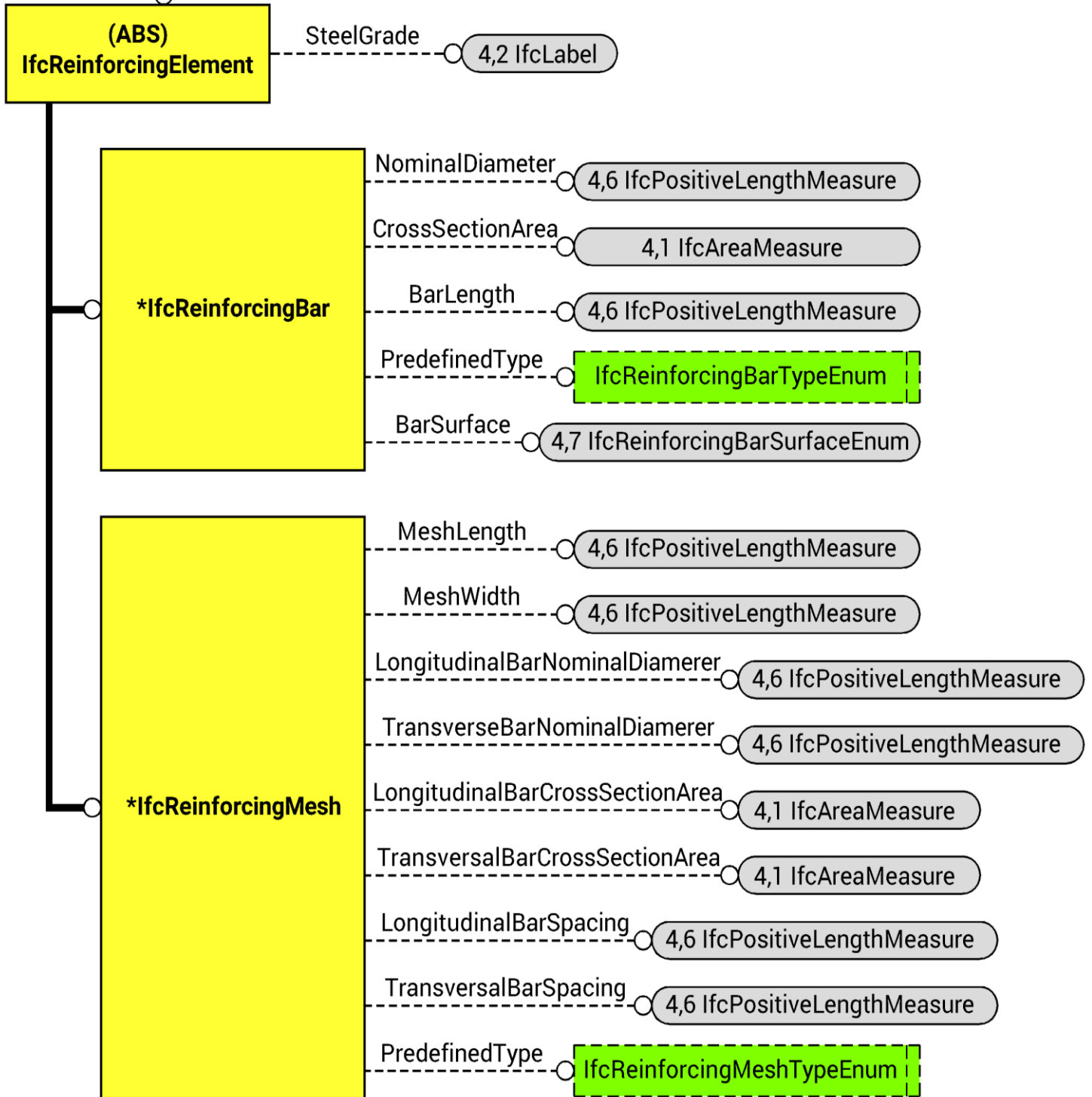


Figura 33 - IFC SED. Barre e reti d'armatura.



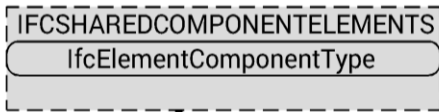
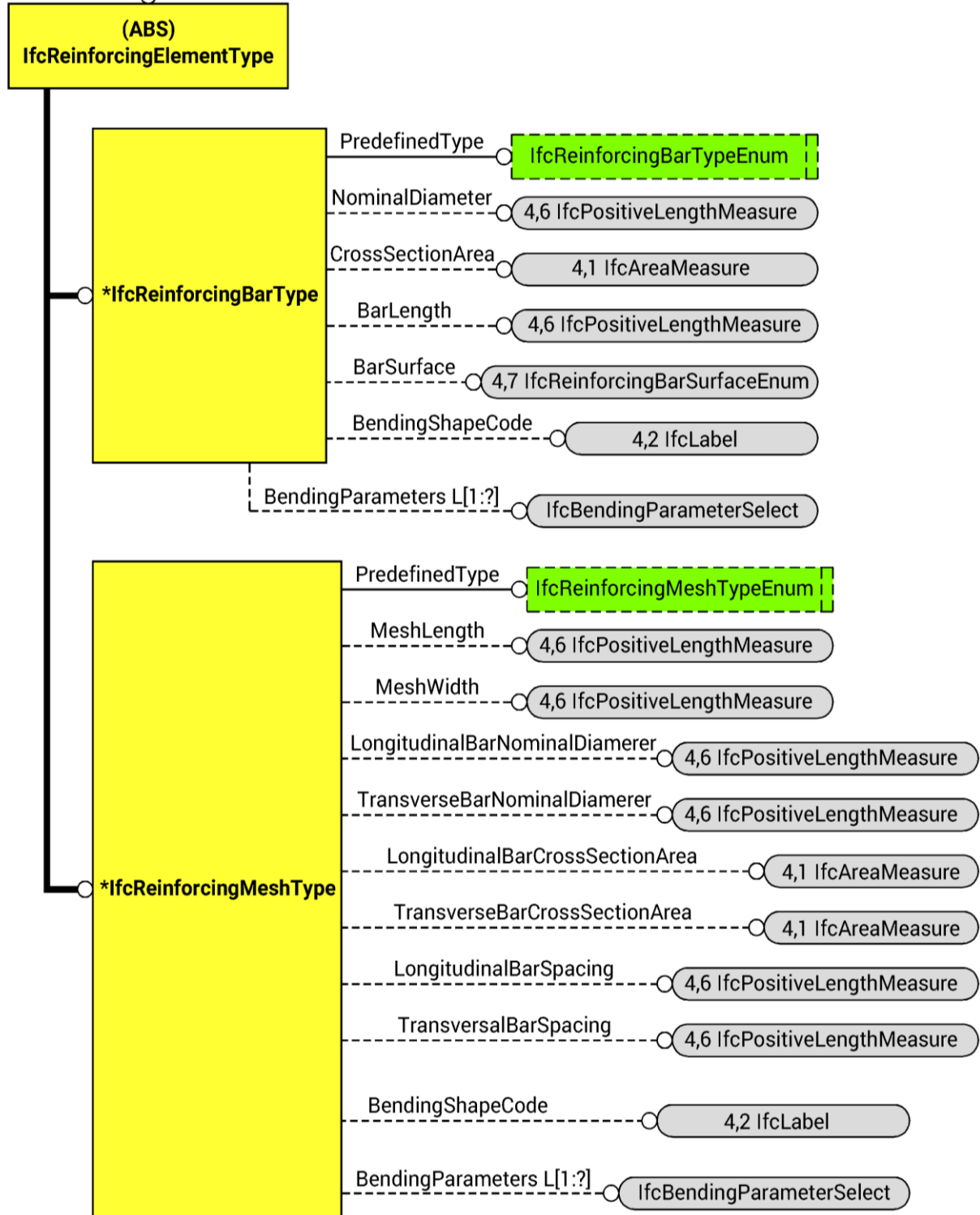


Figura 34 - IFC SED. Tipi di barre e reti d'armature.



## STRUCTURAL ELEMENTS DOMAIN<sup>36</sup>

Lo *Structural Elements Domain* fornisce la possibilità di rappresentare diversi elementi dell'organismo edilizio, o loro parti, che, in generale, hanno sì dimensione fisica, ma natura esclusivamente strutturale: descrive elementi di fondazione, armature lineari o a rete, cavi, capicorda e tiranti, attraverso le *entities* *IfcFooting*, *IfcPile*, *IfcReinforcingBar*, *IfcReinforcingMesh*, *IfcTendon* e *IfcTendonAnchor*.

Similmente allo *Shared Building Elements Domain*, è formato da *entities* che estendono il dominio *IfcSharedComponentElements* nelle sue *subentity* *IfcElementComponent* e *IfcElementComponentType* descrivendo, appunto, armature, cavi e loro dettagli costruttivi.

Poi due *entity* e le loro gemelle per descrivere il tipo, descrivono fondazioni profonde e superficiali. In questo caso, però, sono definite sottoposte di *IfcBuildingElement* e *IfcBuildingElementType*.

Infine, due *entity* descrivono particolari azioni effettuate su elementi strutturali, che ne modificano la geometria superficiale o volumica, come segni, trattamenti oppure fori, nicchie e canalette.

Sono oggetti molto compiutamente definiti e particolarmente utili in fase esecutiva per descrivere un oggetto così com'è realizzato, potendo, ad esempio, descrivere lunghezze delle barre d'armatura, superficie laterale, diametro nominale e sezione trasversale, codici particolari, lunghezze d'ancoraggio e particolari parametri di piegatura per il confezionamento.

Nelle immagini alle pagine precedenti sono riportate le *entities* che descrivono barre e reticolati d'acciaio per armature.

---

36 - buildingSMART, "IFC4 - Add.2; 7.8 IfcStructuralElementsDomain".

# 4. NAFEMS

Visto il tema trattato si è trovato molto più che necessario il confronto con uno standard anche a riguardo il metodo agli elementi finiti. Ciò si è concretizzato attraverso la National Association for Finite Element Methods & Standards, abbreviato in NAFEMS.

NAFEMS, ovvero *The International Association for the Engineering Analysis Community*, è un'associazione internazionale, nata nel 1983<sup>37</sup>, anch'essa senza scopo di lucro e senza un esplicito intento commerciale, che si pone come obiettivo il promuovere un impiego corretto e affidabile delle Tecnologie di Simulazione Numerica, fornendo un costante aggiornamento e qualificazione delle tecnologie informatiche disponibili. La dichiarazione d'intenti è "*To promote the safe and reliable use of finite element and related technology*" cioè: "promuovere l'utilizzo sicuro e affidabile degli elementi finiti e la tecnologia correlata"; difatti queste tecnologie e l'uso corretto di codici di calcolo sono elementi fondamentali per l'ingegneria moderna, si parla soprattutto di tecnologie FEM e FEA, ma si arriva anche a strumenti CAE e CFD, tutti strumenti essenziali per la ricerca, la simulazione, la prototipazione e l'ottimizzazione. Praticamente promuovono, principalmente, modelli e tecnologie agli elementi finiti che siano affidabili<sup>38,39</sup>.

Questa volontà si concretizza in una serie di servizi di informazione, diffusione, di conoscenze in addestramento e formazione professionale, finanziati attraverso la quota associativa versata all'iscrizione dai suoi associati.

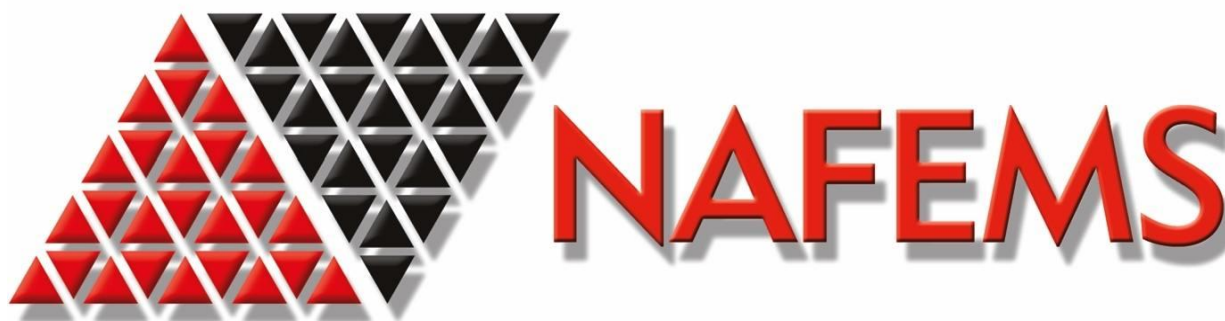


Figura 35 - NAFEMS. Logo.

---

37 - NAFEMS, "History";

38 - NAFEMS, "home page";

39 - NAFEMS, "A Finite Element Primer", pag. vii ss.

Le attività principali di NAFEMS, che opera anche all'interno delle iniziative normativo/didattiche dell'Unione Europea, riguardano l'informazione finalizzata all'utilizzo pratico dei sistemi software, la formazione e il controllo della qualità nonché la definizione di normative per la simulazione numerica. In pratica, per quanto riguarda l'informazione per l'utilizzazione dei programmi, l'associazione prospetta nuove tecnologie numeriche, quando professionalmente applicabili, attraverso ricerche, indagini, rassegne, documenti di revisione critica di tipologie di tecnologie software, casi studio, confronti funzionali fra codici di calcolo e, infine, direttive per la scelta di sistemi di calcolo. Per quanto riguarda la formazione, queste viene perseguita producendo materiale didattico, eserciziari, test base ed avanzati sulle diverse tecniche di calcolo, suggerimenti per le applicazioni, riferimenti bibliografici e simili, ma anche tenendo corsi, seminari, conferenze internazionali e pubblicando la rivista specializzata "*Benchmark*". Infine, per quanto riguarda il controllo della qualità e la definizione di normative per la simulazione numerica, l'associazione produce documenti che trattano l'estensione di norme ISO in materia, protocolli di calcolo, procedure di controllo, schemi per la certificazione del personale, specifiche per l'affidamento e la disciplina di prestazioni di calcolo richieste a terzi o mandate a gara, progetti quali il *SAFES*, cioè il *SAFE Structural Analysis*, che fissano il percorso logico della progettazione assistita all'elaboratore<sup>40</sup>.

In tutto ciò, NAFEMS si rivolge a progettisti e società di progettazione o di servizi di ingegneria, che operano nel campo della progettazione virtuale, soprattutto per quanto riguarda l'informazione e l'analisi comparativa e la qualificazione delle tecnologie software disponibili; i testi di consultazione rapida per l'applicazione corretta delle tecnologie nei diversi campi; la formazione del proprio personale attraverso corsi dedicati e finalizzati alla progettazione, eventualmente sostenuti anche con contributi finanziari di fonte istituzionale disponibili per progetti di formazione continua; la qualificazione del loro personale nel settore della sperimentazione virtuale; la ricerca di personale qualificato; le procedure di Controllo di Qualità per il calcolo e la fornitura di servizi di calcolo; l'informazione sulle iniziative di Ricerca e Sviluppo finanziate dalla UE; la diffusione di notizie sui servizi da loro offerti.

---

40 - NAFEMS, "about NAFEMS".



Negli ultimi 30 anni NAFEMS ha cercato di impegnarsi per semplificare a livello internazionale la collaborazione tra industria, università e governi, facendo leva su competenze ingegneristiche multidisciplinari e imparziali, per migliorare prodotti e simulazione dei processi e per avere un impatto positivo su qualità, profitto, tempistiche e sicurezza<sup>41</sup>.



Figura 36 - NAFEMS Italia. Logo.

---

41 - NAFEMS Italia, “home page”.

## IL RUOLO IN EVOLUZIONE DELLE ANALISI<sup>42</sup>

Nel passato, il dominio delle analisi era di stretto dominio di esperti e specialisti, spesso con interi dipartimenti di larghe compagnie dedicati allo scopo; oggi, potenti macchine computazionali a basso costo e strumenti per analisi ad ampio raggio, sono facilmente reperibili per gli ingegneri. Strumenti come il CAD, *Computer-Aided Design*, la trasmissione della geometria, il *meshing* automatico, il controllo d'errori interno allo stesso programma nonché il perfezionamento e l'ottimizzazione adattivi degli strumenti hanno portato come conseguenza che i prodotti sono anche molto più facili da utilizzare. Questo ha portato a far sì che quantità significative di analisi sono svolte dagli ingegneri attraverso tutto il ciclo di design del prodotto, e che compagnie più piccole hanno rapidamente appreso le tecnologie di analisi; allo stesso tempo le compagnie desiderano ridurre prove e test, in gran parte, sui risultati di analisi ingegneristiche, concepite come esibizione di convalida. Tutto ciò dà una grande responsabilità a ingegneri e analisti.

È ampiamente riconosciuto che, per mantenere un personale qualificato e assolutamente competitivo in un mercato in rapida evoluzione, continui investimenti in formazione debbano essere stanziati. Molte aziende hanno autorizzato i loro impiegati a ottenere idonee conoscenze per il loro posto di lavoro; ciò è molto positivo, perché porta a continui sviluppo e accrescimento di conoscenze. NAFEMS in tutto ciò aiuta poiché uno dei suoi obiettivi centrali di NAFEMS, monitorato internamente, è di esaminare i bisogni di insegnamento ed esercizio per tutti i soggetti coinvolti nelle analisi e simulazioni ingegneristiche, provvedendo a informazioni per soddisfare ai loro bisogni. In pratica questo può essere diviso in due specifici obiettivi: identificare, appunto, ciò che necessario ai soggetti meno esperti per metterli in grado di produrre affidabili e accurati risultati numerici, e di metterli nelle condizioni di poter concretamente fare questo. Il secondo step è invece rivolto ad analisti più esperti, per far sì che possano estendere le loro aree di competenze e quindi condividere le loro conoscenze con una comunità più ampia. A questo provvede NAFEMS, con informazioni, materiali, eventi e con reti di contatti.

---

42 - NAFEMS Italia, "Professional Development – The Changing Role of Analysis".

Tra fine anni 70 e inizio anni 80, mentre macchine potenti diventavano più ampiamente disponibili, l'industria iniziava sempre più a risolvere problemi ingegneristici sfruttando tecniche di analisi agli elementi finiti; c'era comunque una notevole preoccupazione riguardo la precisione di questi metodi e le loro implementazioni software, precisione d'altro canto che deve necessariamente essere verificata per poter permettere l'utilizzazione di questi risultati. A seguito di una grossa attività lobbistica da parte di industrie e università, il dipartimento del commercio e dell'industria del governo del Regno Unito mise in piedi un progetto all'interno del Laboratorio Nazionale d'Ingegneria, con base nell'East Kilbride, in Scozia, per poter assolvere questo compito. In breve tempo nacque la *National Agency for Finite Element Methods and Standards*, abbreviato in NAFEMS, nel 1983, con lo specifico obiettivo di stabilire un sicuro e affidabile utilizzo degli elementi finiti e delle tecnologie collegate.

Ai tempi in cui questo obiettivo veniva stilato, la comunità di ingegneri era preoccupata principalmente dell'accuratezza di codici di analisi delle tensioni, basate per lo più su metodi agli elementi finiti. L'impegno iniziale si concentrò sullo sviluppo dello standard "*Benchmarks*", in rapporto al quale i codici potevano essere testati. NAFEMS pubblicò i risultati di questi "*Benchmarks*" per vari codici e l'industria dei software rispose in breve tempo adottando questi testi come metodo per migliorare e verificare l'accuratezza dei codici. Oggigiorno la maggior parte dei produttori utilizza questi standard come parte del loro continuativo processo di controllo qualità. È per questo motivo che per cui si è scelta NAFEMS come punto di riferimento per un'informazione di qualità riguardo le analisi agli elementi finiti.

Parallelamente a queste attività, fu anche riconosciuto anche il bisogno di formazione di un certo livello; NAFEMS iniziò a commissionare alcuni libri di testi e dettagliati resoconti tecnici per aree disciplinari per cui le informazioni erano percepite come essenziali o, semplicemente, lacunose. Questi testi sono ancora prodotti per NAFEMS da alcuni esperti nel campo; una delle più importanti caratteristiche di questi testi, che velocemente si sono guadagnati una buona reputazione, era che ognuno subiva rigorose esamine e critiche da NAFEMS stessa, che commissionava il lavoro; questi gruppi di lavoro formarono un ben amalgamato gruppo di eccellenti ricercatori accademici, ingegneri professionisti e produttori di software, dando un punto di vista e una prospettiva unica al problema.

Ogni nuovo testo viene, sin dagli inizi, reso disponibile gratuitamente ai membri dell'associazione, ciò ha fatto sì che si formasse una larga libreria di materiale di riferimento, che continua a essere disponibile anche per i nuovi membri.

Poi, vennero fin da subito, tenuti seminari di consapevolezza, che continuano ancora oggi, come mezzi per creare una rete e di condivisione effettiva di informazioni. Una delle caratteristiche chiave era l'ampio spaccato di industrie rappresentate, fornendo quindi eccezionali opportunità di trasferimento di tecnologie. Si stabilì che queste conferenze avessero cadenza biennale e che potessero raggiungere una scala globale; ancora oggi prendono luogo e sono un punto di riferimento per chi è del mestiere.



Figura 37 - NAFEMS. Locandina NWC 2017.

Infine, per mantenere gli ingegneri aggiornati sugli sviluppi sul mondo delle analisi, venne fondato, nel luglio del 1987 e con cadenza trimestrale, il periodico “*Benchmark*”<sup>43</sup>.

---

43 - NAFEMS, “Benchmark Magazine”.

## CONSIDERAZIONI

In breve l'associazione si propone come riferimento per le analisi agli elementi finiti: il confronto con il materiale prodotto a riguardo è servito a capire compiutamente ciò che era necessario implementare all'interno dello standard propugnato da buildingSMART. È stato fondamentale il libro "A Finite Element Primer" pubblicato nel 1986 che, come da titolo, si vuole porre come punto di partenza per la conoscenza della materia, cercando di mantenere una prospettiva molto ampia.

Infatti per la descrizione del vasto mondo dei modelli e delle analisi agli elementi finiti, il libro comincia spiegando le origini della teoria numerica e come si è affermata e sviluppata con il passare degli anni. Affronta poi, in maniera sufficientemente generale, ma mai generica o banale, il funzionamento di questa teoria, gli elementi che la costituiscono e la governano, quindi sia il funzionamento matematico generale, sia l'applicazione in base alla dimensionalità degli elementi utilizzati per la discretizzazione. Di questi vari elementi vengono ampiamente specificate le caratteristiche di funzionamento, di schematizzazione e di risoluzione, comprese in che situazioni si preferiscono quali. Infine, è presente anche un capitolo che tratta dei risultati, per lo più come questi debbano essere, testuali parole, "assemblati", ossia combinati tra di loro<sup>44</sup>.

Un capitolo poi si è rivelato essere, fortuitamente, dedicato alle analisi dinamiche, volendo esporre, dopo una cospicua introduzione che ne spiegava la teoria fisica, anche come queste debbano essere impostate, condotte e quali siano i risultati importanti quando siano eseguite su di un modello agli elementi finiti.

È stato importante anche il confronto tra questo testo e specifici libri didattici sugli argomenti, cioè elementi finiti e dinamica, per poter abbracciare più ampiamente possibile la disciplina<sup>45</sup>.

---

44 - NAFEMS, "A Finite Element Primer";

45 - Enzo Tonti, "Introduzione Elementare al Metodo agli Elementi Finiti".

Infine, molto utile si è trovato essere anche un breve confronto con il manuale d'uso del software TNO Diana, che ha contribuito alla formazione di un punto di vista che considerasse anche una specifica applicazione pratica di queste teorie<sup>46</sup>.

---

46 - TNO Diana, "User's Manual".



# **5. necessità**



Ci si vuole porre come obiettivo poter descrivere attraverso IFC il mondo delle analisi strutturali nella maniera più completa possibile. Ciò significa poter mantenere informazioni di ogni tipo di analisi e verifica statica e ogni tipo di analisi e verifica dinamica, siano esse effettuate in un qualsiasi momento storico del ciclo di vita dell'edificio considerato e, almeno inizialmente, con strumenti software, agli elementi finiti. Difatti rimane necessario poter interfacciare attraverso IFC anche solutori diversi, come quelli a telaio equivalenti e programmi che, ad esempio, trattano specificatamente di muratura portante.

Come si è visto ciò che riguarda la parte di analisi statiche di un edificio da realizzare è praticamente completa, a meno di pochi, facilmente implementabili, dettagli secondari che esulano dall'obiettivo di questa tesi, come il poter descrivere carichi di precompressione nel calcestruzzo armato. Ciò che rimane tuttora escluso dal mondo IFC e che la stessa buildingSMART vuole prioritariamente aggiungere è tutto ciò che riguarda le analisi dinamiche di un qualsiasi sistema strutturale e l'assoluta impossibilità di descrivere un modello agli elementi finiti<sup>47</sup>. Il passo successivo di estensione si auspica essere ciò che riguarda analisi e verifiche per un edificio esistente, anche e soprattutto per sistemi costruttivi in muratura, ancora assenti nelle *Industry Foundation Classes*, sia che si tratti di muratura storica che di muratura moderna.

Volendo implementare nelle *Industry Foundation Classes* le analisi dinamiche e i modelli agli elementi finiti, si vuole, a questo punto, vagliare le informazioni che risulta effettivamente necessario poter consultare e inserire all'interno di un qualsiasi modello, in questo caso BIM, per poter condurre analisi e verifiche dinamiche delle strutture e si vorranno esplicitare, poi, tutti i dati necessari per poter descrivere un modello agli elementi finiti, considerando che già si dispone di un modello analitico.

A seguire verranno fatte alcune considerazioni perché sia possibile inserire qualsiasi sistema strutturale ordinario. Questo si traduce nel poter trattare valutazioni di sicurezza e di vulnerabilità di un edificio già realizzato e, ancora di più, sul poter considerare un qualsiasi sistema strutturale in muratura portante.

---

47 - buildingSMART, "IFC4 - Add.2, 7.7 IfcStructuralAnalysisDomain".

**5A. in analisi agli e. f.**

## MODELLO AGLI ELEMENTI FINITI

Passando a considerazioni a riguardo del mondo delle analisi agli elementi finiti la situazione si fa più variegata, per l'enorme variabilità dei singoli solutori in commercio. Infatti, se una sola è la teoria degli elementi finiti, possono essere più numerose le singole modalità di risoluzione pratiche di volta in volta applicate.

Di per sé, l'idea che sta alla base è relativamente semplice e si può vedere come due parti separate: la prima è la divisione di una struttura in ipotetici elementi più piccoli, gli elementi finiti, che sono abbastanza piccoli da poter approssimare correttamente gli sforzi e le deformazioni. Questa è la discretizzazione del problema e a questo punto rimane solamente la soluzione numerica di un qualcosa che può risultare in diversi tipi di funzioni. La seconda parte è l'assemblaggio di questi elementi, in maniera tale che sforzi e deformazioni siano continui tra i vari elementi finiti e, di più, in equilibrio con i carichi esterni e le condizioni di vincolo. È la seconda parte di cui, in questa tesi, ci si interessa e coinvolge i codici di calcolo proprietari, per cui è necessario una prospettiva sufficientemente generale, ma non per questo imprecisa o incompleta e, se possibile, aperta a sviluppi futuri.

Sappiamo poi che il metodo agli elementi finiti è governato da tre grandi argomenti, cioè l'equilibrio delle forze e degli sforzi, sia statici che dinamici, sia in ambito lineare che non, poi la compatibilità degli spostamenti e delle deformazioni, infine la legge sforzo-deformazione in campo elastico o elastoplastico, per i metalli, con plasticizzazione. Se ne sottolinea l'importanza sapendo che alcuni risolutori possono basarsi sul principio dei lavori virtuali, per modelli semplici e monodimensionali soprattutto, per cui si possa calcolare la soluzione esatta<sup>48</sup>.

Si rivela essere necessario poter descrivere elementi monodimensionali, di cui, attraverso la legge sforzo-deformazione, è possibile ottenere una soluzione esatta attraverso lo stesso principio dei lavori virtuali, ciononostante la maggioranza dei solutori utilizza sempre la soluzione approssimata agli elementi finiti, per semplicità di calcolo.

---

48 - NAFEMS, "A Finite Element Primer", pag. 14 ss; pag. 29 ss.

Si rivelano essere necessari per la descrizione di un elemento monodimensionale, rettilineo o curvo:

-I vari tipi elementi finiti;

-I vari nodi di ciascuno e come sono condivisi, siano essi due o tre nodi ciascuno;

-I punti per l'integrazione secondo la regola di Simpson, utili per la risoluzione;

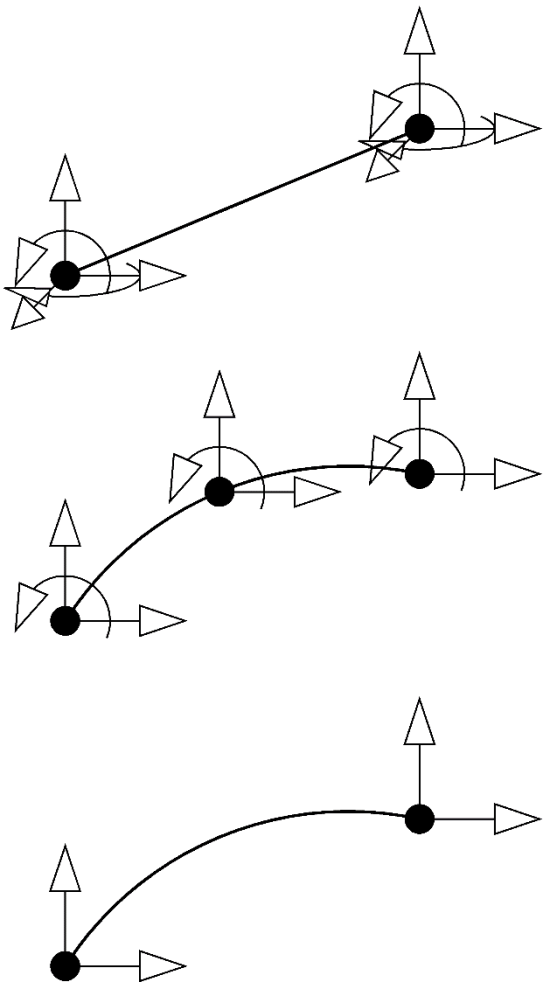


Figura 39 - Elementi finiti monodimensionali. Gradi di libertà.

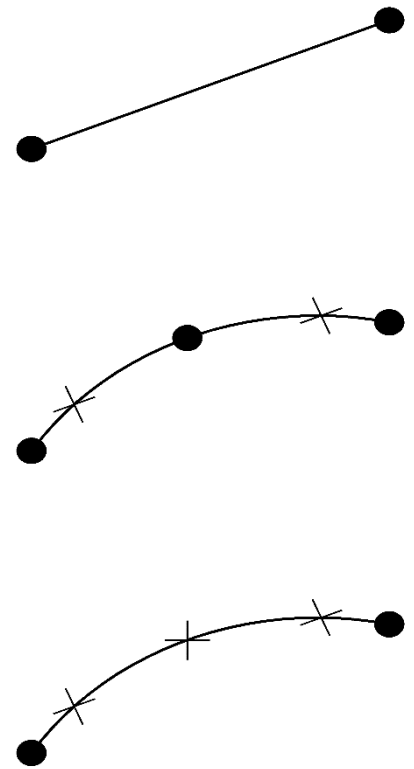


Figura 38 - Elementi finiti monodimensionali. Nodi e punti di integrazione.

-I gradi di libertà di ciascun nodo, per cui ritorna utile la definizione delle funzioni di forma di cui a punto seguente;

-Le funzioni di forma dei nodi, che hanno la proprietà di descrivere spostamento e rotazione dei nodi e di restituire un valore nullo in nodi diversi da quello cui si riferiscono, mentre restituiscono valore unitario quando riferite al nodo stesso, poiché sono funzioni della lunghezza dell'elemento. Possono essere funzioni polinomiali, ma, sempre nell'ottica di considerare l'enorme ampiezza di queste e, soprattutto, ogni possibile futuro le si considererà semplici e indeterminate funzioni. Qualora un grado di libertà di un elemento non sia attivato si

considererà la corrispondente deformazione come nulla a priori.<sup>49</sup>

Per quanto riguarda gli elementi bidimensionali, sia triangolari che quadrangolari, sia dai lati rettilinei che curvi, la situazione è simile poiché dovremo descrivere:

-I vari elementi finiti;

-I vari nodi che questi possiedono e come li condividono, sapendo che possono essere 3, 4, 5, 6, 7, 8, 9, 12, 15 e 16 anche al variare della geometria;

-I punti di integrazione che ne gestiscono la risoluzione, questa volta che possono seguire anche il Teorema di Gauss, che dovranno essere collegati anche al cambio di coordinate, specie per elementi

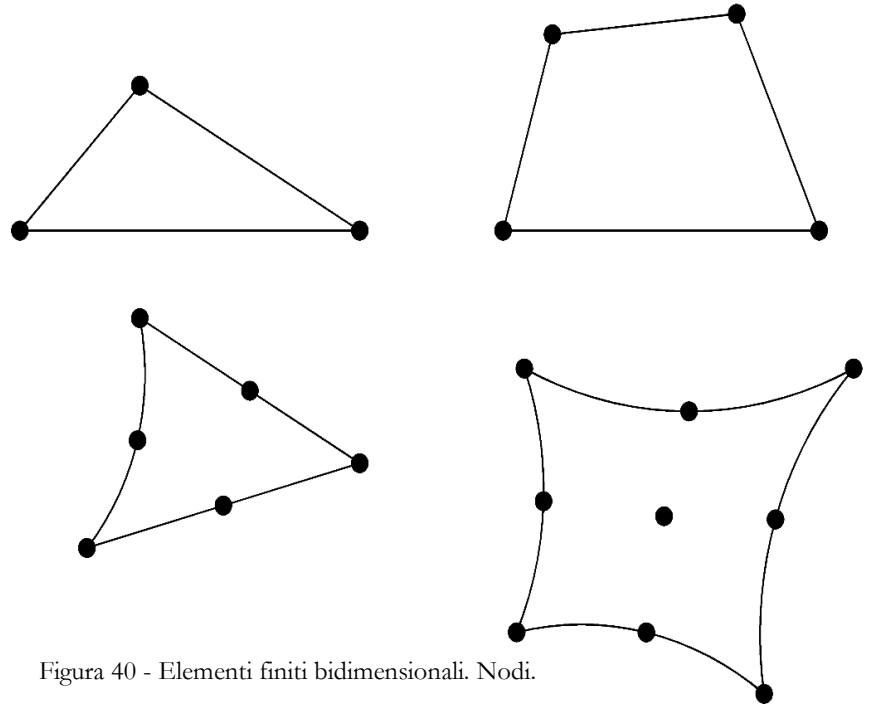


Figura 40 - Elementi finiti bidimensionali. Nodi.

isoparametrici. Difatti per completezza dell'informazione può essere necessario indicare quali punti di integrazione sono stati utilizzati per l'integrazione numerica che ha portato alla soluzione. Quando non è possibile integrare analiticamente, e quindi esattamente, una funzione di forma, si ricorre al metodo di integrazione dei punti di Gauss;

-I gradi di libertà di ciascun nodo;

-Le funzioni di forma dei nodi, con le considerazioni fatte anche per gli elementi monodimensionali;

-Inoltre sarà necessario riportare il cambio di coordinate che comunemente, per semplicità, viene adottato per descrivere ogni elemento e le sue funzioni di forma in coordinate adimensionali centrate

---

49 - NAFEMS, "A Finite Element Primer", pag. 29 ss.

nell'origine, potendo anche, in caso di necessità, rettificare i bordi curvilinei di un elemento isoparametrico di forma e qualsiasi.<sup>50,51</sup>

Infine, nonostante elementi mono e bidimensionali descrivano da sole più dell'80% delle strutture analizzate oggi, si rende necessario descrivere ulteriori elementi finiti, elementi tridimensionali, senza per ciò riformulare ciò che è postulato finora, nonostante la complessità decisamente maggiore

del metodo e il calcolo decisamente più oneroso. Difatti, si cerca di ricondursi il più possibile a problemi mono e bidimensionali, in generale. Rimane normalmente necessario poter descrivere:

-I vari elementi finiti, siano essi delle ormai più svariate geometrie:

-I vari nodi che questi possiedono e come li condividono, sapendo che possono essere 4, 10,

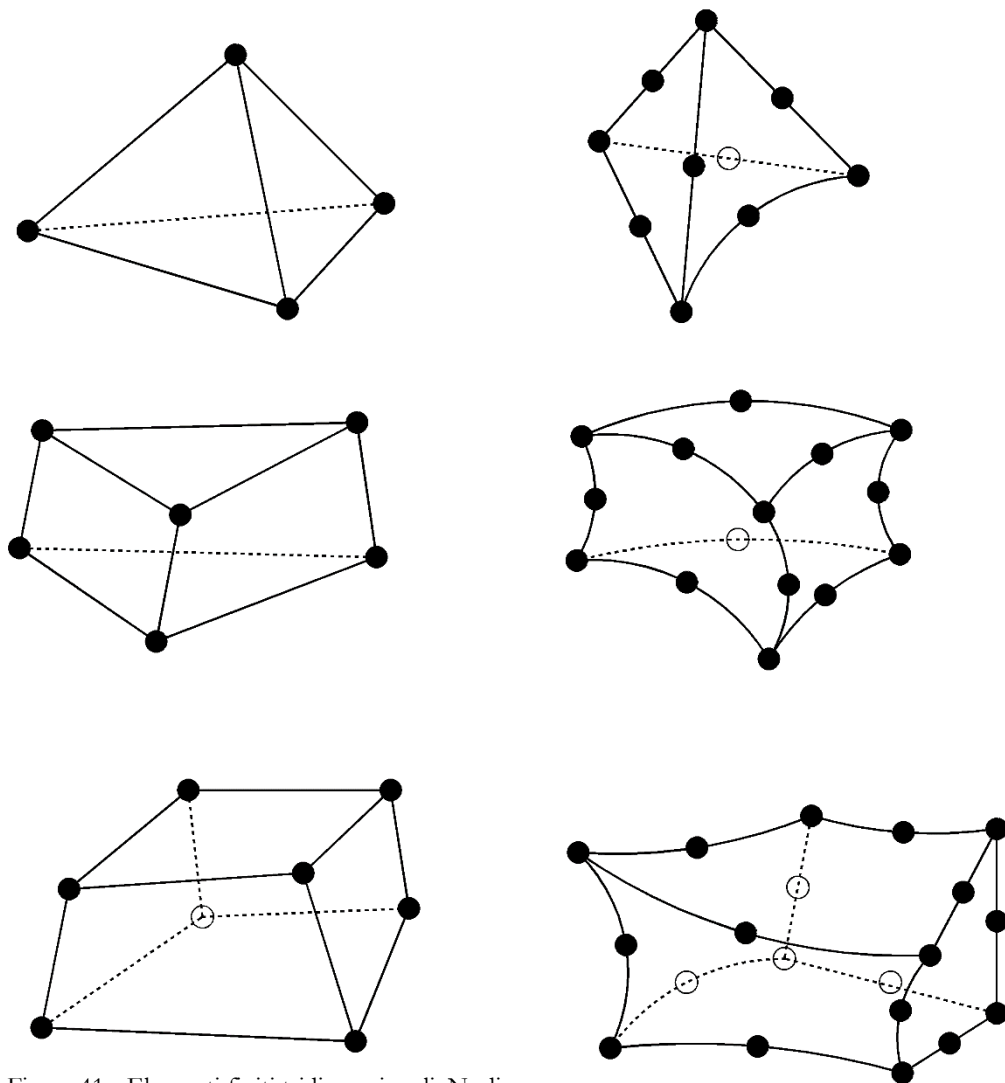


Figura 41 - Elementi finiti tridimensionali. Nodi.

50 - NAFEMS, "A Finite Element Primer", pag. 43 ss;

51 - Enzo Tonti, "Introduzione Elementare al Metodo agli Elementi Finiti", pag. 63 ss.

14, 16, 25 se riconducibili, anche con cambio di coordinate, a tetraedri, oppure 5, 13, 18, 21, 37 nodi, se riconducibili a elementi piramidali a base quadrangolare. O 6, 15, 18, 24 o 42 nodi se riconducibili a prismi a base triangolare, oppure 8, 20, 26, 32 o 48 se riconducibili a parallelepipedi. A seconda che i nodi siano nei soli vertici, oppure anche uno o due lungo gli spigoli, oppure uno o più anche al centro della faccia;

-I punti di integrazione che ne gestiscono la risoluzione, di nuovo secondo il Teorema di Gauss, che ancora dovranno essere collegati anche al cambio di coordinate, specie per elementi isoparametrici;

-I gradi di libertà di ciascun nodo;

-Le funzioni di forma dei nodi, con le considerazioni fatte anche per gli elementi di altre dimensionalità;

-Di nuovo sarà necessario anche riportare il cambio di coordinate che comunemente, per semplicità, viene adottato descrivere ogni elemento e le sue funzioni di forma in coordinate adimensionali centrate nell'origine, potendo anche, in caso di necessità, rettificare i bordi curvilinei di un elemento isoparametrico di forma qualsiasi.<sup>52</sup>

Una volta descritto tutto ciò, per completare il modello agli elementi finiti, è necessario implementare altri quattro blocchi di informazioni: la geometria, ovvero l'alternarsi dei elementi, le condizioni di carico e di vincolo e, infine, i risultati della analisi condotte.

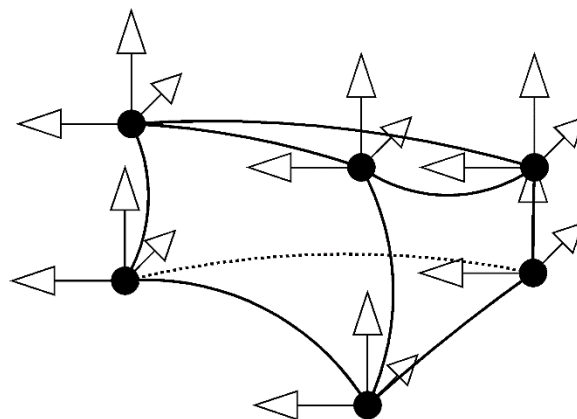
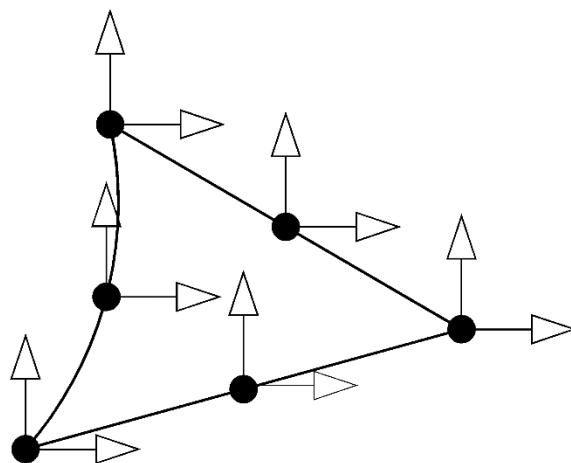


Figura 42 - Elementi finiti tridimensionali. Gradi di Libertà.

52 - NAFEMS, "A Finite Element Primer", pag. 68 ss.

## CONSIDERAZIONI PRELIMINARI

Alla luce di tutto ciò si è iniziato a pensare come implementare tutto ciò all'interno dello standard IFC. Riportare i vari elementi in sé è necessario, sicuramente, come *entities* che ereditino la definizione geometrica, dimodoché l'identificazione topologica dell'elemento sia lasciata a *entities* funzionali in questo. Difatti la maggior preoccupazione è la descrizione di complessi elementi isoparametrici: si vuole essere sicuri che qualsiasi elemento sia ascrivibile alle *Industry Foundation Classes* e, così facendo, se ne può essere sicuri. Infatti in IFC si può agevolmente descrivere una qualunque geometria e, appoggiandosi a questa, si possono descrivere elementi finiti anche molto complessi. In questo modo, poi, a una geometria ben definita, si possono aggiungere le proprietà tipiche degli elementi finiti, ad esempio le proprietà meccaniche che sono conseguenza della caratterizzazione dei materiali. Un *attribute* verso una *class* che riporti un materiale con le sue caratteristiche meccaniche dovrebbe assolvere brillantemente allo scopo.

È poi imprescindibile descrivere i nodi appartenenti all'elemento finito. Si dovrà descrivere tutti i nodi presenti, sia quelli obbligatoriamente presenti nei vertici, sia quelli che possono essere presenti lungo gli spigoli o al centro delle facce. Sarà necessario riportare le interconnessioni tra gli elementi finiti, cioè se un nodo appartenga a diversi elementi. Infine, ma sicuramente altrettanto importante, sarà necessario riportare la funzione di forma del nodo qualunque sia la loro espressione matematica. Nel caso in cui si riesca a distinguere i nodi in base alla dimensionalità dell'elemento finito cui appartengono, si possono fare

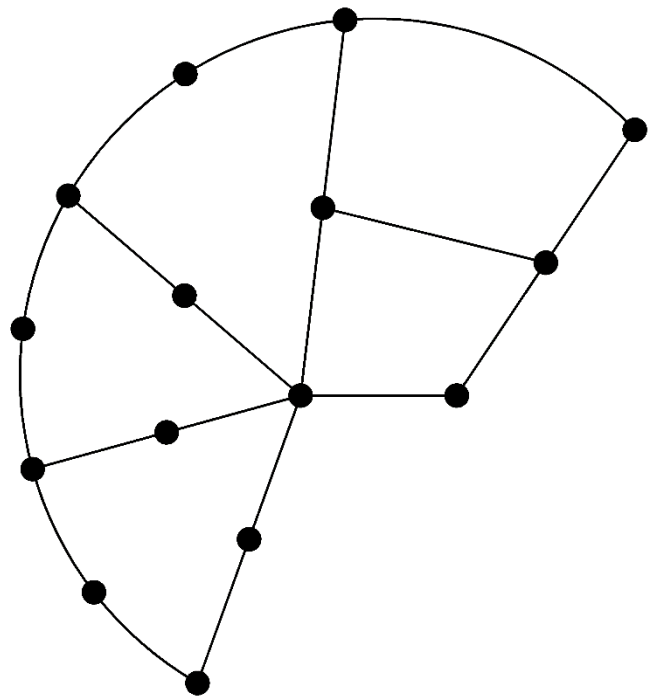


Figura 43 - Elementi finiti. Congruenza degli elementi.

delle considerazioni sapendo che, al netto di nuove e valide funzioni di forma, la maggior parte di esse sono conosciute. Tuttavia la stessa NAFEMS non distingue i singoli tipi di funzioni di forma, perché



non dipendono dal metodo in sé, ma sono collegate alle necessità di calcolo e, soprattutto, possono evolversi e cambiare grazie alla continua ricerca in materia. In conclusione, sembra utile poter implementare una *class* in cui inserire queste funzioni, a prescindere dalla loro precipua formulazione matematica, che può essere mutevole. Di conseguenza sembra essere non necessario distinguere i nodi in base alla dimensionalità dell'elemento cui appartengono.

È all'interno della trattazione dei nodi che deve trovare spazio la descrizione di elementi speciali, come molle, *truss*, smorzatori e quant'altro. Si deve esplicitare contemporaneamente nel nodo sia la funzione di forma nella direzione desiderata sia l'impossibilità di deformazione del nodo in un'altra direzione. Ovvero dev'essere tutto collegato anche con i gradi di libertà del nodo stesso e con modalità di deformazione particolari.

Sembra poi utile descrivere gli eventuali punti di integrazione. Di nuovo, nel distinguere sulla dimensionalità dell'elemento finito si potrebbe distinguere tra punti di integrazione necessari o meno e sulla regola che li governa. Un'idea iniziale è di mantenere opzionale l'*attribute* che li collega all'elemento finito e con un *enumerated type* riportare la regola integrativa. Così facendo la modellazione rimane flessibile e funzionale.

Infine è necessario riportare i risultati. Prima di tutto è necessario parlare di forze e spostamenti dei nodi. Poi di sforzi e deformazioni, di risultati, anche grafici, e di cambi di coordinate, eventuali, per gli elementi finiti stessi e il risultato dei punti di integrazione.

Risulta intuitivo che sforzi e deformazioni dei nodi di un modello agli elementi finiti siano concettualmente assimilabili a sforzi e deformazioni di un modello analitico, quindi per informatizzare questi potrebbe essere semplicemente necessario una specifica *entity* che viene "riempita" di significato da *attributes* che ricollegano, appunto a sforzi e deformazioni puntuali. Questa strada sembra preferibile anche per il leggero impatto che ha sullo standard IFC. Altrimenti sarebbe necessario creare nuove *entities* con i loro *attributes* per descrivere a pieno questi risultati.

A seguire, è utile riportare anche la geometria deformata degli elementi finiti in sé, in modo che anche graficamente siano restituibili i risultati delle analisi condotte. Ma ancora più importante sarà riportare i cambi di coordinate effettuati, con i parametri utilizzati nelle direzioni considerate. Come i punti di integrazione, questa è un'informazione di per sé trascurabile: è utile invece riportarla perché racconta il processo con cui si sono ottenuti i risultati, non solo i dati di input e di output. Se i dati in output, ad esempio, sono diversi tra diverse analisi svolte, con diversi software, i valori dei parametri e dei

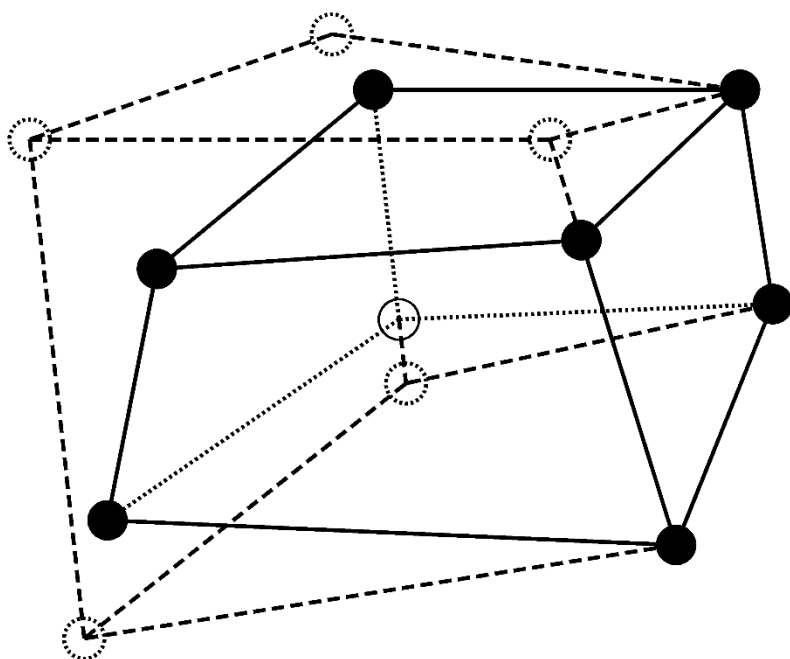


Figura 44 - Elementi finiti tridimensionali. Deformazioni.

all'interno dell'elemento e i risultati che gli sono propri: una semplice *entity* dovrebbe essere sufficiente allo scopo.

punti di integrazione ci aiutano a capire se sia un problema dato dal software o un errore umano del progettista. Per lo più si tratterà di difficoltà nel processo esportativo nel formato “.ifc”. In pratica, tornando ai ragionamenti, sarà utile alla nostra causa riportare all'*entity* dei risultati dell'elemento finito uno o più *attributes* che riportano i valori dei parametri utilizzati per cambiare variabile.

Infine sarà, come già anticipato, utile riportare i risultati dei punti di integrazione, con la loro posizione



# **5B. in dinamica delle strutture**

Il primo passo è stato, quindi, quello di confrontarsi con la conoscenza scientifica di ciò che riguarda le sollecitazioni dinamiche di una struttura, sempre con la consapevolezza che deve essere possibile ampliare il tutto, in un secondo momento, a comprendere i comportamenti degli edifici esistenti e in muratura. Alla conoscenza prettamente scientifica si sono, quindi, aggiunte le richieste e i modelli proposti dalle normative, in modo da ottenere un punto di vista complessivo del tutto, comprese situazioni meno rigorose. Si è rilevato molto utile, anche in questo frangente, lo stesso standard sviluppato dalla National Agency for Finite Elements Methods & Standards che affronta l'argomento in maniera sufficientemente generale, con l'intento poi di applicarlo ai modelli agli elementi finiti.

## INFORMAZIONI DEL SISTEMA

Dall'analisi di letteratura è emersa la necessità di poter accedere alle seguenti proprietà di un sistema strutturale:

Pulsazione del sistema  $\omega = \sqrt{k/m}$ ;

Pulsazione del sistema smorzato  $\omega_d = \omega\sqrt{1 - \xi^2}$ ;

Frequenza circolare o periodo del sistema non smorzato  $T = 2\pi/\omega$ ;

Frequenza circolare o periodo del sistema smorzato  $T_d = 2\pi/\omega_d$ ;

Coefficiente di smorzamento viscoso  $c$ ;

Fattore di smorzamento  $\xi = c/c_{cr} = c/2m\omega \cong \frac{\ln(x_n/x_{n+1})}{2\pi}$  dove l'ultimo passaggio vale per valori piccoli del fattore di smorzamento stesso;

Coefficiente di smorzamento critico  $c_{cr} = 2m\omega = 2\sqrt{km}$ ;

Fattore di decadimento  $\alpha = \xi\omega$ .<sup>53,54,55</sup>

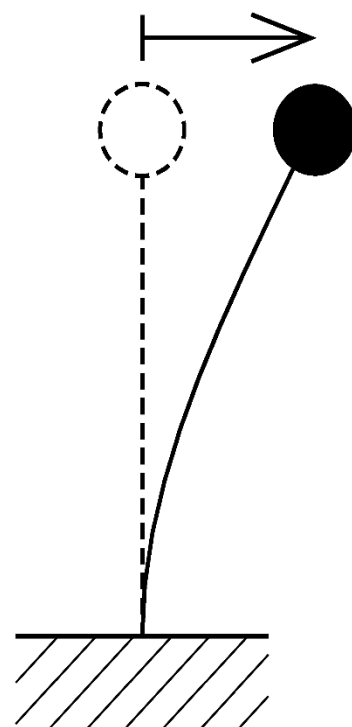


Figura 45 - Dinamica delle strutture. Elemento a un grado di libertà.

53 - NAFEMS, "A Finite Element Primer", pag. 170 ss;

54 - Francesca Da Porto, "Dinamica dei Sistemi Lineari ad 1 GdL";

55 - Francesca Da Porto, "Dinamica dei Sistemi Lineari ad n GdL".

## FORZANTI ESTERNE

Alle precedenti, che sono informazioni proprie in ingresso della struttura, si aggiungono informazioni che riguardano le azioni. Sarà infatti necessario poter definire la forzante esterna per descrivere la risposta del sistema al di là delle oscillazioni libere; sia che sia sollecitato armonicamente, sia impulsivamente o quant'altro, per poter completare i dati in ingresso.

Si distinguono, sostanzialmente, quattro tipi di forzanti:

1. Una forzante periodica, con variazioni che si ripetono nel tempo, tra queste rientrano le forzanti sinusoidali;
2. Una forzante di intensità per lo più deterministica, di carattere transitorio, tipicamente impulsiva;
3. Una forzante di intensità casuale, di carattere stazionario, cioè ripetitiva e con lo stesso andamento temporale al ripetersi di svariati cicli;
4. Una forzante di intensità casuale, di carattere transitorio.

A seconda del tipo di forzante che insiste sul sistema sarà necessario poter definire un insieme di valori utili a descriverla compiutamente:

- Il periodo della forzante;
- L'intensità massima della forzante;

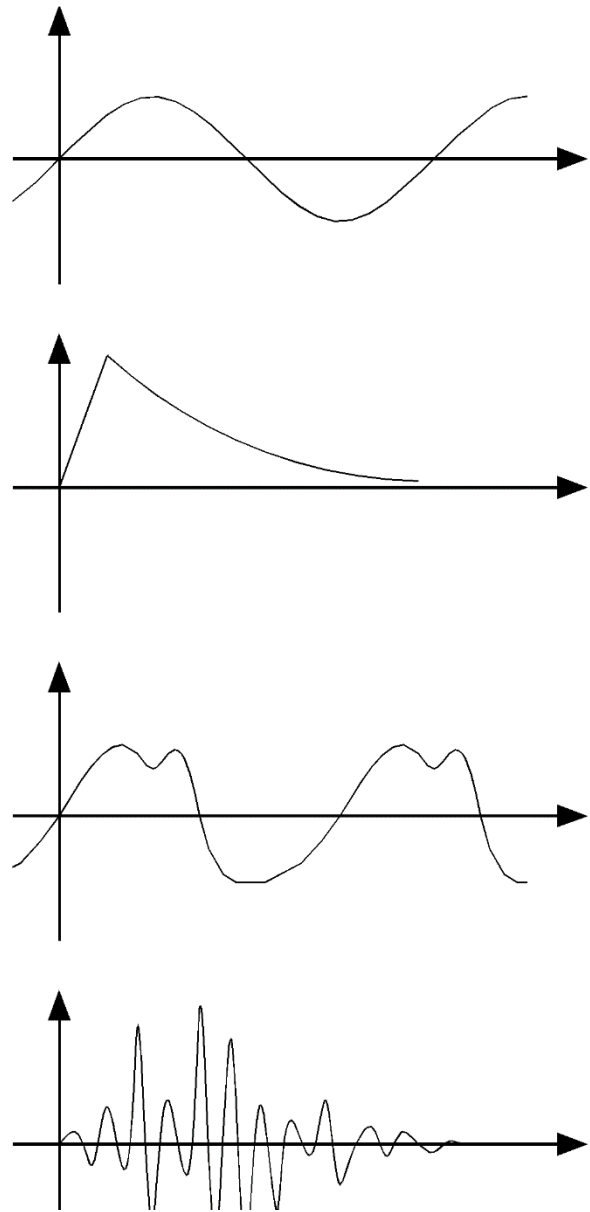


Figura 46 - Dinamica delle strutture. Tipologia di forzanti esterne.

- Il conseguente rapporto tra frequenza del sistema e frequenza della forzante;
- Coefficienti di amplificazione dinamica per struttura smorzata e non smorzata, nonché risultati statici equivalenti.
- Accelerogrammi;
- Spettri in spostamento;
- Spettri in velocità;
- Spettri in accelerazione<sup>56,57,58</sup>.

L'utilità di forzanti armoniche e impulsive rimane in ambiti diversi da quelli sismici, si pensi a simulazioni con particolari impianti o situazioni ambientali, o a particolari Stati Limiti per determinate tipologie di strutture, per cui non verranno considerate, ma sarà descritto compiutamente il problema sismico e vista la ancora limitata capacità importativa dei software.

Vista complessità data dalla dinamica delle strutture, soprattutto per quanto riguarda, ad esempio, la non-linearità del comportamento in alcuni tipi di analisi, si è preferito poter descrivere compiutamente azioni dell'ambito dinamico, come si spiegherà meglio a tempo debito, e risultati di azioni dinamiche vere e proprie. Rimarranno escluse e da implementare in un secondo momento molte altre informazioni, come la non linearità del comportamento in alcuni tipi di analisi, in cui bisogna considerare legami costitutivi non lineari, problemi di instabilità e di grandi spostamenti.

---

56 - Francesca Da Porto, "Dinamica dei Sistemi Lineari ad 1 GdL";

57 - Francesca Da Porto, "Dinamica dei Sistemi Lineari ad n GdL";

58 - NAFEMS, "A Finite Element Primer", pag. 170 ss.

## RISULTATI

Infine, dopo i dati in ingresso, deve essere possibile informatizzare nello stesso file i risultati delle analisi dinamiche, a prescindere dal fatto che esse siano lineari o non lineari e dal metodo di calcolo con cui esse siano state condotte. Più in particolare deve essere possibile inserire, almeno:

- Spostamenti iniziali e massimi;
- Velocità iniziali e massime;
- Accelerazioni iniziali e massime almeno;
- Sforzi e tensioni interne statici equivalenti, corrispondenti a queste situazioni;

Di conseguenza significa poter inserire, prioritariamente, come risultati del particolare sistema strutturale e delle azioni ad esso collegate:

- Ampiezza delle oscillazioni;
- Fase iniziale delle oscillazioni;
- Il numero di sistemi di libertà del sistema;
- Autovalori;
- Autovettori;
- Costanti moltiplicative per la normalizzazione, ingegneristica o matematica, degli autovettori;
- Modi di vibrare;
- Masse partecipanti;
- Fattori di smorzamento viscoso modali e/o coefficienti di Rayleigh<sup>59,60,61</sup>.

---

59 - Francesca Da Porto, “Dinamica dei Sistemi Lineari ad 1 GdL”;

60 - Francesca Da Porto, “Dinamica dei Sistemi Lineari ad n GdL”;

61 - NAFEMS, “A Finite Element Primer”, pag. 170 ss.



## CONSIDERAZIONI PRELIMINARI

Ritornando al singolo problema sismico, si inizia a considerare come inserire le informazioni di cui si è rilevata la necessità, conformandole all'architettura di IFC.

Per le caratteristiche intrinseche del sistema strutturale, appare strettamente necessario implementare solamente una *entity* che riporti il coefficiente di smorzamento viscoso. Infatti pulsazione e periodo non smorzati della struttura, così come il coefficiente di smorzamento critico, sono semplici funzioni di massa e rigidità del sistema; mentre pulsazione e periodo smorzato dipendono dai rispettivi omonimi non smorzati e il fattore di smorzamento è dato dal rapporto tra il coefficiente di smorzamento viscoso e il coefficiente di smorzamento critico. L'implementazione di *classes* che possano descrivere queste caratteristiche sarà completamente dipendente da altre che descrivono quelle che le determinano, saranno, sostanzialmente funzione di altre. Si può persino pensare di non implementarle, rimandando al software per le analisi l'onere di ricavarlo, ma non si seguirà questa strada per poter dare al progettista la possibilità di intervenire con considerazioni personali su questi parametri e, quindi, di poter mantenere nello sviluppo del progetto almeno le conseguenze di tutto ciò.

Considerando invece le azioni esterne, per i nostri scopi, è necessario implementare una classe di azioni con le sue caratteristiche, anzitutto per gli spettri in spostamento e in accelerazione, nonostante siano anch'essi funzione di altri parametri. Difatti un'altra soluzione, potrebbe prevedere la descrizione attraverso classi di accelerazione orizzontale massima del sito, di massimo fattore di amplificazione dello spettro in accelerazione orizzontale e periodo di inizio del tratto a velocità costante dello spettro in accelerazione orizzontale, attraverso caratteristiche di sottosuolo e topografiche e i loro relativi fattori, rimandando quindi, il tutto a essere una funzione della collocazione geografica dell'edificio. Questo farebbe sì che anche gli spettri, similmente ai parametri dinamici della struttura, possano essere calcolati dal software strutturale, sulla base di parametri standardizzati già presenti nel file IFC e sulla base dello Stato Limite considerato e della classe d'uso, conseguentemente funzione della vita nominale e della probabilità di superamento dell'azione considerata per il relativo Stato Limite. Il grande svantaggio di un'implementazione siffatta sarebbe la sua relativa inutilità nel caso in cui la

normativa di riferimento evolvesse andando a considerare altri parametri oltre a quelli appena elencati, soprattutto se questi, nuovi, non siano già previsti dall'alfabeto IFC. È sicuramente preferibile, però, poter piuttosto inserire uno, o più, spettri di progetto, o accelerogrammi, rimandando la definizione di questi a una, sempre più da favorire, responsabilità del progettista.

Per quanto poi riguarda il campo più generale di analisi strutturali in ambito sismico, sembra utile poter descrivere anche tutte quelle azioni per analisi non direttamente dinamiche ma che trovano il loro scopo, appunto, nel campo della sismica: stiamo parlando, ad esempio, delle analisi *push-over* e di tutte analisi che si svolgono in ambito statico ma che servono a descrivere il comportamento e la risposta in ambito dinamico. Per questo obiettivo sarà semplicemente necessario implementare una *class* apposita che per descrivere il tutto faccia riferimento ad azioni e spostamenti come già descritti in campo statico attraverso un semplice *attribute*. Ciò per poter discriminare quale, in concreto, sia stato il tipo di analisi svolta.

Nel voler, poi, descrivere i risultati delle analisi, sarà necessario implementare una *superentity* che ritorni, alla fine, nello *Structural Analysis Domain* che raccolga le varie soluzioni dalle analisi dinamiche, che andrà a sua volta a confluire nella *superentity* dei generici risultati delle varie analisi strutturali. In questa *entity* sarà necessario poter inserire informazioni relative a:

- Il numero di sistemi di libertà del sistema;
- Autovalori;
- Autovettori;
- Fattori di smorzamento viscoso modale;
- Coefficienti di Rayleigh;
- Fattori di partecipazione modale;
- Costanti moltiplicative per la normalizzazione, ingegneristica o matematica, degli autovettori;

-Masse partecipanti<sup>62,63,64</sup>.

Ricordando che con gli autovalori si descrivono le pulsazioni naturali del sistema e con gli autovettori le forme naturali di vibrazione, di più ricordando come tutti questi debbano essere collegati, sarà verosimilmente utile sfruttare *attribute* che colleghi entrambi alla forma modale cui fanno riferimento. Infine, per quanto riguarda la dinamica delle strutture, sembra essere poco utile catalogare attraverso nuovi oggetti sforzi e tensioni causate dall'azione sismica, considerando le corrispettive forze statiche equivalenti, in una diversa classe di reazioni rispetto a quella relativa alle analisi statiche. Si vuole, quindi, creare una *class* che, come nel caso delle analisi statiche equivalenti, possa assolvere al compito appoggiandosi a oggetti già descritti nell'ambito statico, particolarizzandoli attraverso una *superentity*.

Di nuovo, nell'ambito statico, la catalogazione di informazioni relative a sforzi, tensioni e deformazioni è tradotta in IFC con la possibilità di inserire attraverso *IfcStructuralReaction*, sottoclasse di *IfcStructuralActivity*, reazioni vincolari, sollecitazioni interne agli elementi e spostamenti, relativamente a determinati punti, curve o superfici. Questo è possibile grazie alle tre sottoclassi: *IfcStructuralPointReaction*, *IfcStructuralCurveReaction* e *IfcStructuralSurfaceReaction*. Poter informatizzare informazioni siffatte in ambito dinamico, oltre all'utilità in fase progettuale, dimostra un'enorme validità in fasi successive del processo edilizio, soprattutto per quanto riguarda il confronto dei risultati ottenuti con schemi strutturali diversi o per la verifica di un edificio esistente. Di nuovo, si vuole implementare una *entity* per le azioni dinamiche, che si appoggi alle equivalenti statiche attraverso una *entity* superiore che le distingua come originate da analisi dinamiche.

---

62 - Francesca Da Porto, "Dinamica dei Sistemi Lineari ad 1 GdL";

63 - Francesca Da Porto, "Dinamica dei Sistemi Lineari ad n GdL";

64 - NAFEMS, "A Finite Element Primer", pag. 170 ss.

## **6. proposte in letteratura**

Nella più che ventennale storia di IFC non sono molte le iniziative che si sono preposte di poter estendere il dominio delle analisi strutturali così da includere i modelli agli elementi finiti e le analisi dinamiche. È quindi sembrato giusto confrontarsi con queste, cercando di capire come sono strutturate, così da capire quali siano i rispettivi punti di forza e di debolezza.

Molto più perché non partire da zero, confermata la validità del punto di partenza, permette di proseguire e migliorare, arrivando così più lontano che non altrimenti.

Si sono trovate, in pratica, due proposte, una datata 2007 e una più recente, di quest'anno corrente, il 2017.

# **6A. proposta gruppo st-7**

Nel luglio 2007 M. Hassanien Serror, Junya Inoue, Yoshinobu Adachi e Yozo Fujino pubblicarono un lavoro intitolato “*Shared computer-aided structural design model for construction industry (infrastructure)*”. Questo partiva dalla considerazione, oramai consolidata, che l’interazione tra i partecipanti alla costruzione di un progetto richiede molto tempo e spesso è causa di errori e fraintendimenti; i miglioramenti quindi di questa capacità di comunicazione possono migliorare notevolmente l’intera industria delle costruzioni.

Il lavoro, poi sottoposto, pare senza ulteriori sviluppi all’allora IAI, oggi diventata buildingSMART, rientrava nel settimo step di sviluppo delle strutture di IFC, nel gruppo cosiddetto ST-7. Si poneva come obiettivo di continuare il lavoro iniziato in ST-4, di descrivere le analisi strutturali, a differenza degli altri gruppi che si erano concentrati sulla schematizzazione e rappresentazione geometrica delle varie tipologie strutturali, anche a seconda del materiale costitutivo.

Basandosi su tre standard, lo standard STEP per lo scambio di dati dei prodotti “*STandard for Exchange of Product Model Data*” della ISO 10303-104 e 107, lo standard “*CIMsteel Integration Standard*” della CIS/2.0 e lo standard delle *Industry Foundation Classes* della International Alliance for Interoperability, sviluppano una proposta per lo modello di trasferimento di informazioni di ambito strutturale che coinvolge per lo più progettisti di strutture, produttori di acciaio e calcestruzzo, construction manager e tutte le figure coinvolte nel ciclo di vita del progetto. Il modello, così, sviluppato è chiamato, come da titolo della pubblicazione, “*Shared Computer-Aided Structural Design*” abbreviato in “modello sCAsD”. Oltre a essere presentato e discusso, si tanta la valutazione della validità e della solidità del modello attraverso un applicativo, oggi irreperibile, che trasmette da un software FEM a un software BIM un semplice modello.

Ovviamente lo standard su cui si interviene per implementarlo è lo standard IFC di buildingSMART alla versione IFC2x, ma il confronto con la risorsa STEP e CIS/2.0 permette, secondo gli autori, di dare al modello un punto di forza nella combinazione di standard per le analisi strutturali e con gli effetti sinergici di IFC. Chiaramente la proposta si confronta con la parte dello standard per software BIM a quello che era lo stato corrente, ovvero con la parte relativa alle analisi strutturali statiche già compiutamente definita, mantenendo l’integrità delle *entities* già definite in IFC. Vengono, così, definiti

quattro processi, e i relativi sotto processi, che definiscono quattro modelli di schemi di dati. Il primo schema è relativo alle analisi dinamiche, il secondo ai modelli agli elementi finiti, il terzo ai risultati delle analisi dinamiche, il quarto alle relazioni tra i primi tre.

Il processo 1 ha due sotto processi, il primo definisce le azioni dinamiche, limitandosi alle sole azioni sismiche, mentre il secondo definisce le condizioni di vincolo esterno in termini dinamici. Questo modificherà lo *Structural Load Resource*;

Il processo 2 ha, invece, tre sotto processi, il primo atto a definire le *entities* di un modello agli elementi finiti, il secondo le integra le une con le altre, mentre la terza le collega alle azioni strutturali. Viene pensato ad uno nuovo *data schema*, relativo al *Domain Layer*, chiamato *Finite Element Analysis Domain*;

Il processo 3 ha due sotto processi, il primo integra i *data schema* *IfcMeasureResource* e *IfcStructuralAnalysisDomain* definendo le *entities* base all'interno del secondo *data schema* per descrivere i risultati delle analisi strutturali e, nel primo *data schema*, le risorse per misurarle. Il secondo processo vuole assegnare i risultati delle analisi strutturali al modello agli elementi finiti, ovvero integra il nuovo *data schema* del *Finite Element Analysis Domain*.

Il processo 4 ha due sotto processi, il primo definisce le relazioni per l'integrazione meccanica tra il modello agli elementi finiti e il modello architettonico, il secondo definisce le relazioni per l'integrazione fisica. Tutto ciò avviene nel *Finite Element Analysis Domain*.



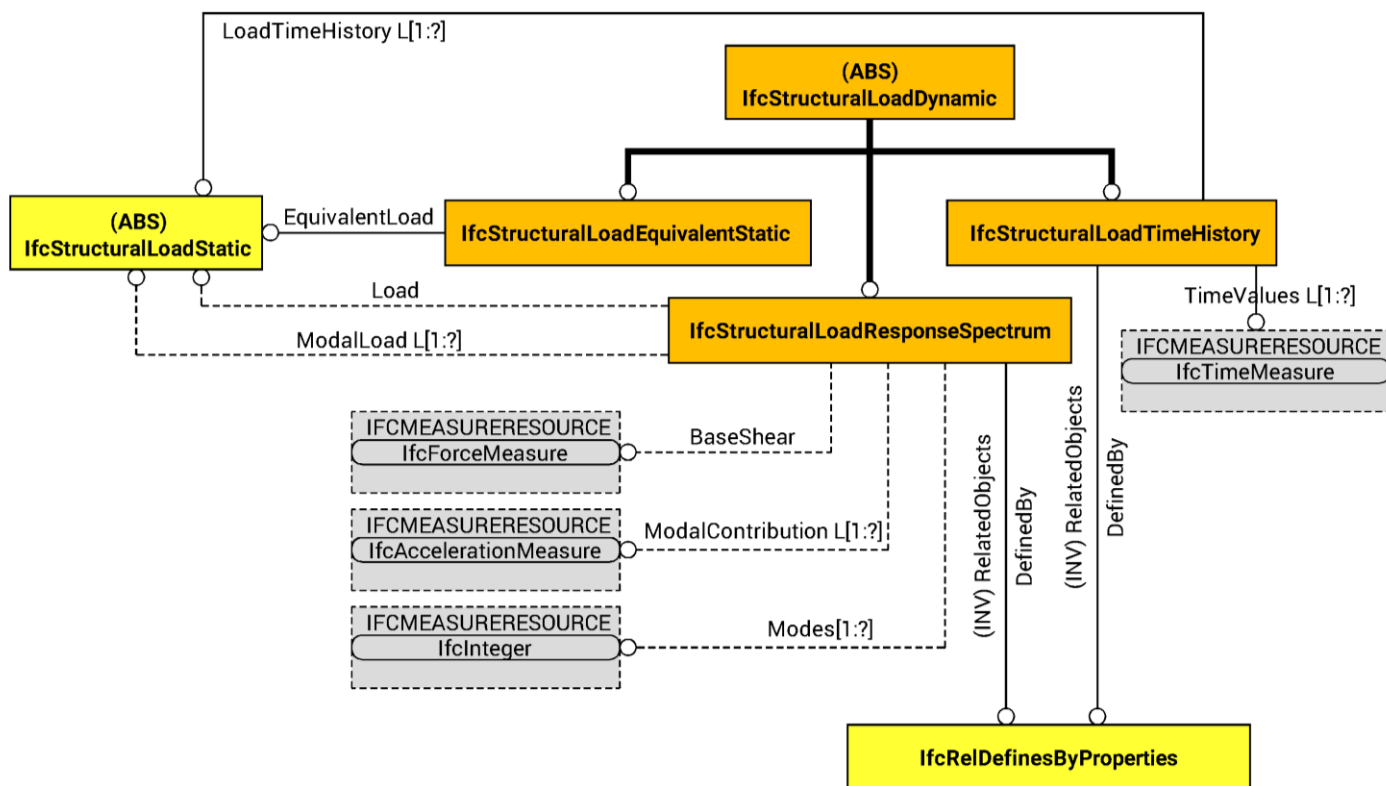


Figura 47 - Proposta ST-7. Carichi dinamici. In arancione le *entities* proposte, in giallo quello già esistenti.

## DYNAMIC ANALYSIS SCHEMA

Per quanto riguarda la definizione delle azioni dinamiche, si individuano due “input”: la risorsa che descrive le azioni strutturali e la risorsa che descrive le proprietà. La definizione delle azioni dinamiche è controllata dal tipo di carichi statici equivalenti, dalla norma tecnica utilizzata per la definizione dello spettro di progetto, nonché dalla storia dei terremoti. Vengono, così definite tre categorie di azioni dinamiche, più precisamente, tre *subentities* della nuova *superentity* **IfcStructuralLoadDynamic**.

-La prima è **IfcStructuralLoadEquivalentStatic**, per descrivere, come da nome, un’azione meccanicamente equivalente come risposta all’azione dinamica;

-La seconda è un'*entity* che descrive lo spettro di progetto, **IfcStructuralLoadResponseSpectrum**, che, eloquentemente, specifica lo spettro di progetto come calcolato dalla norma prescelta oppure uno spettro, più irregolare, effettivamente accaduto e descrivente, quindi, un reale terremoto;

-La terza è **IfcStructuralLoadTimeHistory** e specifica un'azione dinamica calcolata sulla base delle componenti in spostamento, velocità e accelerazione di un terremoto effettivamente avvenuto.

Le azioni dinamiche così definite attraverso azioni statiche equivalenti, possono essere referenziate ad azioni statiche, o a una lista di azioni statiche, per ereditarne gli *attributes* e le caratteristiche in genere.

È necessario, inoltre, secondo gli autori, ampliare le *entities* che descrivono le condizioni di vincolo in modo che possano descrivere non solo vincoli linearmente elastici, ma anche un comportamento in genere non lineare. Deve essere inoltre possibile, secondo gli autori, descrivere accelerazione, velocità e spostamento dei vincoli esterni attraverso la *entity* *IfcBoundaryNodeConditionDynamic*. Non implementano però alcunché.

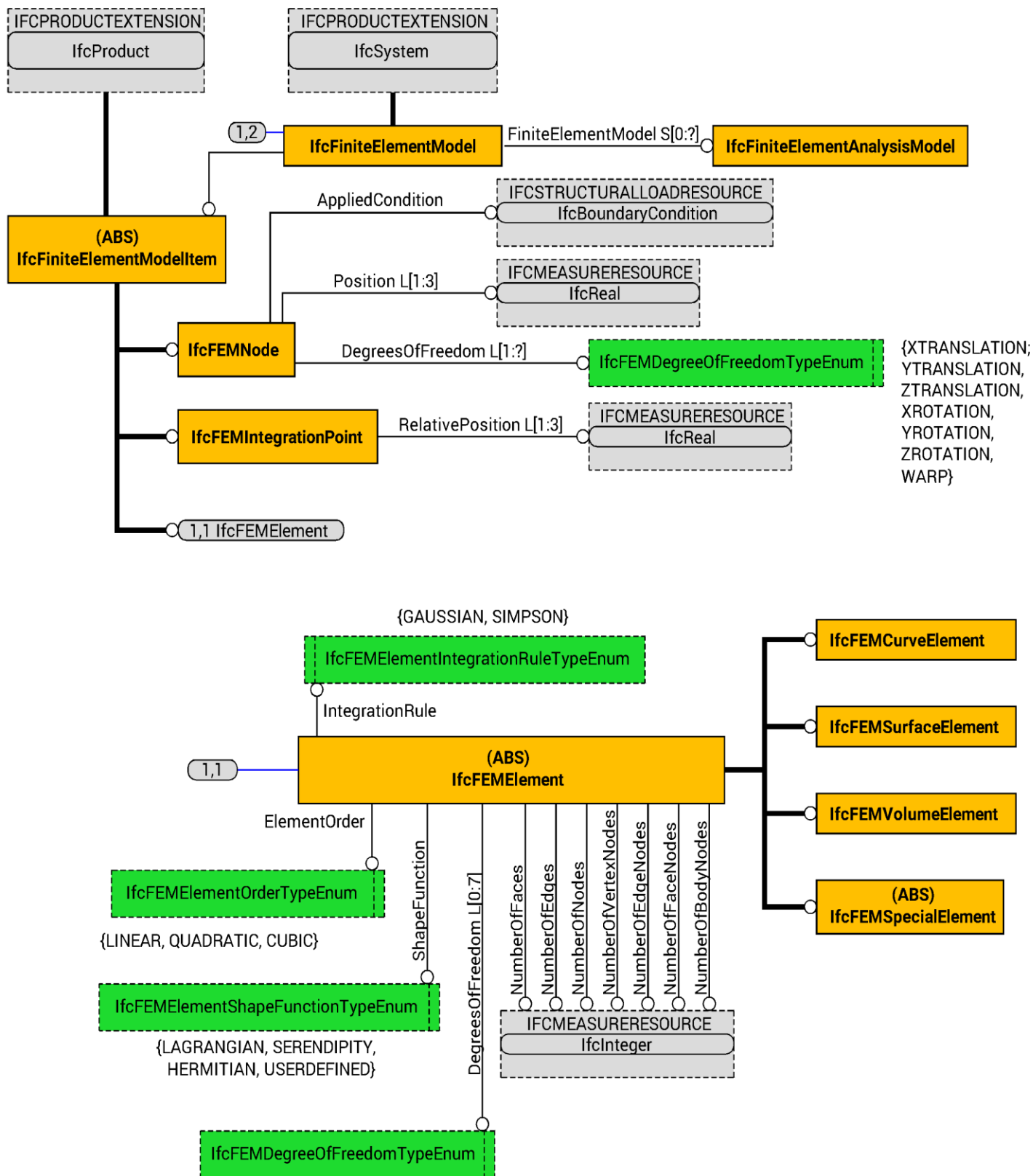


Figura 48 - Proposta ST-7. Modello agli elementi finiti. In arancione le entities proposte, in giallo quello già esistenti.

## FINITE ELEMENT MODEL SCHEMA

Per descrivere compiutamente un modello agli elementi finiti è necessario implementare la rappresentazione topologica delle varie *entities* FEM e le risorse delle attività strutturali che vi devono essere assegnate; per rappresentare compiutamente è necessario specificare i gradi di libertà, le condizioni di vincolo, la dimensionalità spaziale e il comportamento meccanico dell'oggetto. Per fare ciò gli autori propongono un nuovo *data schema* chiamato *Finite Element Analysis Domain*.

Le entità del modello agli elementi finiti, siano elementi, nodi o punti di integrazione sono descritti attraverso l'*abstract entity* **IfcFiniteElementModelItem**, che è contenuta nell'*entity* **IfcFiniteElementModel** a sua volta riferita a **IfcFiniteElementAnalysisModel**, che combina i modelli FEM integrati con i carichi applicati e i risultati delle analisi. *IfcFiniteElementModelItem* ha tre *subentities* che gli si riferiscono, per poter, appunto descrivere elementi, nodi o punti di integrazione.

-**IfcFEMNode** ha tre *attributes*: *Position*, che lista le specifiche coordinate del nodo; *DegreesOfFreedom* che, invece restituisce i gradi di libertà dello stesso nodo considerato; infine l'*attribute* opzionale *AppliedCondition* che si riferisce a *IfcBoundaryCondition*. Grossolano errore: per descrivere la posizione del nodo si utilizza la classe *IfcReal* listando tre valori reali, considerabili quindi come adimensionali all'interno di IFC, invece della più corretta *IfcLocalPlacement*;

-**IfcFEMIntegrationPoint** in cui è nuovamente presente un *attribute* opzionale, *RelativePosition*, per specificare la posizione relativa del punto di integrazione rispetto al relativo elemento FEM di riferimento, anche qui viene, però, utilizzato *IfcReal* invece di *IfcLocalPlacement*;

-**IfcFEMElement** è a sua volta un'*abstract entity* per le entità FEM che sono state categorizzate a loro volta, sulla base della loro dimensionalità: elementi curvi, superficiali o volumetrici, oppure come elementi particolari: molle, smorzatori o masse.

La metodologia adottata per integrare le entità FEM all'interno del modello è di riferire ai nodi FEM e ai punti di integrazione l'*entity* *IfcFEMElement* attraverso gli *attributes* *Connectivity* e *IntegrationPoint* rispettivamente. Mentre gli elementi FEM sono riferiti a *IfcFEM* attraverso l'*attribute* *Elements*.

Qui si mostra anche un grossolano errore di modellazione della classe: si riferiscono a questa *entity* una serie di *attributes*, descritti molto genericamente attraverso *IfcInteger*, che fanno riferimento a nodi o punti di integrazioni, oppure sono caratteristiche collegate alla dimensionalità dell'elemento. In altre parole di attribuiscono ad *IfcFEMElement* delle proprietà che o appartengono intrinsecamente ed esclusivamente altri oggetti di gerarchia inferiore, quindi ridondanti, oppure che sono relativamente inutili come, ad esempio, il numero di vertici e bordi di un elemento. Inutili perché ereditando, attraverso la gerarchia una certa geometria, risulta superfluo riportare, ad esempio, il numero di vertici e di spigoli di un tetraedro. La loro unica utilità è di poter far capire come descrivere fin troppo compiutamente un modello per analisi agli elementi finiti. Questo è stato fatto intenzionalmente per poter riferire gli *attributes* alla loro *abstract superentity*.

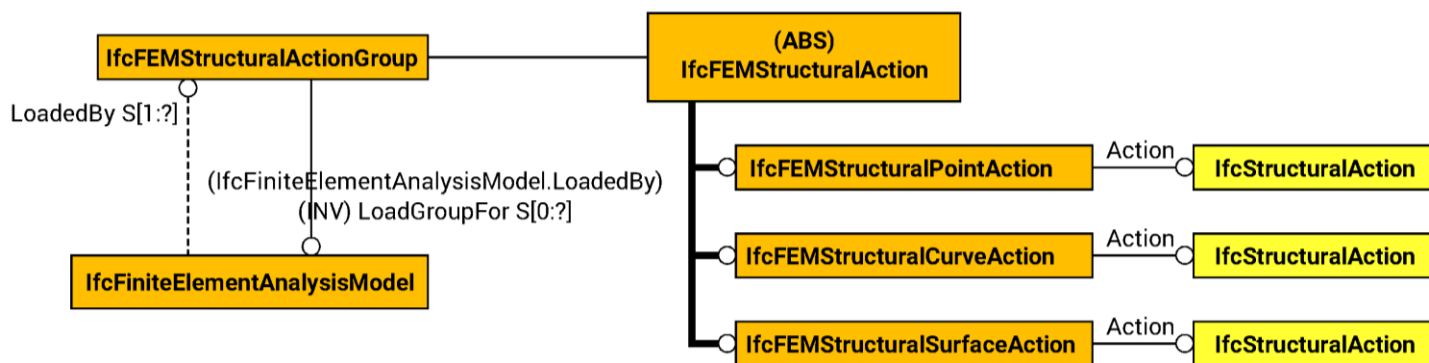


Figura 49 - Proposta ST-7. Azioni sul modello agli elementi finiti. Ancora in arancione le nuove *classes*, in giallo quelle già esistenti.

-Per la connettività delle azioni alle entità del modello viene utilizzata **IfcFEMStructuralAction**. Anche questa viene definita *abstract* e vi ci riferiscono tre *subentities* dimodoché ciascuna descriva un'azione che insiste su un elemento mono, bi o tridimensionale. Ognuna, infatti, attraverso l'*attribute Action* viene riferita a *IfcStructuralAction* e si collega, così facendo, la richiesta azione strutturale statica o dinamica all'entità FEM relativa.

-Sarà poi presente **IfcFEMStructuralActionGroup** a cui verrà riferito un set di *IfcFEMStructuralAction*. A sua volta poi sarà riferito a un'entità *IfcFiniteElementAnalysisModel* attraverso l'attribute *LoadedBy*.

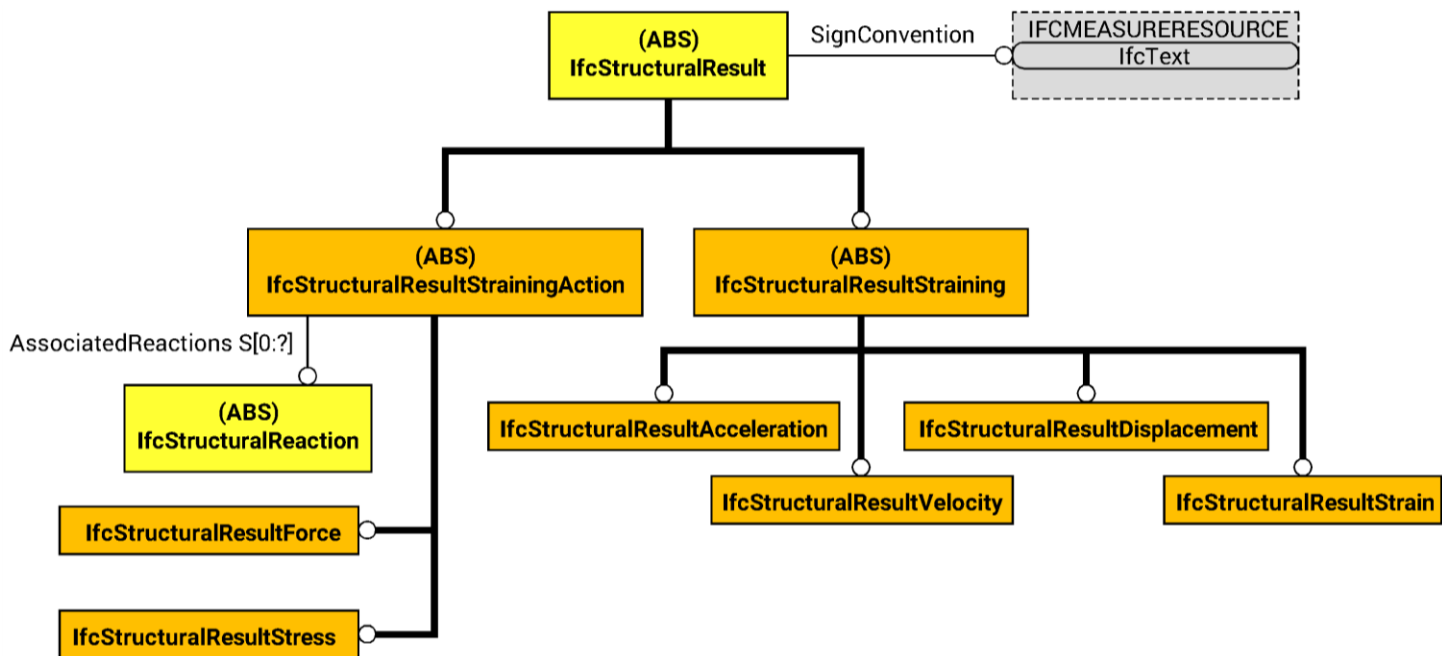


Figura 50 - Proposta ST-7. Risultati di analisi strutturali. Ancora in arancione le nuove *classes*, in giallo quelle già esistenti.

## STRUCTURAL ANALYSIS RESULTS SCHEMA

Sarà necessario, per poter salvare in IFC i risultati di un'analisi strutturale, la risorsa per le misure, per poter quantificare i risultati e la risorsa per le azioni, anche dinamiche, dalle quali le analisi vengono condotte, ricordando inoltre che in IFC con “azioni” possono essere descritte azioni e reazioni propriamente dette, siano esse interne o esterne. Molto importante ricordare, inoltre, che alla data in cui la proposta veniva scritta, *IfcStructuralLoad* e *IfcStructuralResult* sono due diverse *entity* non ancora raggruppate in un'unica *IfcStructuralLoadOrResult*. Alla data attuale questa proposta ha in gran parte perso di significato, se non per la parte relativa a risultati di spostamento, velocità e accelerazione degli elementi, ma anche questa rimane da sistemare.

Perciò viene presa l'*abstract superentity*: *IfcStructuralResult* a cui si riferiscono due *entities* in base alla categoria di risultato.

-**IfcStructuralResultStrainingAction** è la prima *subentity*, a sua volta del tipo *abstract*. Le si collegano due *subentities* per poter descrivere le forze e le tensioni interne.

-**IfcStructuralResultStraining** è la seconda *subentity*, anch'essa del tipo *abstract*, definita per descrivere, attraverso le sue *subentities*, deformazioni, spostamenti, velocità e accelerazioni.

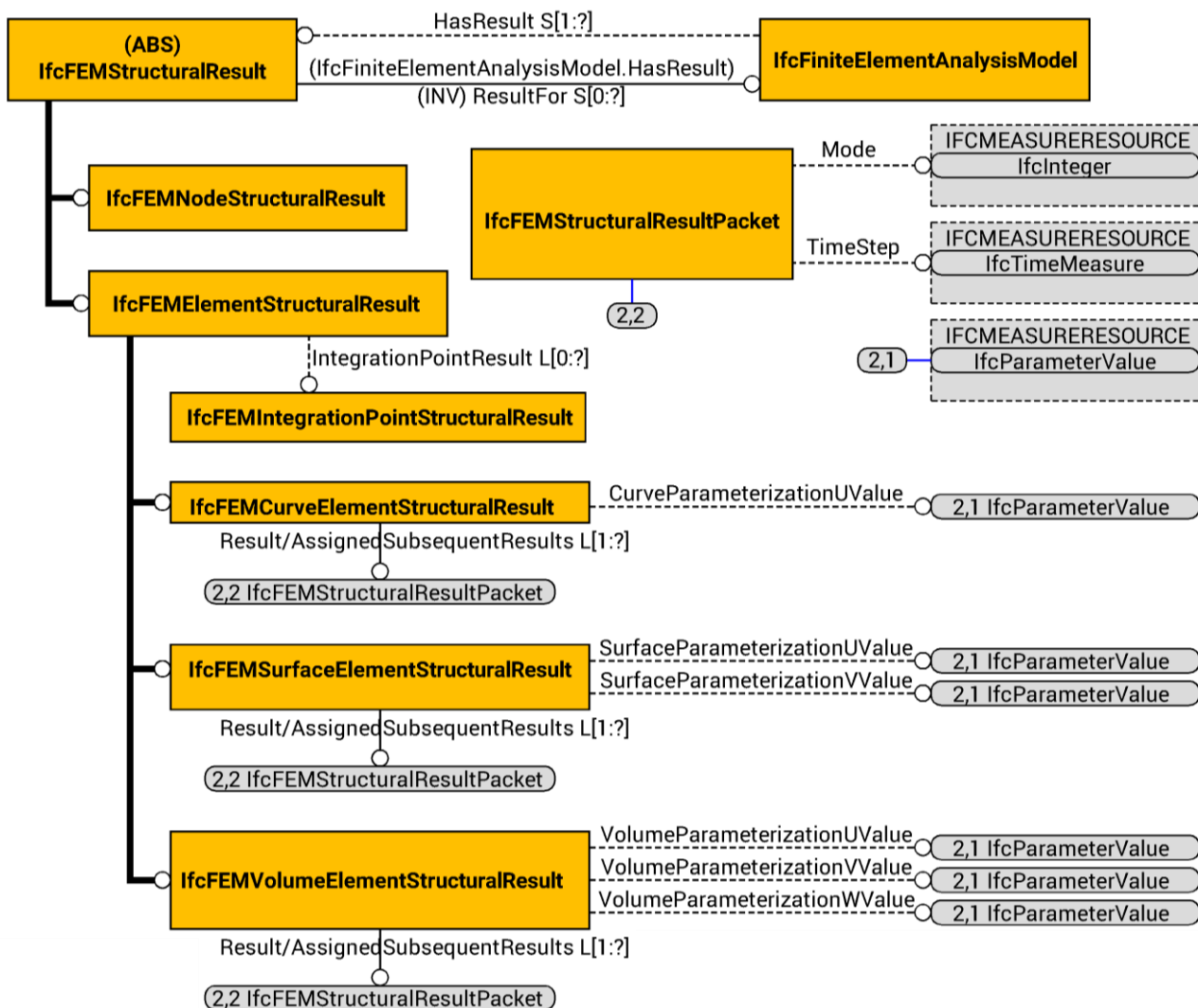


Figura 51 - Proposta ST-7. Risultati di analisi sul modello agli elementi finiti.

Per assegnare i risultati delle analisi al modello agli elementi finiti, viene definita **IfcFEMStructuralResult**. Questa è un'abstract entity a cui si riferiscono *IfcFEMNodeStructuralResult* e

*IfcFEMElementStructuralResult*, che, come da nome, servono a descrivere, rispettivamente, i risultati strutturali dei nodi e degli elementi del modello agli elementi finiti.

-**IfcFEMElementStructuralResult** è, a sua volta, un'abstract superentity per risultati di elementi strutturali mono, bi o tridimensionali. Ognuna di queste *subentities*, poi, è riferita a una lista della *entity* **IfcFEMStructuralResultPacket** attraverso l'attribute *Result/AssignedSubsequentResults* per scegliere il relativo risultato strutturale statico o dinamico all'interno di un insieme e per assegnarlo alla pertinente entità del modello agli elementi finiti. *IfcFEMStructuralResultPacket* è considerata essere l'entity che provvede a descrivere risultati di analisi sia statiche che dinamiche, attraverso i suoi *attributes*. Questi potranno descrivere risultati modali e al variare del tempo attraverso gli *attributes* *Mode* e *TimeStep*. Siccome al variare dell'elemento FEM, mono, bi o tridimensionale, possono variare i risultati, i valori di parametrizzazione sono stati utilizzati per descrivere la distribuzione del risultato strutturale. Ad esempio, la *subentity* **IfcFEMVolumeElementStructuralResult** ha tre *attributes*, *VolumeParameterizationUVvalues*, *VolumeParameterizationVVvalues* e *VolumeParameterizationWVvalues* che contengono i valori dei parametri U, V e W che sono necessari a descrivere la posizione del risultato sul relativo *IfcFEMVolumeElement*. Quindi dall'attribute *AssignedSubsequentResults* i valori dei risultati delle analisi statiche o dinamiche sono assegnati alle posizioni così definite. In più, i risultati possono essere assegnati ai punti di integrazione dell'elemento FEM attraverso l'attribute *IntegrationPointResult* che collega una *entity* *IfcFEMIntegrationPointStructuralResult*; i raggruppamenti di risultati FEM possono essere gestiti attraverso *IfcFEMStructuralResultGroup*. L'entity *IfcFiniteElementAnalysisModel* collega tutte le *entity* *IfcFiniteElementModel* definite attraverso l'attribute *FiniteElementModel* e i gruppi di azioni e risultati FEM sono assegnati al modello per le analisi attraverso l'attribute *LoadedBy* e *HasResult* rispettivamente.



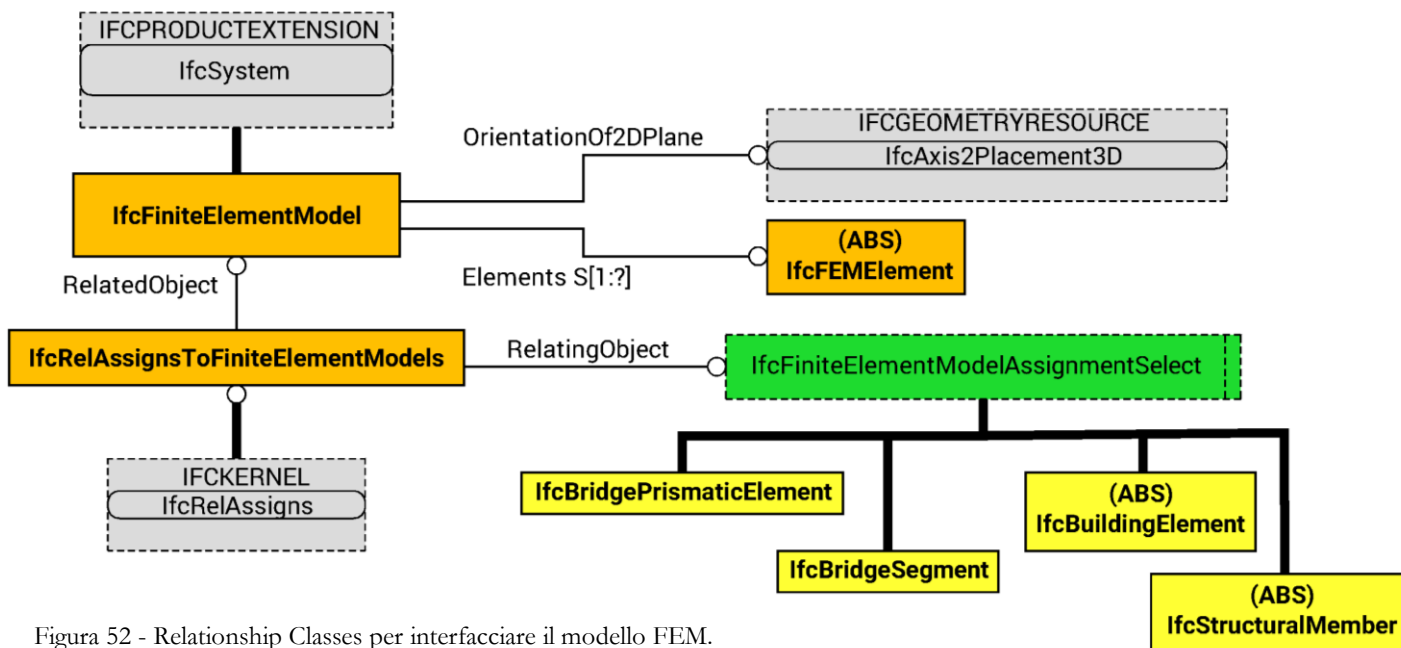


Figura 52 - Relationship Classes per interfacciare il modello FEM. Di nuovo in arancione le nuove *classes*, in giallo quelle già esistenti.

## RELATIONSHIP SCHEMA

È necessario a questo punto fornire metodologie di collegamento tra il modello architettonico e il modello FEM, definiti entrambi, oramai, in IFC. Sono state fornite due metodologie di integrazione: FEM meccanico e FEM fisico. Nel primo caso, il modello meccanico è prima estratto dal modello architettonico attraverso un processo di idealizzazione e, dopo, di discretizzazione in un modello FEM, nel secondo caso, il modello FEM è estratto direttamente dal modello architettonico unendo i due processi, di idealizzazione e discretizzazione, in un unico processo ed estraendo implicitamente, usando altri termini, il modello meccanico.

La definizione delle relazioni per il modello FEM meccanico è controllato dalla successione delle *entities* sia del modello architettonico, sia del modello meccanico, mentre la definizione delle relazioni per il modello FEM fisico è controllata solamente dalla successione delle *entities* architettoniche. Così

facendo l'*entity* definita permette di assicurare l'integrità tra modello FEM, meccanico e architettonico e, così facendo, sono assicurate e sistemate discrepanze tra i modelli.

Viene, così, definita l'*entity IfcRelAssignsToFiniteElementModels* che assegna una o più *entity* del tipo *IfcFiniteElementModel* al modello architettonico o meccanico. Ciò avviene attraverso il *type IfcFiniteElementModelAssignmentSelect* che riferisce o al modello meccanico, attraverso *IfcStructuralMember* o al modello fisico attraverso *IfcBuildingElement*, oppure *IfcBridgeSegment* o *IfcBridgePrismaticElement*.

## VALUTAZIONE ANALISI DINAMICHE

Ciò che appare essere un punto di forza della proposta del gruppo ST-7 è la classificazione delle analisi dinamiche, che, in maniera ampia prevede una nuova classe che raccoglie tutte le azioni, statiche e dinamiche, utilizzate nell'ambito di analisi dinamiche. Di più, le azioni dinamiche considerano anche azioni di progetto espresse come spettri e azioni di progetto risultanti dal variare di spostamenti, velocità e/o accelerazioni al suolo dati da un effettivo terremoto. Tutto ciò è da valutare ancora più positivamente poiché tutto ciò si verifica come modifica del già esistente *data schema Structural Load Resource*. La proposta per quanto riguarda tutto ciò è da valutarsi più che positivamente, soprattutto perché facilmente aggiornabile e integrabile con la proposta Softing, che vedremo poi; difatti, come si accennava prima la proposta si inserisce all'interno della versione IFC2x, decisamente datata e superata alla data attuale. Ad esempio, sono ancora previste le due distinte classi, *IfcStructuralLoad* e *IfcStructuralResult*, invece dell'odierna *IfcStructuralLoadOrResult* che le unisce.

Da un lato, la proposta si presenta completa ed efficace, sebbene imprecisa, per quanto riguarda i dati in ingresso, cioè azioni sollecitanti la struttura; dall'altro lato appare gravemente incompleta per quanto riguarda l'espressione di dati in uscita, assolutamente scarna e imprecisa. Difatti l'imprecisione dei dati in ingresso riguarda il fatto che smorzamenti e massività della struttura devono fare parte del modello analitico e non solamente del modello agli elementi finiti, come invece succede, altrimenti l'informazione della struttura risulta gravemente incompleta sia come correttezza dell'informazione, sia nell'ottica di voler standardizzare dati per software di calcolo diversi da quelli agli elementi finiti. Di più mancano comunque informazioni come frequenze di vibrazione della struttura, smorzato o meno, pulsazioni, fattori di smorzamento e fattori di decadimento. Per quanto poi riguarda i dati in uscita, è sì molto interessante e da valutare positivamente la possibilità di descrivere risultati come la velocità, lo spostamento, l'accelerazione e, soprattutto, la deformazione di un elemento sollecitato dinamicamente; anche se sono comunque da antecedere i risultati così come previsti anche da NAFEMS, ovvero ampiezza e fase delle oscillazioni, il numero di gradi di libertà, autovalori, autovettori, forme modali e fattori di smorzamento viscoso modali. Solo le forme modali sono implicitamente considerate, poiché, nel descrivere un'accelerazione spettrale si considera la massa

partecipante per ogni modo di vibrare che mette in gioco lo spettro suddetto. Non sono però da scartare nell'ottica di una descrizione dettagliata di un'analisi dinamica, ma rimane in ogni caso assolutamente necessario esplicitare le soluzioni così definite per ogni forma modale cui corrispondono.

Per quanto poi riguarda le azioni definite nel modello agli elementi finiti, si rivelano essere decisamente sovrabbondanti: nel caso in cui non fosse stato creato un nuovo *data schema*, ma fosse ampliato lo *Structural Analysis Domain*, come tra l'altro buildingSMART richiedeva, non sarebbe stato affatto necessario implementare queste *classes* di azioni, che ricalcano e richiamano le azioni sul modello analitico. Sono quindi sostanzialmente evitabili e, di conseguenza, si vorrà evitarle, facendo sì che modello analitico e modello agli elementi finiti dialoghino affinché le azioni del primo siano riversate sul secondo, appartenente allo stesso dominio.

Altra implementazione evitabile, ma forse utile nell'ottica di possibili sviluppi futuri, anche se è molto difficile, è la creazione dell'*entity IfcBoundaryNodeConditionDynamic*. Di per sé è evitabile poiché qualsiasi solutore si riconduce, da un'accelerazione, una velocità e/o uno spostamento impressi ai vincoli ad una impresa alla struttura in direzione contraria. È un'approssimazione ampiamente accettata e utilizzata e l'estensione con una tale *entity* è molto snella e veloce tutto sommato: nell'ottica del lungo periodo può essere utile mantenere.

## VALUTAZIONE MODELLO AGLI ELEMENTI FINITI

La proposta avanzata dal gruppo di ricerca presenta degli spunti e delle idee molto interessanti che non possono non essere tenuti in considerazione, soprattutto nell'ottica di sviluppi futuri; purtroppo però è, in generale, da scartare, anche solo per il fatto che la stessa buildingSMART richiede, oggi giorno almeno, che la rappresentazione topologica e la definizione fisico-meccanica di un modello agli elementi finiti sia effettuata all'interno dello *Structural Analysis Domain*, non in un *data schema* a sé stante, come invece si propone.

Andando poi a guardare più nel dettaglio appare come la proposta in sé presenta, come già accennato, grossolane ingenuità nella modellazione delle classi IFC, che, abbinate a una dettagliata ed esaustiva conoscenza di modelli e analisi agli elementi finiti, lascia abbastanza spiazzati. Il primo fatto a cui ci si ritrova davanti è la definizione di *IfcFEMElement*: questa *abstract entity* viene definita dalle sue proprie *subentities* atte a descrivere la dimensionalità dell'elemento. Le vengono però definiti alcuni *attributes* che con l'elemento finito in sé hanno ben poco a che fare: ossia, ad esempio, si enumerano i gradi di libertà e le funzioni di forma dei nodi, nonché vengono listati i punti di integrazione dell'elemento, con una cardinalità minima pari a 1, e la regola di integrazione utilizzata nel calcolo del punto di integrazione stesso. Non sono di per sé informazioni sbagliate, quanto piuttosto è incorretta la loro implementazione in EXPRESS-G: funzioni di forma e gradi di libertà devono essere *attributes* dei nodi, non dell'elemento in sé; così come non è corretto definire obbligatoriamente almeno un punto di integrazione per un elemento, poiché può non essere utile, né definire la regola di calcolo utilizzata per calcolarlo all'interno dell'elemento proprio perché è un'informazione relativa al nodo e non all'elemento finito. Di più, è molto errato il fatto che poi venga correttamente utilizzato lo stesso *enumerated type* per descrivere i gradi di libertà di un nodo, riscontrando una sovrabbondanza inutile ritrovandolo riferito ad *IfcFEMElement*. Oppure che le posizioni, assolute o relative, di nodi e punti di integrazione rispetto ai rispettivi elementi, siano espresse con *IfcReal* attraverso un *attribute*: difatti è adimensionale; va, piuttosto, utilizzata *IfcObjectPlacement* che riporta un posizionamento relativo e rispetta la regola di buildingSMART di preferire i *defined data types* ai *simple data types* per un'informazione più corretta. Questo è tutto sommato corretto logicamente secondo EXPRESS, ma errato secondo le

*Industry Foundation Classes* che, per completezza di informazione vuole la definizione di *defined types*, invece che utilizzare il generico *simple data type*.

Sono poi necessarie alcune precisazioni riguardo *IfcFEMSpecialElement*. Difatti è ampiamente evitabile descrivendo correttamente l'elemento nei gradi di libertà dei suoi nodi.

Se, però, questi sono i difetti della proposta, è evidente che sia concettualmente molto buona e che mostri una conoscenza molto ampia della materia. È molto positiva la costruzione basata del modello agli elementi finiti sull'esempio del modello analitico, che collega tra di loro, attraverso un'apposita *relationship class*, lasciando estrema flessibilità di modellazione. Estremamente positiva e flessibile anche la definizione dei nodi, che lascia possibilità di specificazione dei gradi di libertà, così come vincoli dei nodi. Appare evidente quanto questo aspetto sia da mantenere, soprattutto in un'ottica in cui le *Industry Foundation Classes* vadano a descrivere l'intero settore delle costruzioni, poiché in molti casi elementi con una loro dimensionalità nel modello analitico possono essere modellati con un'altra dimensionalità nel modello agli elementi finiti, ad esempio per ottenere risultati particolarmente precisi.

Il vero, enorme, lato positivo di questa costruzione, di per sé sufficiente a volerla mantenere così, è la possibilità di effettuare micro, meso e macro modellazione di una struttura agli elementi finiti o applicare particolari tecniche di modellazione. Questo proprio perché il modello analitico e quello agli elementi finiti sono indipendenti, sebbene dialoghino e non siano scollegati.

Tanto semplice quanto efficace, poi, la compiuta definizione dei punti di integrazione che si vuole riprendere.

Inutile il definire un nuovo *data schema*, contro le specifiche richieste di buildingSMART, che, probabilmente, porta a un'eccessiva verbosità del tutto: completamente inutile la definizione di specifiche azioni per il modello agli elementi finiti nel caso in cui i due *data schemas* vadano a coincidere, come specificato in seguito. *IfcRelAssignsToFiniteElementModels* infatti assegna a vari oggetti IFC il modello agli elementi finiti attraverso un *select type*. Ciò risulta utile, sebbene in maniera limitata, perché collega a questo modello o un elemento *IfcBuildingElement*, oppure uno *IfcStructuralMember*, o un *IfcBridgeSegment*, o, infine, *IfcBridgePrismaticElement*. Alla data attuale lo standard *IfcBridge* è una sottoclasse di *IfcCivilElement* e fa parte di un progetto in corso d'opera di buildingSMART per poter

arrivare a descrivere anche infrastrutture, e, inoltre, è molto interessante, nonostante forse la scarsa applicabilità, poter descrivere agli elementi finiti un oggetto che non è schematizzabile in un modello analitico, attraverso *IfcBuildingElement*. Tuttavia, per quanto sovrabbondante, non è appunto inutile, e si vuole mantenere questa discriminazione per descrivere, ad esempio, elementi non strutturali quali i parapetti.

Passando a valutare quella che è la *class* di assegnazione dei risultati di un modello agli elementi finiti si riscontra una molto buona struttura concettuale, con qualche precisazione da fare. Difatti la definizione dell'*entity IfcFEMStructuralResultPacket*, a cui gli elementi del modello si riferiscono attraverso l'*attribute Result/AssignedSubsequentResults*, è evitabile e da evitare: evitabile perché basta riferire a *IfcFEMStructuralResult* un'*attribute* e il suo inverso diretti verso *IfcStructuralAnalysisModel*, rimuovendo la necessità di esplicitare a quale modello per le analisi i risultati si riferiscono. Con una siffatta relazione biunivoca tra modello analitico per le analisi e risultati del modello agli elementi finiti non è più che a ogni risultato dell'elemento sia riferito a che tipo di analisi, statica o dinamica, si riferisce; aggiungendo, allo stesso modo, anche una più specifica definizione per i risultati dei nodi. Sarà, per contro, da aggiungere un *attribute* verso *IfcFiniteElementModel* per poter ricondurre ad uno specifico modello agli elementi finiti i suoi propri risultati. Ma è tutto qui il perché l'implementazione analizzata è "da evitare": perché con due soli *attributes* si può molto snellamente evitare la definizione di tre nuove *subentities*, listate in quantità, con due *attributes* ciascuna.

Infine c'è una ridondanza nell'implementare *IfcFiniteElementAnalysisModel*. Questa *entity* lega il modello agli elementi finiti, le azioni che insistono su di esso e i risultati delle analisi condotte. Ridondante perché è di per sé sufficiente collegare via *attributes* direttamente queste tre *superentities* per avere un'informazione completa e perfettamente coerente.

## **6B. proposta softing srl**



Nel corso di questo 2017, la software house italiana Softing srl, nelle persone di Roberto Spagnuolo e Amedeo Farello, è andata a sviluppare una proposta di modifica dello standard IFC, partendo dalla considerazione che questo non descrive quelli che sono il problema sismico e la schematizzazione e discretizzazione agli elementi finiti delle strutture. Le modifiche in questione interessano due schemi: lo *Structural Load Resource*, relativo al *Resource Layer*, e lo *Structural Analysis Domain*, relativo al *Domain Layer*. Altri due schemi di interesse strutturale, che però non vengono modificati, sono lo *Shared Building Elements* e lo *Structural Elements Domain* anch'essi relativi al *Domain Layer*.

Spagnuolo e Farello, prima di avanzare la proposta di modifica, fanno alcune considerazioni su IFC. L'azienda valuta l'interoperabilità attraverso IFC, rilevando come sia importante, per uno strutturista, poter disporre di un modello basato sullo schema dello *Structural Analysis Domain* poiché questo sarà sostanzialmente un modello analitico pronto all'uso, nonché con la possibilità arricchire il file, di volta in volta, con informazioni, considerazioni, verifiche, post-elaborazioni o quant'altro possa essere utile. Difatti, dato il numero molto ridotto di applicazioni che riescono a gestire lo *Structural Analysis Domain*, oggi la pratica più diffusa per un software per strutture consiste nello sfruttare un modello basato sullo *Shared Building Elements* ed eventualmente sullo *Structural Elements Domain*. Il problema principale di ciò è la necessità di convertire, ogni volta che si riveli necessario, il modello architettonico in un modello analitico, che risulterebbe non presente nel file IFC, con la tutt'altro che interoperabile conseguenza di dover sistematicamente archiviare il complesso di informazioni, come il modello analitico, i risultati di analisi e le verifiche, nel formato proprietario dell'applicazione, ripetendo ad ogni più piccola modifica la conversione di formato e l'inserimento di carichi, vincoli, eccetera.

Altro aspetto che Softing considera è la difficoltà di conversione in un modello analitico, dovuta alla modellazione informatica della geometria: ad esempio la miglior rappresentazione per il modello analitico è basata su *IfcExtrudedAreaSolid*, in cui, come da nome, il solido è descritto dall'estrusione di un'area lungo un percorso, che si traduce in un elemento monodimensionale con una sua sezione. Altri tipi di rappresentazione, come *IfcFacetedBrep*, possono presentare ambiguità tipologiche difficilmente, o assolutamente, interpretabili per un software agli elementi finiti, che può leggere il solido sia come un elemento monodimensionale variamente orientato, sia come un elemento

bidimensionale dotato di spessore, anch'esso di giacitura qualunque rispetto agli assi del solido, sia, addirittura, come un elemento tridimensionale.

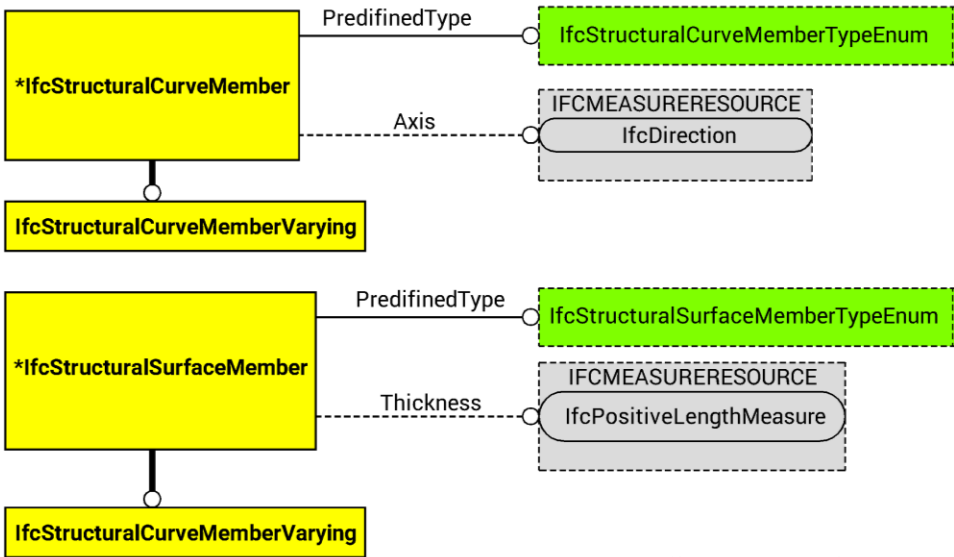
Un vantaggio sottolineato dai due programmatori nel disporre di due schemi distinti per progetto architettonico e strutturale è il poter mantenere separate le due descrizioni geometriche, potendo attribuire l'onere di mantenere la congruenza agli strumenti che operano sui dati e non al formato in sé.

È sulla base di queste considerazioni che Spagnuolo e Farello avanzano una proposta di estensione dello standard IFC, cercando di uniformarsi all'architettura di questo e di apportare il minor numero di modifiche possibili. Con le modifiche proposte sperano di poter gestire azioni e reazioni nel campo della dinamica delle strutture, all'interno di IFC, che si rivela essere già ben definito e molto completo, senza modificare un eccessivo numero di classi, né aggiungendo entità o tipi superflui. In sostanza, migliorando ragionevolmente il linguaggio esistente, che ha poche mancanze.

Per poter, però, affrontare adeguatamente il problema strutturale, sulla base di quanto appena detto, risulta, secondo Spagnuolo e Farello, necessario anche poter classificare e informatizzare anche un modello agli elementi finiti, sia nella classificazione topologica, che nella rappresentazione fisica, che negli sforzi e spostamenti ai nodi.

La proposta si limita all'implementazione agli elementi finiti di elementi mono e bidimensionali, ritenendo di facile implementazione l'estensione a elementi tridimensionali.

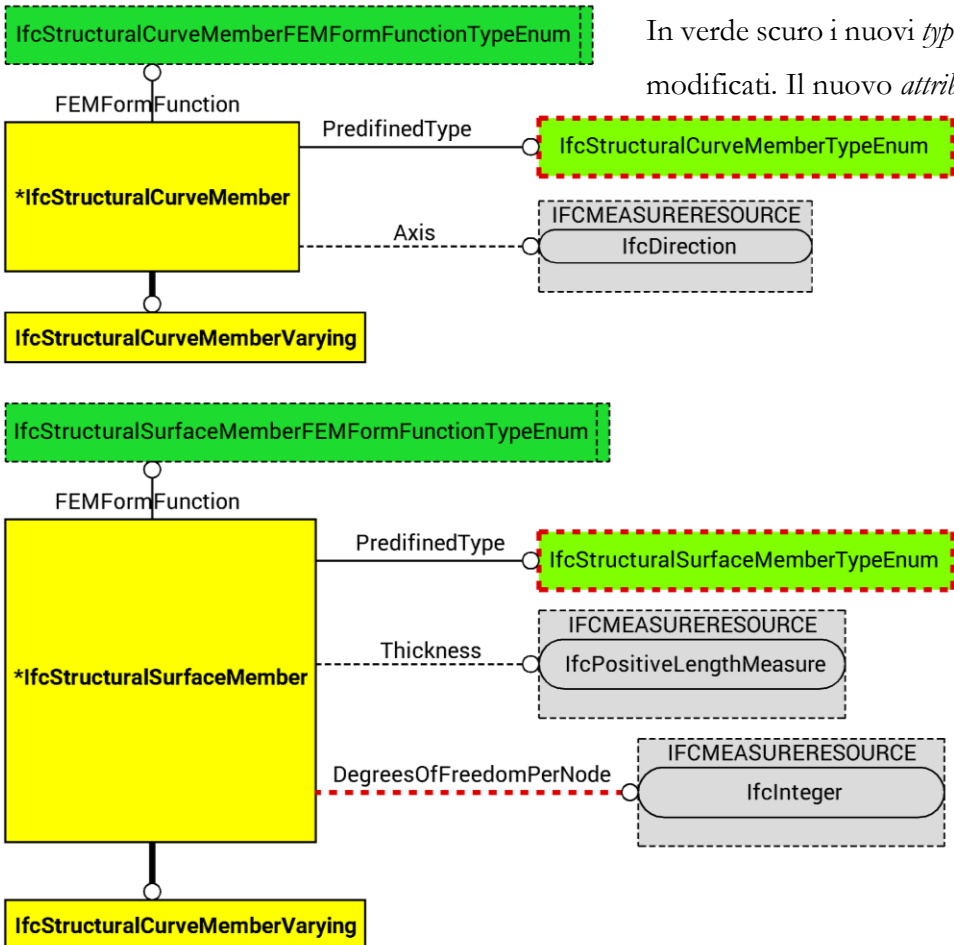
Figura 53 - Stato di fatto e proposta di modifica Softing srl dello SAD



Allo stato attuale

Proposta di modifica di Softing srl:

In verde scuro i nuovi *types*, con contorno rosso i *types* modificati. Il nuovo *attribute* è in linea rossa.



## MODIFICHE ALLO STRUCTURAL ANALYSIS DOMAIN

La modifica proposta del *data schema* dello *Structural Analysis Domain* si concentra sull'implementazione di due *abstract entities* e le relative *subentities* di base; la modifica di cinque classi esistenti, andando ad aggiungere valori di enumerazione o, in due casi, implementando due nuove classi di enumerazione collegate.

La prima *abstract supertype* definita è **IfcStructuralDynamicAction**, atta a sostenere azioni dinamiche, come da nome. Spagnuolo e Farello implementano anche una sola sottoclasse per quest'entità astratta: *IfcSpectralAction*, attraverso cui si possono descrivere gli spettri di spostamento relativo, di pseudo-velocità relativa e di pseudo accelerazione assoluta di progetto. Diversi tipi di azioni dinamiche esulano dall'ambito dell'ingegneria sismica, per cui non vengono ora come ora proposte.

Il secondo *abstract supertype* definito è **IfcStructuralDynamicReaction**, che è la base per le classi che descrivono le reazioni risultato di analisi dinamica, ricordando che con “reazioni” IFC include, ad esempio, reazioni vincolari, sollecitazioni interne agli elementi e spostamenti nodali. Infatti Spagnuolo e Farello implementano quattro classi: **IfcStructuralEigenValueReaction**, **IfcStructuralEigenVectorReaction**, **IfcStructuralDampingFactorReaction** e **IfcStructuralParticipationFactorReaction**. Queste come da nome, rispettivamente, servono per descrivere gli autovalori, gli autovettori, i fattori di smorzamento viscoso modale e i fattori di partecipazione modale. Di per sé sono completamente vuote, perché abbiano senso è necessario guardare alle omonime *classes* implementate nello *Structural Load Resource*. Queste infatti “raccolgono” i valori e li indicizzeranno in queste *entities*.

Nell'ottica, poi, di consentire l'uso degli oggetti appartenenti alle classi *IfcStructuralCurveMember* e *IfcStructuralSurfaceMember* all'interno di un software agli elementi finiti, l'azienda ha quindi introdotto due enumerazioni, andando a modificare le due *entities* sopracitate. Questo anche per gestire le classi di azioni dinamiche date dalle masse, classificate nello *Structural Load Resource*. Di conseguenza, rispettivamente, *IfcStructuralCurveMemberFEMFormFunctionEnum* e *IfcStructuralSurfaceMemberFEMFormFunctionEnum* indicano quale funzione di forma viene assegnata ad

uno specifico spostamento di un nodo, a seconda che questo appartenga, rispettivamente, a elemento monodimensionale o bidimensionale.

**-IfcStructuralCurveMemberFEMFormFunctionEnum** è l'*enumerated type* introdotto per consentire l'uso di *IfcStructuralCurveMember* all'interno di un software agli elementi finiti e permette la definizione secondo otto costanti enumerative delle funzioni di forma dei nodi di un elemento monodimensionale. Queste sono:

-*LINEAR*: per descrivere un elemento finito dalle funzioni di forma lineari dei nodi;

-*HYBRID*: per descrivere un elemento finito dalle funzioni di forma miste dei nodi;

-*CUBIC*: per descrivere un elemento finito dalle funzioni di forma cubiche dei nodi;

-*HYBRID\_CUBIC*: per descrivere un elemento finito dalle funzioni di forma miste dei nodi centrali, di cui una sia cubica;

-*QUADRATIC*: per descrivere un elemento finito dalle funzioni di forma quadratiche dei nodi;

-*ISOPARAMETRIC*: per poter descrivere un elemento isoparametrico, in cui, cioè, sia stato necessario un cambio di variabili poiché l'elemento ha una geometria particolare;

-*USERDEFINED*, utilizzabile per descrivere un elemento finito in cui le funzioni di forma siano definite secondo le indicazioni dell'utente;

-*NOTDEFINED*, utilizzata ogniqualvolta non vengano assegnate funzioni di forma ai nodi di un elemento finito.

**-IfcStructuralSurfaceMemberFEMFormFunctionEnum** è l'*enumerated type* introdotto per utilizzare all'interno di un solutore agli elementi finiti, invece, *IfcStructuralSurfaceMember* e definisce sette costanti enumerative per descrivere le funzioni di forma dei nodi di un elemento bidimensionale:

-*ISOPARAMETRIC*: per poter descrivere un elemento isoparametrico, cioè di nuovo l'elemento ha una geometria particolare ed è stato necessario un cambio di coordinate per la risoluzione;

-*SERENDIPITY*: per descrivere un elemento a 8 nodi;

- LAGRANGE*: per descrivere un tipo di elemento a 9 nodi;
- HETEROSYS*: per descrivere un secondo tipo di elemento a 9 nodi;
- LINEAR*: per descrivere un elemento finito dalle funzioni di forma lineari dei nodi;
- USERDEFINED*, enumerazione classica utilizzata per descrivere un elemento dalle funzioni di forma definite dalle indicazioni dell'utente;
- NOTDEFINED*, utilizzata ogniqualvolta non vengano assegnate funzioni di forma ai nodi di un elemento finito.

Poi, tre enumerazioni già presenti, vengono modificate per coprire nuovi tipi di elemento, si tratta di *IfcStructuralCurveMemberTypeEnum*, di *IfcStructuralSurfaceMemberTypeEnum* e di *IfcAnalysisTheoryTypeEnum*. La prima ora presenta la possibilità di definire l'elemento bidimensionale come elemento molla o come elemento rigido attraverso, la seconda presenta, adesso, la possibilità di essere assialsimmetrica e di avere uno sforzo o una deformazione piana generici. La terza include anche alcuni tipi di analisi modale: la teoria modale, la sovrapposizione modale di tipo SRSS, di tipo CQC e di tipo CQC-ASCE, per la catalogazione delle informazioni relative al tipo di analisi dinamiche condotte.

-Le modifiche relative a **IfcStructuralCurveMemberTypeEnum** aggiungono due costanti enumerative:

- SPRING\_MEMBER* per descrivere un elemento molla;
- RIGID\_LINK\_MEMBER* per descrivere un elemento rigido.

-Relativamente a **IfcStructuralSurfaceMemberTypeEnum** sono state aggiunte tre costanti enumerative:

- AXIALSYMMETRIC* per descrivere un elemento assialsimmetrico, ossia ottenuto da una rivoluzione;
- PLANE\_STRESS* per descrivere un elemento bidimensionale atto a sostenere sforzi nel piano;

-*PLANE\_STRAIN* per descrivere un elemento bidimensionale atto a sostenere deformazioni nel piano.

-Invece **IfcAnalysisTheoryTypeEnum** vede aggiungersi quattro costanti enumerative per descrivere le analisi modali nel caso si svolgano analisi dinamiche:

-*MODAL\_THEORY*: per descrivere analisi in cui sia stata applicata la teoria modale;

-*MODAL\_SUPERPOSITION\_SRSS\_THEORY*: per riportare analisi in cui sia stata applicata la sovrapposizione modale dei risultati di tipo SRSS;

-*MODAL\_SUPERPOSITION\_CQC\_THEORY*: per riportare analisi in cui sia stata applicata la sovrapposizione modale dei risultati di tipo CQC;

-*MODAL\_SUPERPOSITION\_CQC\_ASCE\_THEORY*: per riportare analisi in cui sia stata applicata la sovrapposizione modale dei risultati di tipo CQC-ASCE.

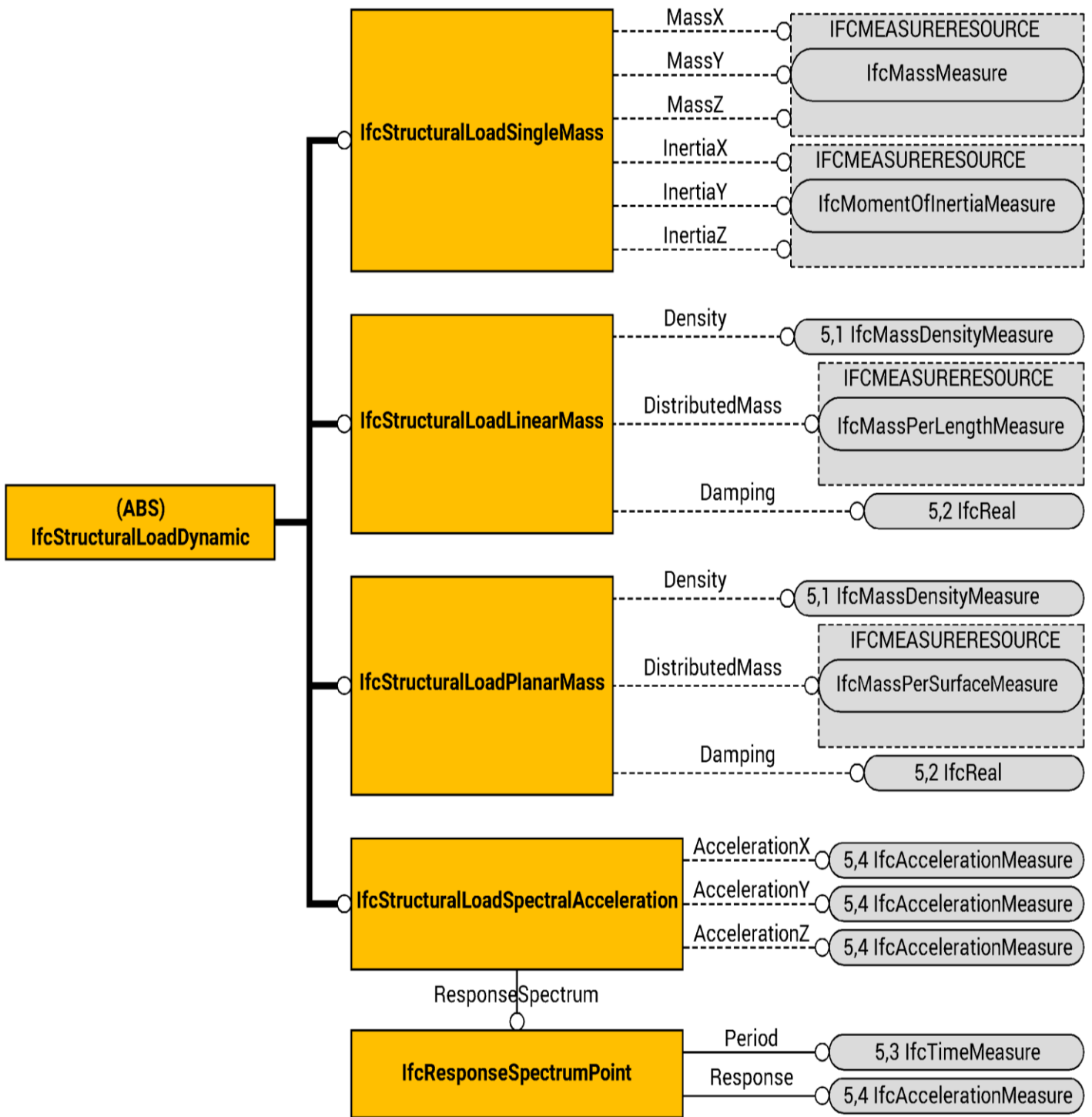


Figura 54 - Proposta Softing srl per l'implementazione azioni dinamiche



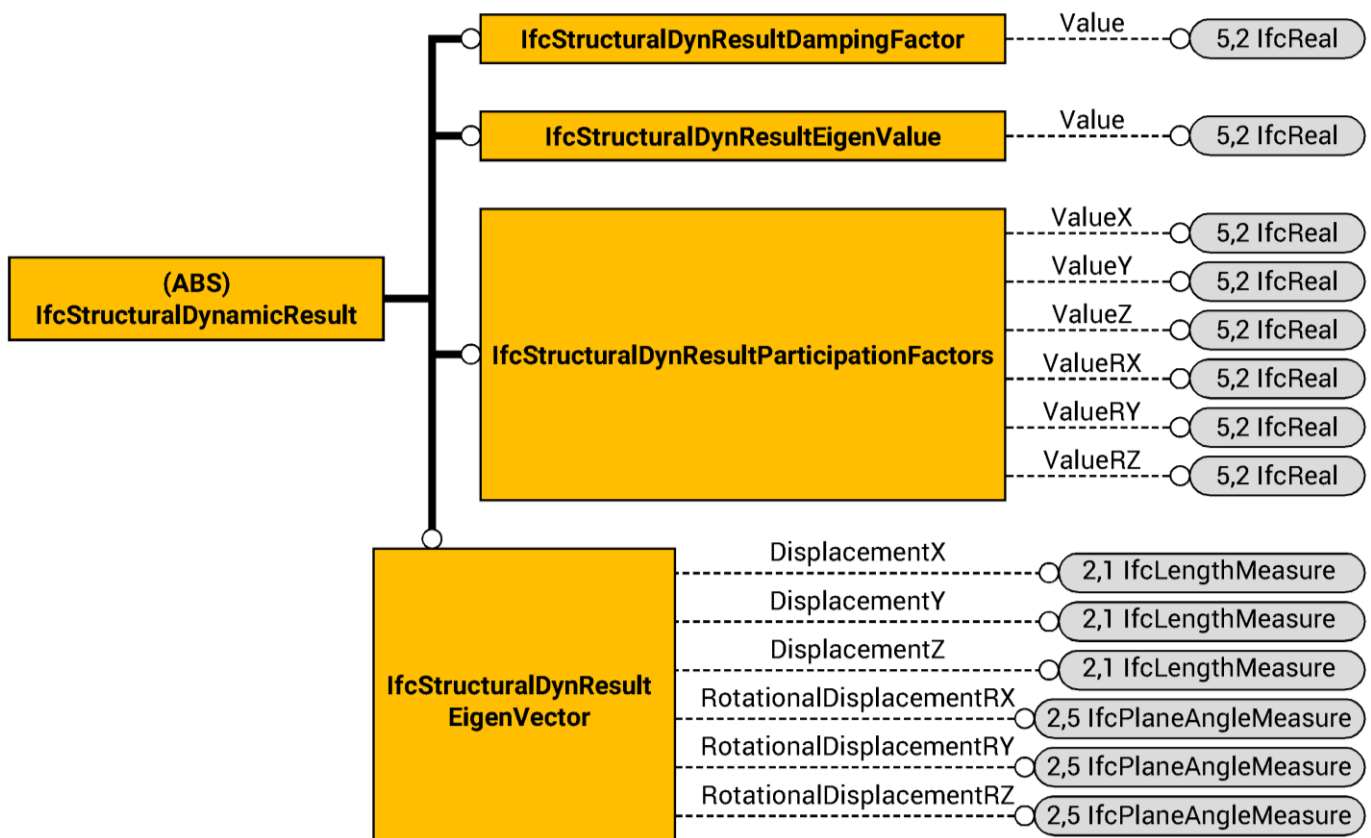


Figura 55 - Proposta Softing srl per l'implementazione dei risultati di dinamica

## MODIFICHE ALLO STRUCTURAL LOAD RESOURCE

Le modifiche proposte per lo *Structural Load Resource* invece riguardano l'aggiunta di due entità astratte e di una entità.

Viene, infatti, implementata da Spagnuolo e Farello l'*abstract entity* **IfcStructuralLoadDynamic** per poter descrivere le azioni dinamiche; da essa derivano *IfcStructuralLoadSingleMass*, *IfcStructuralLoadLinearMass* e *IfcStructuralLoadPlanarMass*, che gestiscono le masse applicate alle entità *IfcStructuralPointConnection*, *IfcStructuralCurveMember* e *IfcStructuralSurfaceMember*. Inoltre, sempre dalla stessa entità astratta, dipende anche *IfcStructuralLoadSpectralAcceleration*, che, con *IfcResponseSpectrumPoint*, implementa uno spettro di risposta e il relativo vettore d'azione sismica.

La seconda *abstract entity* proposta è **IfcStructuralDynamicResult** e anch'essa come la precedente deriva dalla già esistente *IfcStructuralLoadOrResult*. *IfcStructuralDynamicResult* regge i risultati di un'analisi dinamica:

*IfcStructuralDynResultEigenValue*,                      *IfcStructuralDynResultEigenVector*,  
*IfcStructuralDynResultDampingFactor* e *IfcStructuralDynResultParticipationFactor*.

In concreto, sotto *IfcStructuralLoadDynamic*:

-**IfcStructuralLoadSingleMass** presenta massa in x, in y e in z e momento di inerzia in x, y e z attraverso gli *attributes* *MassX*, *MassY* e *MassZ* diretti verso *IfcMassMeasure* e attraverso *InertiaX*, *InertiaY* e *InertiaZ* diretti verso *IfcMomentOfInertiaMeasure*;

-**IfcStructuralLoadLinearMass**, invece, presenta densità, massa distribuita per unità di lunghezza e smorzamento attraverso tre *attributes*: *Density* che riporta *IfcMassDensityMeasure*, *DistributedMass* che riporta *IfcMassPerLengthMeasure* e infine *Damping* che, invece, riporta *IfcReal*;

-**IfcStructuralLoadPlanarMass**, similmente al precedente, densità, massa distribuita per unità di superficie e smorzamento attraverso tre *attributes* identici ai precedenti: *Density*, *DistributedMass* e *Damping*. Come prima, *Density* e *Damping* riportano, rispettivamente *IfcMassDensityMeasure* e *IfcReal*, mentre *DistributedMass*, diversamente da prima, riporta *IfcMassPerSurface*;

-**IfcStructuralLoadSpectralAcceleration**, regge *IfcResponseSpectrumPoint* ed è governato da tre misure di accelerazione, in x, in y e in z. Ciò attraverso quattro *attributes*: *AccelerationX*, *AccelerationY*, *AccelerationZ* rivolti verso *IfcAccelerationMeasure* e *ResponseSpectrum* rivolto verso *IfcResponseSpectrumPoint*;

-**IfcResponseSpectrumPoint** è una *entity* che dà la possibilità di listare i risultati in base all'accelerazione di risposta e al periodo attraverso due *attributes*: *Period* e *Response* che riportano, rispettivamente *IfcTimeMeasure* e *IfcAccelerationMeasure*

Tutti gli *attributes*, ad eccezione di *ResponseSpectrum*, *Period* e *Response* che sono obbligatori, sono opzionali.

Invece, sotto *IfcStructuralDynamicResult*:

-**IfcStructuralDynResultEigenValue** e **IfcStructuralDynResultDampingFactor** reggono un singolo valore reale per descrivere, reciprocamente l'autovalore e il fattore di smorzamento viscoso modale. In entrambi i casi si tratta di un *attribute Value*, non opzionali, diretto verso *IfcReal*;

-**IfcStructuralDynResultEigenVector** regge, come da nome, un vettore a sei componenti, tre spostamenti lungo le direzioni principali e tre rotazioni attorno agli assi principali. Per fare ciò sfrutta sei *attributes*: *DisplacementX*, *DisplacementY*, *DisplacementZ*, *RotationalDisplacementX*, *RotationalDisplacementY* e *RotationalDisplacementZ*. I primi tre sono diretti verso *IfcLengthMeasure*, i secondi tre verso *IfcPlaneAngleMeasure*;

-**IfcStructuralDynResultParticipationFactor** regge anch'esso sei valori indicando il valore di partecipazione modale lungo le tre direzioni principali e attorno ai tre assi principali. Lo fa attraverso sei *attributes*: *ValueX*, *ValueY*, *ValueZ*, *RValueX*, *RValueY* e *RValueZ*, tutti colleganti *IfcReal*.

## VALUTAZIONI MODELLO AGLI ELEMENTI FINITI

La proposta Spagnuolo e Farello, secondo le intenzioni, si inserisce molto efficacemente nell'architettura di IFC andando a definire piuttosto bene ciò che si propone.

Rimane però da rivalutare in alcune sue parti, soprattutto per quanto riguarda il modello agli elementi finiti: se, infatti, risulta molto snella ed efficace la definizione di un elemento finito sulla base della dimensionalità dell'elemento strutturale che descrive, ciò non toglie che possa non essere univoca la cosa. Questo per una distanza intrinseca tra quella che è la schematizzazione in un modello fisico e geometrico e la rappresentazione dello stesso all'interno di un software agli elementi finiti. In altre parole, un elemento trave, ad esempio, deve poter essere descritto non solo come un elemento *beam*, ma anche, a seconda della scelta del progettista o del software, anche come elemento *plate* o come elemento *brick* a seconda delle scelte di modellazione che vengono effettuate. Questo, secondo la proposta Softing, non è al momento possibile, anche implementando elementi strutturali tridimensionali e collegandovi appositi *item* per la definizione di un tridimensionale elemento finito. Altro punto critico è la trasmissione dell'informazione di come, esattamente, un oggetto viene suddiviso nei vari elementi finiti; cioè, come si verrebbe a comportare, ad esempio, un elemento monodimensionale del modello analitico che viene rappresentato in vari elementi finiti, sempre monodimensionali, disposti regolarmente uno dietro all'altro? Ossia, c'è la concreta possibilità che la suddivisione in elementi finiti si rifletta nella suddivisione del modello analitico, cosa altamente non apprezzabile. Con il rischio che si vengano a creare vari elementi monodimensionali nel modello analitico, o quello che non sia possibile mantenere l'informazione di un passaggio così importante della modellazione, è necessario garantire l'integrità di questa informazione, senza dover delegare alle software house questo onere. Infine l'informazione che si viene così a trasmettere rischia di escludere completamente la micromodellazione, ossia la possibilità di modellare con grado di accuratezza e/o di dimensionalità diverso da quello previsto per la schematizzazione analitica.

La motivazione di una scelta di modellazione siffatta è da cercarsi nella volontà di mantenere snello il file IFC, assegnando direttamente al modello analitico anche le proprietà del modello agli elementi finiti. E in effetti, per la maggior parte dei programmi FEM questo è un qualcosa di facilmente

implementabile con minime problematiche. Questo però, oltre alle limitazioni che si vengono a formare, come già detto, pone anche dei notevoli interrogativi riguardo la congruenza nei punti di congiunzione. Come sottolineava la stessa software house, l'onere di mantenere la congruenza viene assegnato al modello analitico, cui si associano le ipotesi del modello agli elementi finiti, e al software stesso, rimanendo, quindi, scongiurato il pericolo, ad esempio, che uno stesso nodo, comune a due elementi, presenti differenti funzioni di forma poiché esso viene modellato correttamente fin dall'inizio. Tuttavia, anche visto che, in ogni caso la responsabilità finale è del progettista e non del produttore di software, sembra apprezzabile la possibilità di poter personalizzare, in ambito IFC le funzioni di forma, al di fuori di un elenco predefinito, che può oltretutto sempre risultare limitante per eventuali sviluppi dei programmi dato che tutto rientrerebbe nella voce *USERDEFINED*. Di più, sebbene di nuovo non sia possibile per le modalità di modellazione, sembra preferibile anche poter controllare e, eventualmente, imporre la congruenza tra i singoli elementi finiti, che, cioè, tra due elementi finiti continui siano condivisi almeno i nodi ai vertici, senza che accada che un nodo di un elemento si ritrovi ad essere disposto lungo il bordo, o su di una faccia, di un altro elemento

Un'importante aggiunta che implementa Softing srl è sulla classificazione di elementi del modello analitico. L'associazione buildingSMART prevedeva un numero molto più limitato di costanti enumerative per descrivere elementi mono e bidimensionali e la software house ne aggiunge altri per descrivere compiutamente il sistema strutturale. Ossia si può, accettando la modifica di Spagnuolo e Farello descrivere un elemento rigido o un elemento a molla, per fare l'esempio nel campo monodimensionale. Questo rimane davvero utile anche per la definizione dello schema strutturale per il progettista.

## VALUTAZIONI ANALISI DINAMICHE

Molto interessante il modo in cui viene gestita, per le conseguenze che porta e per l'impressionante snellezza del tutto, la catalogazione dei risultati di analisi agli elementi finiti: avendo fatto coincidere il modello analitico e il modello agli elementi finiti i risultati di sforzi e deformazioni nei nodi sono ricondotti, implicitamente, a sforzi e deformazioni monodimensionali come classi già implementate nello standard IFC.

Rimane altamente positivo il collegamento, all'interno di già affermati *data schema*, tra il modello analitico e il modello geometrico, la definizione di azioni date dalla massività della struttura

Per quanto riguarda la descrizione di un'azione dinamica la proposta di Spagnuolo e Farello è, probabilmente, molto più coerente e funzionante, risultando persino facilmente integrabile con la prima proposta. Difatti, grandissimo lato positivo della proposta della software house è la facilità di integrazione del mondo delle azioni dinamiche: integrando al modello fisico e geometrico il modello agli elementi finiti, qualsiasi tipo di sollecitazione statica o dinamica risulta molto snellamente implementata nel modello IFC. Questo perché, in parole povere, si vanno a sovrapporre i due modelli con il risultato che le azioni e reazioni descritte per il primo modello risultano compiutamente definite per il secondo e viceversa. Questo, però è purtroppo ampiamente limitante per il mondo delle analisi agli elementi finiti per le considerazioni fatte prima.

Rimane, in ogni caso, estremamente valida la proposta dei due programmatori italiani perché completa la descrizione del sistema strutturale, riferendosi ad azioni tipologicamente valide per analisi dinamiche. Ad esempio il fatto che siano descritte le masse più o meno concentrate. Con delle piccole aggiunte e considerazioni si può benissimo migliorare lo schema di Spagnuolo e Farello perché siano descritte tutte le possibili azioni per analisi dinamiche o statiche equivalenti, sull'esempio di quanto avanzato dal gruppo ST-7. Di più, da valutare in maniera assolutamente positiva è il fatto che, così facendo, è già possibile considerare una condizione di carico che, oltre all'azione sismica opportuna, consideri anche le combinazioni di carichi statici necessari, come, ad esempio, i carichi portati. In questo modo, a differenza della proposta Egizio-Giapponese è possibile, come necessario, trasmettere

anche questo tipo d'informazione dato che, nell'altro caso, non erano ancora classificabili le combinazioni di carico.

Per quanto riguarda il sistema di archiviazione di risultati di analisi in ambito dinamico, esso è coerente e corretto: sarà semplicemente necessario aggiungere alcuni *attributes* per potersi conformare allo standard informativo proposto da NAFEMS a riguardo.

Guardando, poi le modifiche agli *enumerated types* va riferito come anche queste non siano affatto da scartare. Quelle relative alla classificazione tipologica degli elementi strutturali mono e bidimensionali si vorranno mantenere, perché possono essere utili a descrivere più compiutamente un qualsiasi sistema strutturale. Mentre ciò che è veramente interessante tra queste è la modifica ad *IfcAnalysisTheoryTypeEnum* in cui vengono aggiunte delle enumerazioni per descrivere la tipologia di analisi vengono condotte; le si vorrà anche ulteriormente ampliare, avendo sempre una prospettiva rivolta al futuro: difatti saranno utili a descrivere anche tipologie di analisi strutturali diverse, non ancora previste e, soprattutto, al di fuori di analisi agli elementi finiti.

L'unica vera critica apportabile a questa parte di proposta riguarda la sola implementazione di alcuni *attributes*, quelli relativi al *Damping*, allo smorzamento. Come Serror et al. viene utilizzato un *simple data type* sostanzialmente corretto in EXPRESS, ma che buildingSMART preferisce sostituire da un dedicato *defined type* per la specifica tipologia di informazione in modo da rendere il tutto più chiaro. Di per sé è cosa da poco e la si risolve in poco tempo.

# **7. una nuova proposta**



Dopo aver valutato le proposte avanzate da varie personalità nel corso degli anni si vuole sintetizzare una nuova proposta. Indubbi vantaggi sono presenti in ciò che dai gruppi di ricerca di ST-7 e Di Spagnuolo e Farello è emerso, si vuole quindi cercare anche di unire ciò che di positivo presentano, senza ripartire da zero.

In breve, si è voluta mantenere, per la flessibilità e la completezza, la struttura di identificazione topologica di un elemento finito, circa come presentata da Serror e il suo gruppo di ricerca. Indubbiamente efficace è infatti il classificare un elemento finito con una struttura che ricalca quella del modello analitico, in cui una *superentity* chiamata *IfcFiniteElementModelItem* descriva l'intera tipologia possibile di un modello agli elementi finiti raccogliendo sotto di sé i vari oggetti, siano essi nodi descritti con *IfcFEMNode*, siano essi elementi descritti con *IfcFEMElement* oppure punti di integrazione descritti con *IfcFEMIntegrationPoint*. Tutto questo poi raccolto sotto *IfcFiniteElementModel*.

Per quanto riguarda la classificazione di analisi dinamiche si cerca di unire le due proposte: la parte implementata da Spagnuolo e Farello si presenta più completa per quanto riguarda la classificazione di azioni e reazioni prettamente sismiche, dagli spettri in accelerazione agli autovalori e autovettori; mentre la parte dell'altro gruppo si presenta di più ampio respiro, cercando di descrivere tutto il mondo delle analisi dinamiche, interamente assente in ogni caso. Si vorrà quindi, dopo alcune piccole sistemazioni, provvedere ad iniziare ad ampliare la proposta Softing, descrivendo le azioni per altri tipi di analisi dinamiche. Purtroppo, vista l'enormità del problema delle analisi dinamiche in generale, non si è potuto spingersi oltre, ad esempio considerando non linearità

La maggior parte del lavoro svolto si è concentrata nel rendere organici questi "frammenti" tra di loro, aggiungendo una serie di informazioni mancanti, sia in ambito di analisi agli elementi finiti che dinamiche, che sono emerse dall'analisi in letteratura. Soprattutto per quanto riguarda la prima parte si è dovuto modificare anche dei *data schema* al di fuori dello *Structural Analysis Domain* e dello *Structural Load Resource*, soprattutto all'interno del *Measure Resource*. Infine, dato che in ogni caso la proposta egizio-giapponese è oramai datata e implementata per la versione 2x di IFC, si è dovuto provvedere a

sistamarla e renderla coerente alla struttura attuale dello standard, arrivato proprio quest'anno alla versione 4, appendice 2.

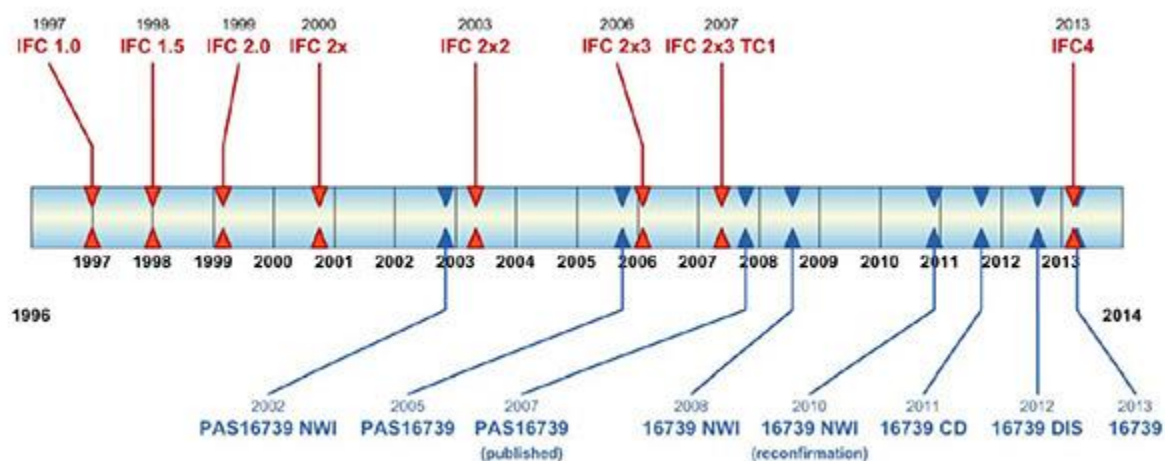


Figura 56 - Sviluppo nel tempo di IFC e differenti versioni.

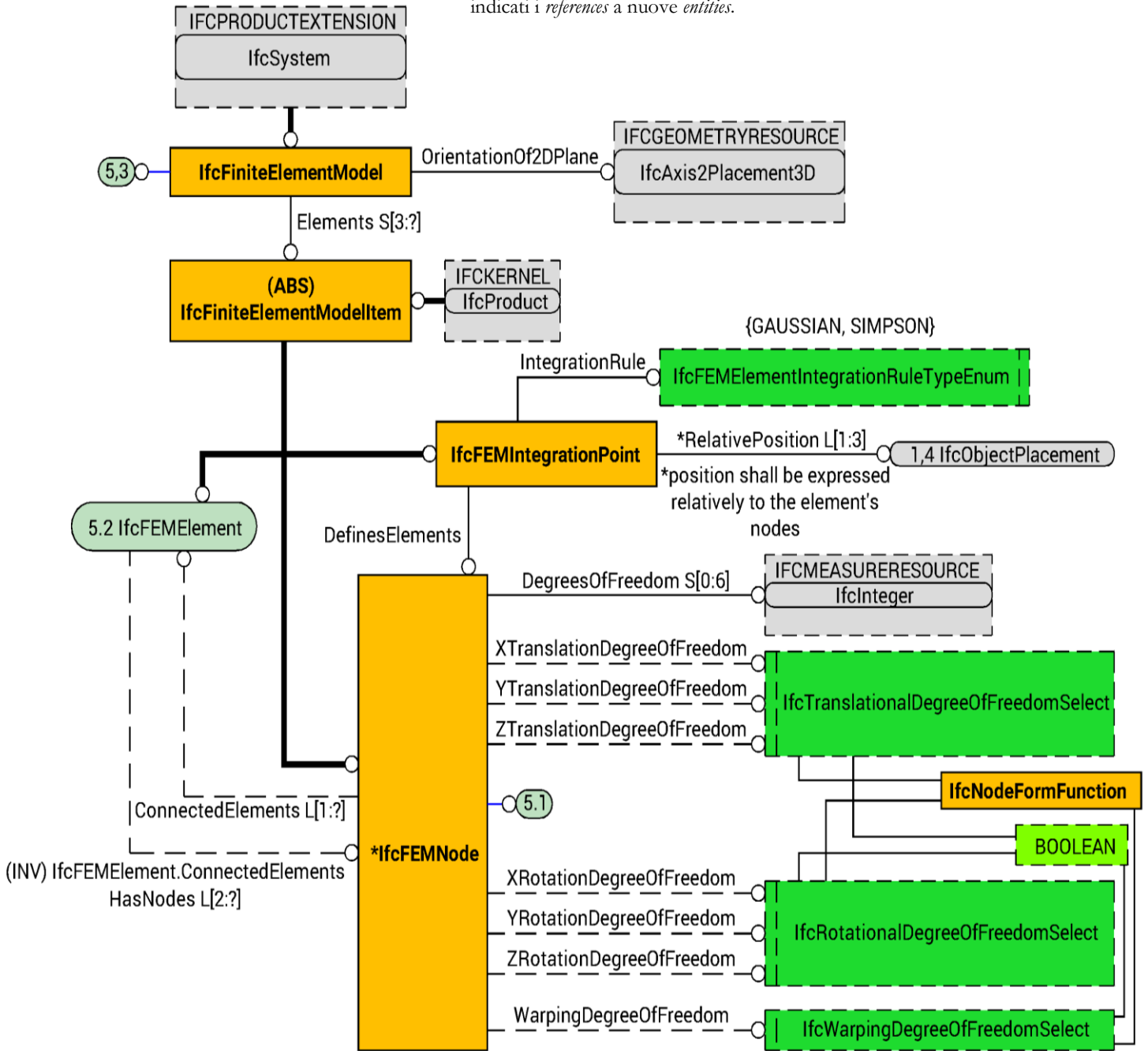


# **7A. elementi finiti**

Per quanto riguarda la struttura della parte relativa al modello agli elementi finiti, come già detto, la si vuole mantenere simile alla proposta ST-7, ma andando a portare alcune modifiche che vogliono essere miglioramenti, anche guardando alla proposta Softing srl. Questo perché presenta due grandi vantaggi rispetto alla proposta di Spagnuolo e Farello, dovuto al fatto di mantenere separati i due modelli, quello analitico e quello agli elementi finiti. Il primo riguarda la possibilità di mantenere collegate ad uno stesso modello analitico differenti mesh agli elementi finiti, senza necessità per questo di produrre più modelli analitici, come porta invece a fare lo schema di Softing srl. Si parla quasi solamente di mesh con elementi finiti dalle dimensioni differenti tra di loro; in pratica scorporando i due modelli, si possono riportare correttamente differenti mesh di elementi finiti, tutte corrispondenti ad uno stesso elemento analitico. Da sottolineare che con questo schema venga riportata esattamente la stessa mesh utilizzata, non una approssimativamente simile. Nel caso, ad esempio, in cui si ipotizzino i giunti di rottura di un maschio murario, sarà necessario riportare differenti modelli analitici ciascuno con il giunto di rottura ipotizzato e con il relativo modello agli elementi finiti. Poi i vari modelli analitici saranno collegati all'elemento architettonico cui fanno riferimento. Infine, scorporando i due modelli si possono anche modellare agli elementi finiti oggetti architettonici cui non ci si sia ricondotti ad un modello analitico.

Il secondo grande vantaggio risiede nel fatto che deve essere possibile confrontare con IFC anche solutori ad elementi finiti che siano del tipo a scatola bianca. La maggior parte dei software sono del tipo a scatola nera, in cui il procedimento di risoluzione non è esplicitato ma è garantito dal produttore. Poiché, in ogni caso, la responsabilità finale rimane al progettista e non al produttore, si stanno iniziando a diffondere, per pochi irriducibili appassionati, anche solutori a scatola bianca, come FEniCS Project, in cui è chi lavora con il software a confrontarsi con la produzione di modelli ad elementi finiti e con la risoluzione di equazioni differenziali a coefficienti parziali. Con la modellizzazione proposta si vuole far sì che entrambi i tipi di solutori possano confrontarsi con IFC. La proposta di Spagnuolo e Farello è ritornata utile per poter far interfacciare i solutori “tradizionali”, riportando le informazioni utili. Rimane facoltativa la possibilità di riportare molte informazioni che per un solutore a scatola bianca sono fondamentali, come l'espressione matematica della funzione di forma, ma che per i normali risolutori sono molto più che superflue.

Figura 57 - Proposta modello elementi finiti. Nodi, punti di integrazione e struttura. In arancione sono indicate le nuove *entities*, rispetto al giallo originale, in verde più scuro i nuovi *types* rispetto al verde chiaro, in celestino sono indicati i *references* a nuove *entities*.



\*when two or more nodes belong to the same IfcFEMElement, they shall have the same value for DegreesOfFreedom.

\*if the Boolean variable is set on TRUE, this means that the node has the degree of freedom activated, but it's not described the precise mathematical form function of the node.

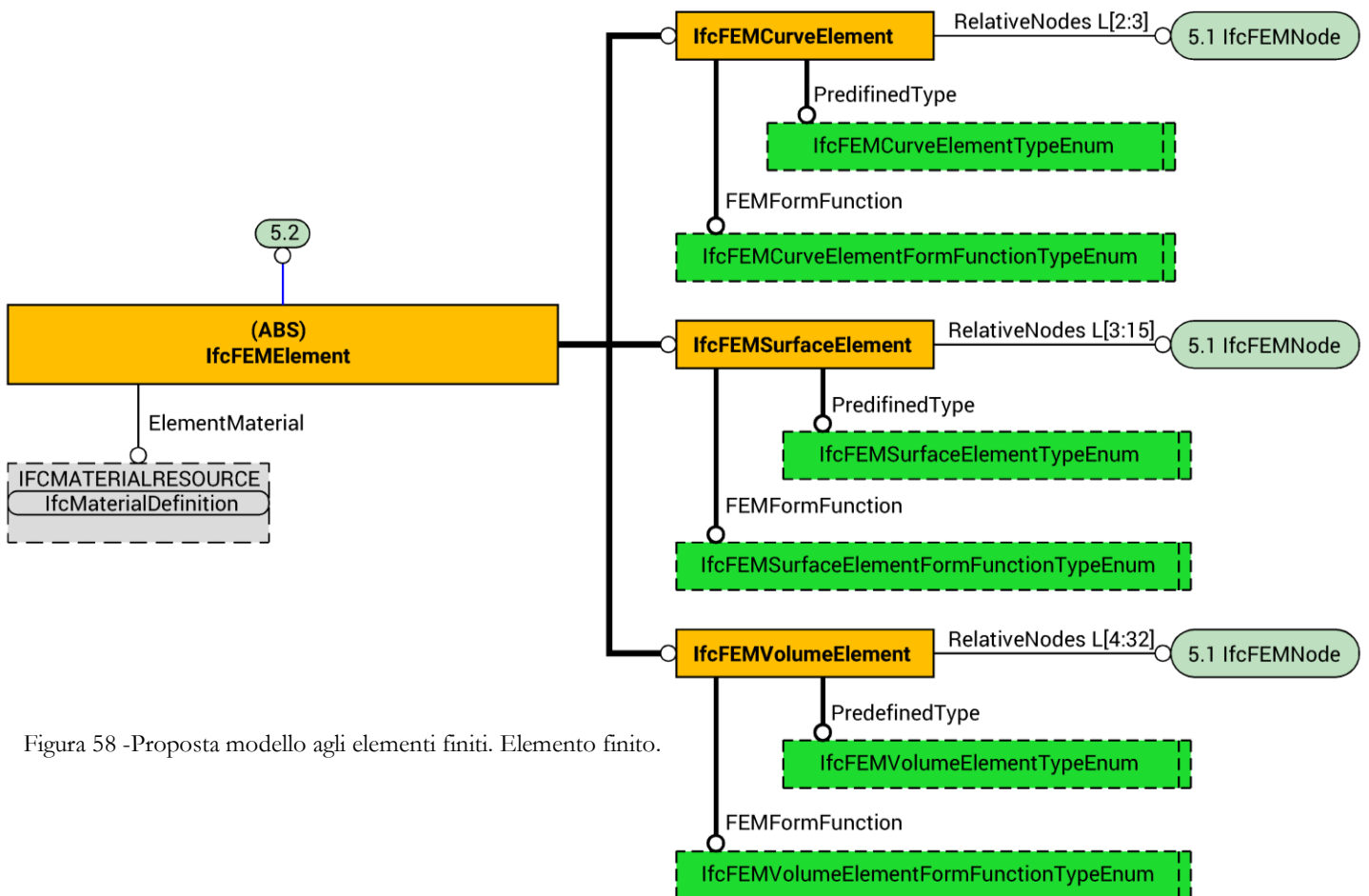


Figura 58 -Proposta modello agli elementi finiti. Elemento finito.

Nuovamente, si definisce un' *abstract entity* che riunisca sotto sé gli oggetti che costituiscono il modello vero e proprio: sotto **IfcFiniteElementModelItem** trovano spazio *IfcFEMElement*, *IfcFEMNode* e *IfcFEMIntegrationPoint*. Tutto sommato sono presenti grosse differenze con la proposta originaria:

Sono specificati gli *attributes* per gli specifici oggetti, non per la classe superiore, per aggiungere solamente le informazioni necessarie e coerenti.

Viene rinnovata la definizione di un elemento finito, se rimane invariato il fatto che l' *entity* **IfcFEMElement** sia del tipo *abstract* in modo che venga ulteriormente definita da **IfcFEMCurveElement**, **IfcFEMSurfaceElement** e **IfcFEMVolumeElement**, cambia, invece la

definizione dei numerosi *attributes* che gli erano propri: vengono tutti rimossi. Gli *enumerated types* non vengono dimenticati ma vengono riportati in sede adeguata, seppur rimaneggiati. Questi infatti sebbene riferiti all'elemento finito erano e rimangono proprietà di nodi e punti di integrazione, a loro saranno riferiti. Vi è un solo *attribute* che ora si riferisce ad *IfcFEMElement* è *ElementMaterial*, che riporta un *item* di *IfcMaterialDefinition* che descrive la caratterizzazione meccanica dei materiali strutturali assegnati all'elemento finito. Gli altri due *attributes* sono *SharedNodes* e il suo *inverse attribute* *HasNode*, che specificano i nodi condivisi con altri elementi e quali siano i nodi dell'elemento stesso collegandosi, appunto, a *IfcFEMNode*.

Passando alle *subclasses* **IfcFEMCurveElement**, **IfcFEMSurfaceElement** ed **IfcFEMVolumeElement**, viene aggiunto loro una *relationship* che vi listi gli *item* di *IfcFEMNode*, cioè i nodi relativi a ciascun elemento finito. È necessario che sia una lista ordinata poiché i nodi sono numerati e in una posizione bene precisa, inoltre, poiché hanno determinate possibilità di aggregazione avranno un limite minimo e un limite massimo ciascuna. Ulteriori sviluppi delle tecnologie FEM dovranno confrontarsi con l'estensione di IFC per quanto riguarda soprattutto il numero massimo di nodi riportabili ad uno stesso elemento. Ogni *subentity* presenta l'*attribute* *FEMFormFunction* che riporta il tipo di elemento finito descritto, quindi il relativo tipo di funzione di forma. A elementi monodimensionali si collega **IfcFEMCurveElementFormFunctionEnum** un *enumerated type* introdotto, sul modello di quanto prodotto da Softing, per riportare la tipologia di elemento monodimensionale utilizzata e le rispettive funzioni di forma dei nodi. Le enumerazioni che permettono ciò sono le seguenti otto: *LINEAR*, *HYBRID*, *CUBIC-HYBRID\_CUBIC*, *QUADRATIC*, *ISOPARAMETRIC*, *USERDEFINED*, *NOTDEFINED*. A elementi bidimensionali si collega **IfcFEMSurfaceElementFormFunctionEnum**, l'*enumerated type* introdotto per riportare la tipologia di elemento bidimensionale utilizzata e le rispettive funzioni di forma dei nodi. Le enumerazioni che permettono ciò sono le seguenti otto: *ISOPARAMETRIC*, *SERENDIPITY*, *LAGRANGE*, *HETEROSIS*, *LINEAR*, *USERDEFINED*, *NOTDEFINED*. A elementi tridimensionali si collega **IfcFEMVolumeElementFormFunctionEnum**, l'*enumerated type* introdotto per riportare la tipologia di elemento tridimensionale utilizzata e le rispettive funzioni di forma dei nodi. Le enumerazioni che permettono ciò sono le seguenti otto: *LINEAR*,



*HEDGE\_QUADRATIC*, *HEDGE\_CUBIC*, *CUBIC-HERMITE*, *CUBIC-HERMITE\_SERENDIPITY*, *BICUBIC-BERNSTEIN*, *BICUBIC-BERNSTEIN\_SERENDIPITY*, *ISOPARAMETRIC*, *USERDEFINED*, e *NOTDEFINED*.

Per la completa definizione degli elementi, anche dei gradi di libertà possibili, si collega anche un *attribute* a ciascun elemento finito che riporta a un secondo *enumerated type*. Si tratta di *IfcFEMCurveElementTypeEnum*, di *IfcFEMSurfaceElementTypeEnum* e *IfcFEMVolumeElementTypeEnum* collegati ciascuno alla rispettiva *entity* da *PredefinedType*. Sono tutte ricalcate sul corrispettivo *enumerated type* che descrive il tipo di elemento nel modello analitico, recependo le stesse modifiche proposte da Softing per questi ultimi. Una regola, per ciascuno, esplicita quindi che debbano restituire la medesima enumerazione se i due modelli sono messi in collegamento e gli *item* degli elementi finiti descrivono un *item* del modello analitico con la stessa dimensionalità. La prima, **IfcFEMCurveElementTypeEnum**, presenta la possibilità di definire l'elemento bidimensionale come elemento molla, come elemento rigido, come cavo, come membro sottoposto soprattutto a tensione o a trazione, come smorzatore, ad esempio. Le enumerazioni sono: *RIGID\_JOINED\_MEMBER*, *PIN\_JOINED\_MEMBER*, *CABLE*, *TENSION\_MEMBER*, *COMPRESSION\_MEMBER*, *SPRING\_MEMBER*, *RIGID\_LINK\_MEMBER*, *DAMPER*, *USERDEFINED* e *NOTDEFINED*.

Attraverso la seconda, **IfcFEMSurfaceElementTypeEnum**, si può descrivere un elemento bidimensionale e il suo comportamento nel modello agli elementi finiti, ad esempio come uno *shell*, come una membrana a rigidezza flessionale, come un elemento assialsimmetrico, come un elemento che sopporta sforzo o una deformazione piana generici, attraverso le enumerazioni: *BENDING\_ELEMENT*, *MEMBRANE\_ELEMENT*, *SHELL*, *AXIALSYMMETRIC*, *PLANE\_STRESS*, *PLANE\_STRAIN*, *USERDEFINED*, *NOTDEFINED*.

Infine, attraverso **IfcFEMVolumeElementTypeEnum**, si descrive un elemento tridimensionale e la sua tipologia, se solida, assialsimmetrica o come shell solido, attraverso le enumerazioni seguenti: *BRICK*, *AXISSYMMETRICAL*, *SOLID\_SHELL*, *USERDEFINED*, *NOTDEFINED*.

Con questo costrutto si includono, quindi, anche elementi speciali, come le molle, i link rigidi e gli smorzatori.

Viene rinnovata anche la definizione di un nodo attraverso **IfcFEMNode**: rimane uguale l'applicazione delle condizioni esterne attraverso l'omonimo *attribute AppliedCondition* verso *IfcBoundaryCondition*; per il resto si introducono tre *select type* per la definizione dei gradi di libertà, a cui si riferiscono sette *attributes*, un ottavo *attribute* riporta il numero dei gradi di libertà del nodo come generica informazione. Quest'ultimo è l'ultimo *attribute* non opzionale, poiché tutti riportano informazioni superflue per i normali software commerciali. Sono davvero utili per solutori a scatola bianca e per chi voglia effettivamente, per amore di completezza, riportare queste informazioni. Il primo *select type* è **IfcTranslationalDegreeOfFreedom** che permette la scelta con tra una variabile Booleana e una nuova *entity* per poter descrivere la possibilità di traslare del nodo; nel caso in cui il grado di libertà non sia attivato la variabile Booleana restituisce la voce *FALSE*, altrimenti si sceglie un *item* di *IfcNodeFormFunction* per esplicitare effettivamente la funzione di forma del nodo nel caso in cui lo si voglia fare. Si può, altresì, non riportare la funzione di forma, ma solamente l'attivazione del grado di libertà semplicemente scegliendo la variabile Booleana *TRUE* e lasciando, così, il nodo privo della definizione matematica della funzione di forma; una regola esprime questo funzionamento. Il secondo e il terzo *select type* sono **IfcRotationalDegreeOfFreedom** e **IfcWarpingDegreeOfFreedom** e sono in tutto simili al primo, per poter descrivere la possibilità di ruotare e ingobbarsi del nodo. Al primo *type* si riferiscono tre *attributes*, cioè *XTranslationalDegreeOfFreedom*, *YTranslationalDegreeOfFreedom* e *ZTranslationalDegreeOfFreedom*, mentre al secondo *type* si riferiscono i tre *attributes* *XRotationalDegreeOfFreedom*, *YRotationalDegreeOfFreedom* e *ZRotationalDegreeOfFreedom*; l'ultimo *attribute* che si riferisce all'ultimo che si riferisce all'ultimo *type* presentato è *WarpingDegreeOfFreedom*. Un'ulteriore regola stabilisce poi che se due o più nodi appartengono allo stesso elemento finito è necessario che la definizione dei gradi di libertà debba essere la stessa, difatti l'ultimo *attribute*, *ConnectedElement*, si collega ad *IfcFEMElement* e sistema in una lista almeno un elemento che il nodo collega, se effettivamente è così.

Per la completa definizione di un modello agli elementi finiti si è poi ritenuto necessario poter descrivere all'interno delle *Industry Foundation Classes* gli eventuali punti di integrazione utilizzati per la

risoluzione<sup>65</sup>. Per poter fare ciò si è ripreso il modello proposto nel 2007: un'entity **IfcFEMIntegrationPoint** con tre *attributes* per poter descrivere a che elemento faccia riferimento il punto considerato, quindi *DefinesElement* si collega ad *IfcFEMElement*; in che posizione si trovi il punto, *RelativePosition* fa riferimento ad *IfcObjectPlacement* e la regola *PositionToNodes* esplicita che ci si deve riferire ai nodi e utilizzare le coordinate locali per poter esprimere il posizionamento del punto di integrazione; l'ultimo *attribute* è *IntegrationRule* e serve a specificare quale sia la regola di integrazione che viene utilizzata nello stesso punto, attraverso l'*enumerated type* **IfcFEMElementIntegrationRuleTypeEnum**. Questo permette di scegliere tra i due metodi previsti, cioè di Gauss o di Simpson, oppure eventuali ulteriori metodi specificati dall'utente o, addirittura, lasciandolo, improbabilmente indefinito; le enumerazioni presenti sono *GAUSSIAN*, *SIMPSON*, *USERDEFINED* oppure *NOTDEFINED*. L'unica modifica sostanziale introdotta è la sostituzione di *IfcReal* con *IfcObjectPlacement* per poter descrivere il posizionamento del punto di integrazione. Il vantaggio di questa costruzione è la facoltatività di introdurre questo tipo di informazioni. Il modello agli elementi finiti è completo e definito anche senza quest'informazione, ma, se si volesse riportarla lo si potrebbe fare. Sia per chi effettivamente si produce la risoluzione matematica, sia per chi voglia riportare completamente il processo risolutivo.

Viene poi aggiunta un'ultima *superentity*: si tratta di **IfcFiniteElementModel**. Questa viene utilizzata per raccogliere tutti gli elementi di un modello agli elementi finiti attraverso l'*attribute Element* che, in un *set*, li cataloga; viene ovviamente corretto rispetto alla proposta di Serror la cardinalità di questo, poiché è impossibile che ci sia solamente un oggetto, visto che ce ne debbono essere almeno tre, l'elemento monodimensionale e i due nodi che lo racchiudono.

---

65 - Enzo Tonti, "Introduzione Elementare agli Elementi Finiti", pag. 63 ss.

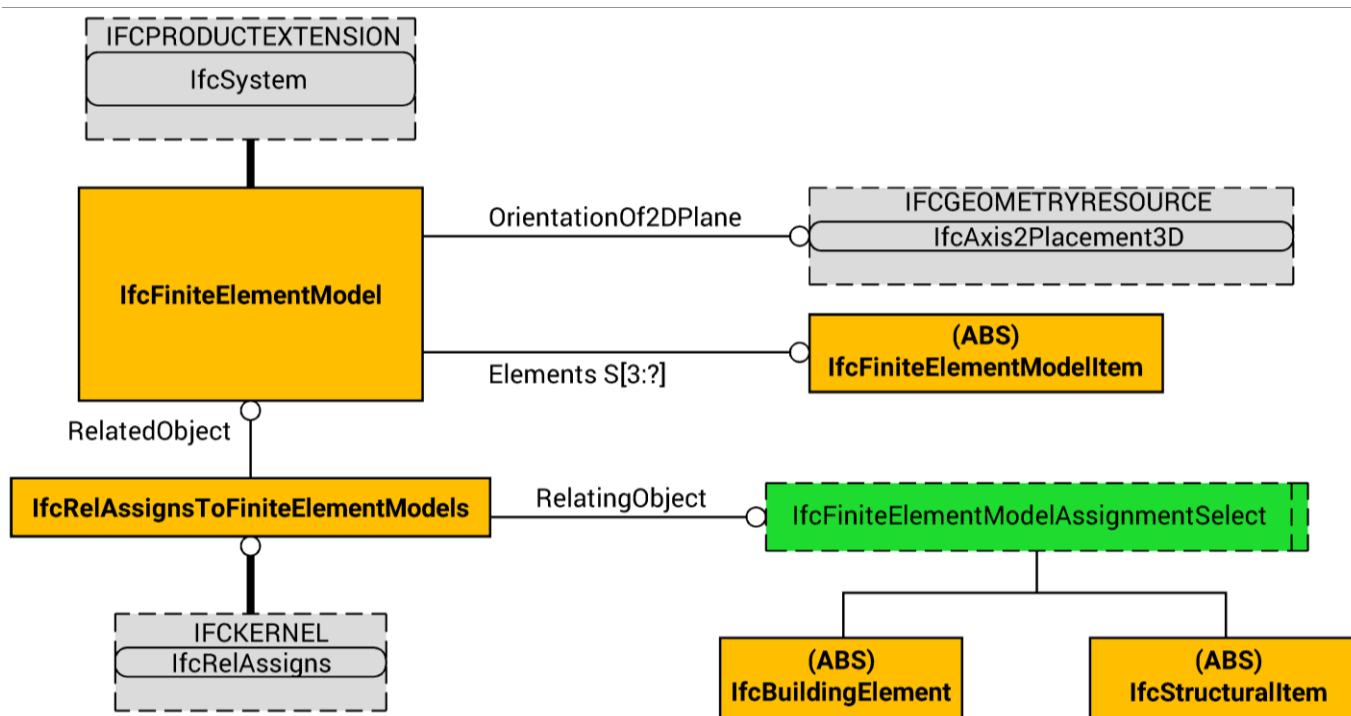


Figura 59 - Proposta modello agli elementi finiti. Relazionalità di questo.

Tutte gli oggetti presentati con colore sono proposti e non attualmente esistenti.

## COLLEGAMENTO CON IL MODELLO AGLI ELEMENTI FINITI

Viene inoltre implementata una classe, sempre sull'esempio proposto da Serror, per poter collegare il modello meccanico agli elementi finiti al modello analitico o, comunque, agli elementi dell'edificio per cui si vuole condurre un'analisi. Per fare ciò si sfruttano due *attributes* e un *select type*.

Viene così data *IfcRelAssignsToFiniteElementModels* che si riferisce al *Kernel Data Schema* del nucleo; con due *attributes*, *RelatedObjects* e *RelatingObjects*, vengono collegati *IfcFiniteElementModel*, che raccoglie tutti gli oggetti di un modello agli elementi finiti, e *IfcFiniteElementModelAssignmentSelect*, che permette di scegliere una *entities* tra *IfcBuildingElement* e *IfcStructuralItem*, per poter descrivere gli oggetti che sono stati modellati agli elementi finiti.

È da valutare la possibilità che il *select type* possa riferirsi anche ad altri oggetti, sul modello di ciò che veniva proposto già nel 2007 dal gruppo di ricerca egizio-giapponese, come, ad esempio *IfcCivilElement*; nel frattempo rimane un qualcosa di più che esula dal nostro compito.

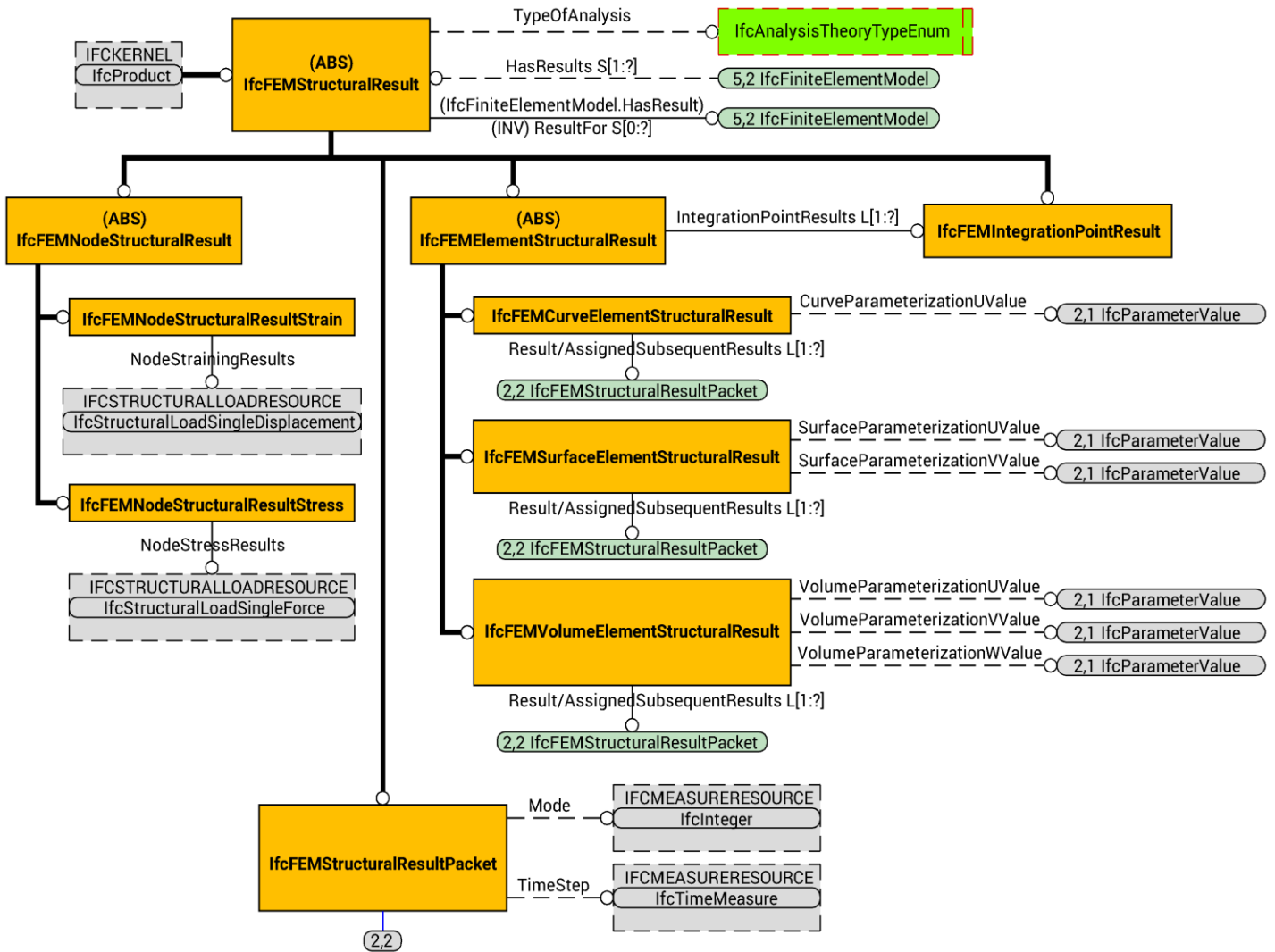


Figura 60 - Proposta modello elementi finiti. Risultati modello. In arancione sono indicate le nuove *entities*, rispetto al giallo originale, in verde più scuro i nuovi *types* rispetto al verde chiaro, in celestino sono indicati i *references* a nuove *entities*.

## RISULTATI DELLE ANALISI AGLI ELEMENTI FINITI

Una classe che necessitava di una forte revisione era quella relativa ai risultati di analisi agli elementi finiti: da un lato Softing non ne proponeva alcuna, poiché facendo coincidere i modelli analitico e agli elementi finiti, anche i risultati dei due modelli vengono a coincidere; dall'altra parte Serror e il suo

gruppo di ricerca ne proponevano una che, per quanto valida, era molto macchinosa e implementava *entities* nuove che avrebbero potute essere evitate riferendosi ad altre già esistenti nel modello analitico.

Si è voluto mantenere la struttura proposta dal gruppo ST-7 in cui un' *abstract superentity* raccogliesse tutti i tipi di analisi agli elementi finiti: **IfcFEMStructuralResult**. Questa, attraverso un *attribute* opzionale, *HasResult*, e il suo inverso riferisce a quale modello agli elementi finiti si riferiscono i vari risultati collezionati e, attraverso *TypeOfAnalysis* riferito a *IfcAnalysisTheoryTypeEnum*, riporta che tipo di analisi sono state condotte. La classe è stata modificata per descrivere anche analisi dinamiche. Questo verrà affrontato all'interno del capitolo 7C.

In ogni caso *IfcFEMStructuralResult* presenta tre *subentities* che riportano i risultati dei nodi, i risultati relativi agli elementi finiti e i risultati relativi ai punti di integrazione.

-**IfcFEMNodeStructuralResult** raccoglie i primi attraverso due *subentities* a sua volta: **IfcFEMNodeStructuralResultStrain** e **IfcFEMNodeStructuralResultStress** che descrivono rispettivamente le deformazioni e gli sforzi del nodo, per fare ciò si riferiscono ad *IfcStructuralLoadSingleDisplacement* e ad *IfcStructuralLoadSingleForce* attraverso gli *attributes* *NodeStrainingResults* e *NodeStressResults*. Si spera che una siffatta modellizzazione dei dati possa essere efficace quanto, se non di più di quella avanzata da Serror e il suo gruppo, ma con una maggior snellezza e coerenza, inserendosi al meglio nell'esistente standard IFC.

-**IfcFEMElementStructuralResult** raccoglie invece i risultati, come nella proposta del 2007, in tre *subentities*: **IfcFEMCurveElementStructuralResult**, **IfcFEMSurfaceElementStructuralResult** ed **IfcFEMVolumeElementStructuralResult** per poter descrivere, rispettivamente i risultati di elementi mono, bi e tridimensionali. A ogni *subentities* sono riportati, esattamente come nella proposta originaria, il cambio di variabili applicate agli elementi. Sono presenti tanti *attributes* per ogni dimensione dell'elemento per cui sia stato effettuato il cambio di variabile. Questi sono collegati ad *IfcParameterValue*, che riporta il valore del parametro attraverso *CurveParameterizationUValue*, per esempio nel caso dell'elemento monodimensionale. Poi ciascuno elemento riporta il risultato dell'elemento stesso: si collega a tutte e tre le *subentities* uno stesso *attribute Result/AssignedSubsequentResult* che, collegato a un *item* di *IfcFEMElement* vuole collegare il risultato delle

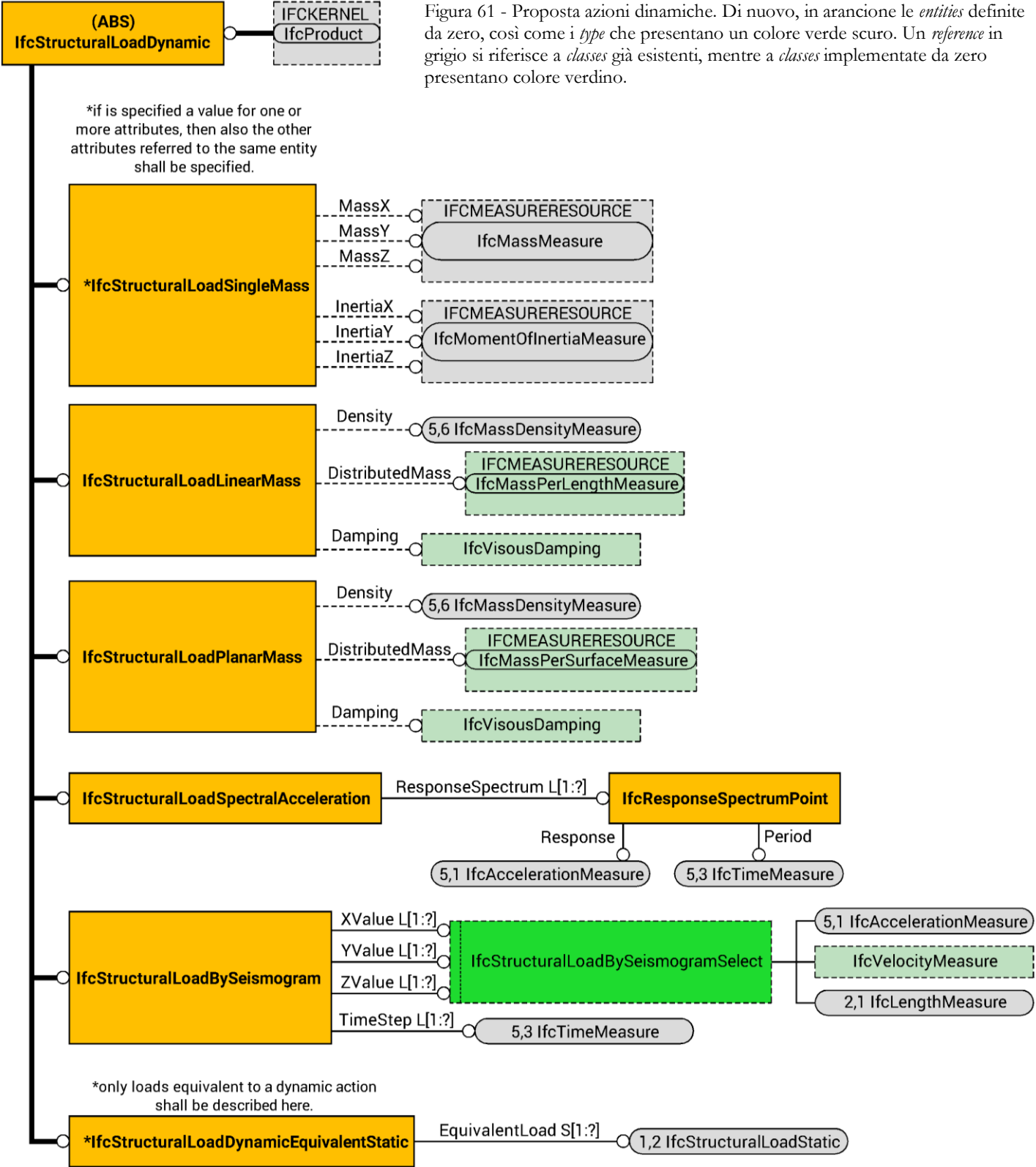
analisi condotte contenute il **IfcFEMStructuralResultPacket**. Questi *attributes* avranno una cardinalità almeno di uno, sulla base della quantità di risultati di analisi che sono state condotte. Infine, un *attribute* verso *IfcFEMElementStructuralResult* lista i vari risultati dei punti di integrazione, esplicitati attraverso **IfcFEMIntegrationPointResult**.





# **7B. analisi dinamiche**

Figura 61 - Proposta azioni dinamiche. Di nuovo, in arancione le *entities* definite da zero, così come i *type* che presentano un colore verde scuro. Un *reference* in grigio si riferisce a *classes* già esistenti, mentre a *classes* implementate da zero presentano colore verdino.



## AZIONI PER ANALISI DINAMICHE

Per quanto riguarda la descrizione di un'azione dinamica e il conseguente mondo delle analisi sismiche, e dinamiche in generale, l'implementazione risulta essere meno ostica. Si tratta infatti di implementare delle informazioni, in maniera organica, in un modello già presente. Il modello Softing aveva l'estremo vantaggio della snellezza in questi termini: collegando il modello agli elementi finiti al modello geometrico e fisico le azioni implementate per quest'ultimo si ritrovavano direttamente sul primo, senza bisogno di ulteriori classi, se non di alcune puntualizzazioni, perché ciò potesse funzionare. Il problema che, così, si viene a porre lo si vuole risolvere mettendo in relazione il modello agli Elementi Finiti con il modello analitico, già descritto all'interno dello standard IFC. In modo che le attività, che secondo il linguaggio IFC comprendono azioni e reazioni, agenti sul modello analitico le si ritrovino nel modello agli Elementi Finiti, senza bisogno di ulteriori specificazioni come, piuttosto cervelloticamente, proponeva il gruppo di ricerca egizio-giapponese. Qui, infatti, per omogeneizzare e collegare il tutto si creava una classe di relazioni, valida solamente per il modello per analisi agli elementi finiti, che serviva a unire il modello agli elementi finiti con le azioni che su di esso insistevano. Si distinguevano così due classi di modello, uno analitico e uno agli elementi finiti, e le due classi di attività relative ad essi.

Rimane, in conclusione, molto valido lo schema proposto dalla software house italiana, in cui si riferisce al modello analitico sia il modello agli elementi finiti, sia le classi di attività dinamiche; per cui lo si vuole prendere a modello, aggiungendo classi atte a descrivere ogni possibile analisi conducibile con finalità sismiche o dinamiche, in generale.

La parte di lavoro più consistente riguarda la classificazione generale delle azioni per analisi dinamiche all'interno dello *Structural Load Resource*: sarà necessario implementare un'abstract subentity di *IfcStructuralLoadOrResult* chiamata **IfcStructuralLoadDynamic** che reggerà, a sua volta, varie subentity: *IfcStructuralLoadSingleMass*, *IfcStructuralLoadLinearMass* e *IfcStructuralLoadPlanarMass* che, mantenute come le propongono Spagnuolo e Farello, servono a descrivere le forze inerziali della struttura, si aggiungeranno alcuni dettagli, come il fatto che possano riferirsi anche a connessioni mono e bidimensionali; le altre subentities sono *IfcStructuralLoadSpectralAcceleration*, *IfcStructuralLoadSeismogram* e

*IfcStructuralLoadDynamicEquivalentStatic* e combinano le due proposte finora analizzate, potendo inserire sia azioni ottenute da un accelerogramma, sia ottenute da un qualsiasi sismogramma. All'ultima *entity* vi si riferiranno, attraverso un apposito *attribute*, le stesse *entity* che descrivono un'azione in campo statico, difatti ciò serve a descrivere analisi statiche equivalenti. Si spera, così facendo, di esaurire le azioni possibili nell'ambito delle analisi dinamiche, sia relativamente alle analisi statiche equivalenti, sia relativamente all'utilizzo di spettri di risposta provenienti da modelli di progettazione e verifica come quelli provenienti dalle normative, oppure provenienti da terremoti effettivamente occorsi, sia, infine, l'aspetto relativo alle verifiche utilizzando lo storico degli spostamenti, velocità o accelerazioni di un sisma.

-**IfcStructuralLoadSingleMass** con tre *attributes* rivolti verso *IfcMassMeasure* descrive la massa lungo gli assi principali, con altri tre invece diretti verso *IfcMomentOfInertiaMeasure* descrive l'inerzia lungo gli stessi assi; la classe è lasciata praticamente uguale a come definita dalla software house, con l'unica aggiunta di una regola esplicita: poiché tutti gli *attributes* sono opzionali, si richiede che, qualora ne sia definito uno, anche gli altri che sono riferiti alla stessa *entity*, vengano specificati, anche se nulli;

-**IfcStructuralLoadLinearMass** viene definita con tre *attributes* opzionali, il primo, *Density*, che specifica la densità dell'*item* a cui si riferisce, il secondo, *DistributedMass*, specifica la massa distribuita equivalente che porta, mentre il terzo, *Damping*, lo smorzamento viscoso che gli è proprio. Per rendere ciò possibile è stato necessario implementare due ulteriori *derived types*: *IfcMassPerLengthMeasure*, *IfcViscousDamping*; il primo permette di descrivere un'unità di misura di kilogrammi/metro, mentre la seconda lo smorzamento, con unità di misura corretta, ossia di una forza fratto una velocità, di Newton\*secondo/metro. Come si specifica nel capitolo 6C;

-**IfcStructuralLoadPlanarMass** permette tre *attributes* opzionali, *Density*, *DistributedMass* e *Damping*, esattamente come *IfcStructuralLoadLinearMass*. L'unica vera differenza, poiché, è un'informazione relativa ad una massa distribuita su di una superficie, non su di una curva, l'*attribute DistributedMass*, si riferisce ad *IfcMassPerSurfaceMeasure*, un *derived type* che, appunto, descrive un'unità di misura di kilogrammi/metri quadri;

-**IfcStructuralLoadSpectralAcceleration** è ripresa così come proposta da Spagnuolo e Farello ma con una piccola differenza, vengono cioè eliminati i tre *attributes* che descrivono una generica accelerazione, lasciando così solamente la possibilità che questa classe descriva un preciso punto dello spettro di risposta, per la generica definizione di un'accelerazione, come da sismogramma, si utilizza la prossima classe;

-**IfcStructuralLoadBySeismogram** è modellata sulla classe *IfcStructuralLoadTimeHistory* come proposta da Serror et al. nel 2007, ma viene aggiunto un *select type*, **IfcStructuralLoadBySeismogramSelect** che permette la scelta tra un'accelerazione, una velocità o uno spostamento in modo che sia descrivibile compiutamente un qualsiasi sismogramma attraverso i tre *attributes*, *XValue*, *YValue* e *ZValue* che ne listano i valori significativi. Infine, la *relationship TimeStep* permette di raccontare la storia temporale listando dei valori significativi di tempi;

-**IfcStructuralLoadDynamicEquivalentStatic** è l'ultima *entity* atta a descrivere un carico dinamico, sebbene effettivamente descriva una combinazione di carichi statici; infatti trova la sua applicazione più generale delle analisi statiche equivalenti, come da nome, permettendo di catalogare le azioni statiche equivalenti ai carichi dinamici. Può fare ciò attraverso un *attribute*, *EquivalentLoad*, che lista una serie di *IfcStructuralLoadStatic*. Viene applicata una regola, *OnlyDynamicLoad* che esplicita che le azioni raccolte in questa categoria di azioni debbano essere solamente quelle equivalenti alle azioni dinamiche, escludendo le azioni di tipo statico che sarebbero comunque presenti, come i carichi portati. Questo perché questi carichi sono listati attraverso *Values* di *IfcStructuralLoadConfiguration* che permette così, in questo caso, di distinguere quali siano i carichi tra carichi portati e carichi inerziali.

Si è scelta come punto di partenza la classificazione proposta da Spagnuolo e Farello poiché presenta due notevoli lati positivi completamente assenti nell'altra proposta: il fatto che sia possibile, sin da subito, descrivere una combinazione di carico statico di un'analisi dinamica. Infatti rimane, in generale, vero che è necessario descrivere anche i carichi verticali e statici che insistono sulla struttura contemporaneamente all'azione dinamica. Mantenendo lo schema di Spagnuolo e Farello questo è immediato, inserendosi organicamente nell'architettura IFC. Creando *IfcStructuralLoadDynamic* come *subentity* di *IfcStructuralLoadOrResult* è possibile descrivere una combinazione di carico attraverso

*IfcStructuralLoadConfiguration* e il suo *attribute Value* che lista i vari oggetti di *IfcStructuralLoadOrResult*. Questo implica che sia possibile descrivere una combinazione di carico che preveda sia oggetti di *IfcStructuralLoadDynamic* che di *IfcStructuralLoadStatic*, ovvero sia azioni dinamiche che azioni statiche.

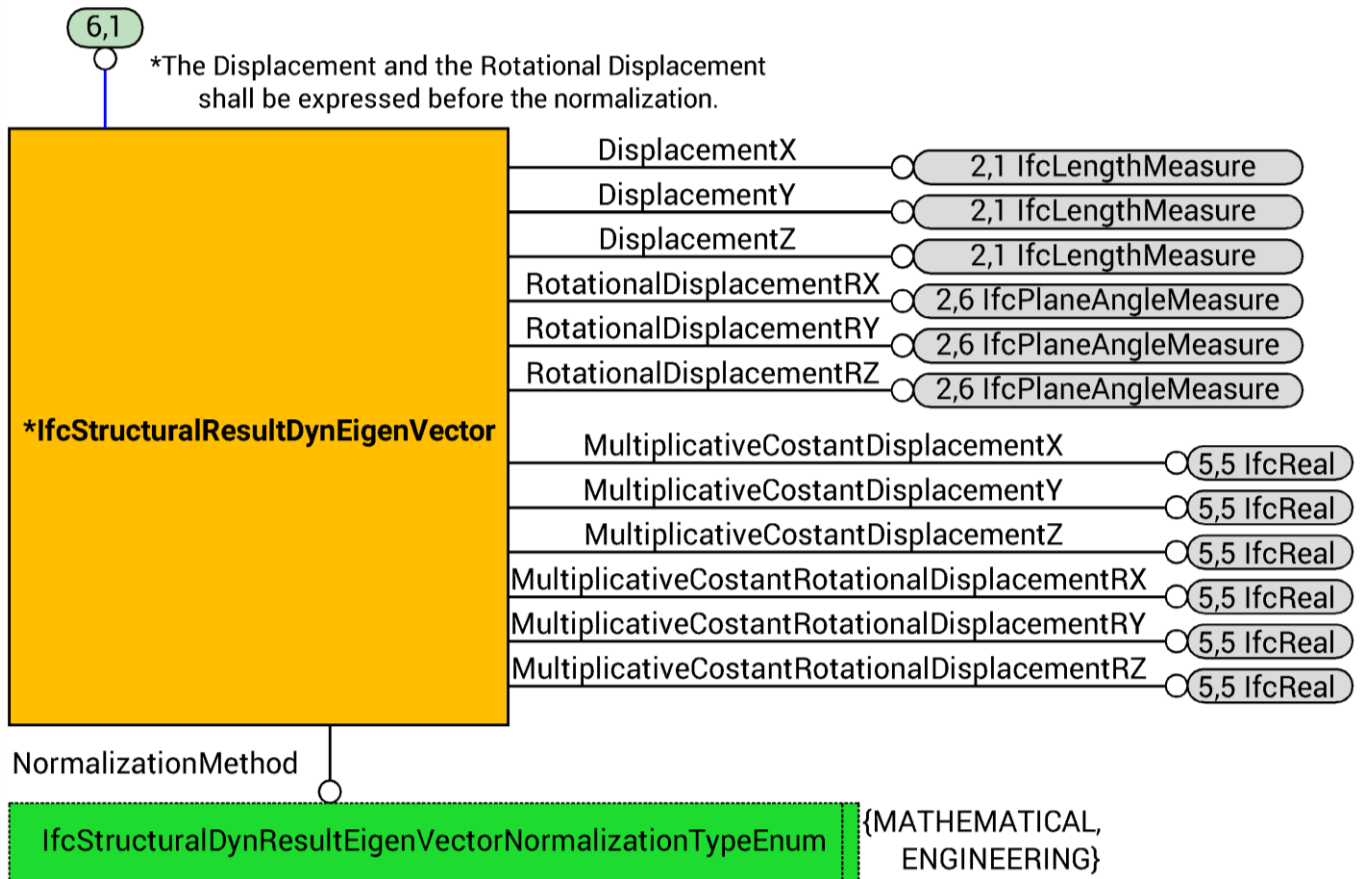
Si riprende, infine, una particolare modifica così come proposta da Spagnuolo e Farello: l'aggiunta delle enumerazioni per *IfcAnalysisTheoryTypeEnum*. Vengono aggiunte alcune enumerazioni per coprire anche le varie tipologie di analisi pienamente dinamiche, ma, come si vedrà nel capitolo 7C., anche le enumerazioni per poter descrivere analisi per meccanismi di collasso fuori dal piano e per solaio non rigido, così facendo si spera di gettare le basi per un ulteriore ampliamento delle *Industry Foundation Classe* per quanto riguarda il restauro e consolidamento strutturale. In ogni caso questo verrà spiegato meglio nel capitolo 8.

Figura 62 - Proposta risultati analisi dinamica. Fattori di smorzamento, fattori di partecipazione, autovalori





Figura 63 - Proposta risultati analisi dinamica. Autovettori e relativa normalizzazione.



## REAZIONI E RISULTATI

Come sempre, in arancione le *entity* nuove, in verde scuro i nuovi *types*, in celestino i *reference* a nuovi *data*, in questo caso un *defined type*.

Le classi **IfcStructuralDynamicReaction** e **IfcStructuralDynamicResult** avanzate da Spagnuolo e Farello si rivelano essere fin troppo coerenti e funzionali; tuttavia il riferimento avanzato da NAFEMS prevede una maggior completezza di informazioni. La software house italiana prevede che nelle suddette *superentities* rientrino quattro *subentities*, atte a descrivere autovalori, autovettori, fattori di smorzamento e fattori di partecipazione modale. Secondo l'agenzia per le analisi agli elementi finiti, quando si parla di analisi dinamiche, è necessario, in generale, poter descrivere anche coefficienti di

Rayleigh, gradi di libertà del sistema, pulsazioni e frequenze delle singole forme modali, costanti moltiplicative per la normalizzazione, matematica o ingegneristica, delle soluzioni, e la massa partecipante per ogni forma modale considerata. Il riportare queste informazioni non rende più corretti i risultati, questo deve essere chiaro, quanto rende più completa e trasparente l'informazione del processo ingegneristica. Comunque in generale tutte informazioni strettamente correlata alle *classes* già implementate da Spagnuolo e Farello. Questo comporta una leggera riorganizzazione, attraverso *attributes* che mettano in ordine i valori, per poter sistemare organicamente i risultati a seconda di quanti sono collegati a una precisa forma modale.

Questo significa che alle classi proposte da Spagnuolo e Farello si aggiunge una sola classe. La software house implementa due nuove *subentities* di *IfcStructuralLoadOrResult*: **IfcStructuralDynamicResult** e **IfcStructuralLoadDynamic**, e la seconda è già stata affrontata mentre la prima è atta a raccogliere e classificare i risultati di un'analisi dinamica. Essendo, infatti un' *abstract entity* raccoglie sotto di sé quattro *subentities*, formalmente identiche a quelle proposte da Softing srl, e modificate nei loro *attributes* per

poterle migliorare. Difatti,

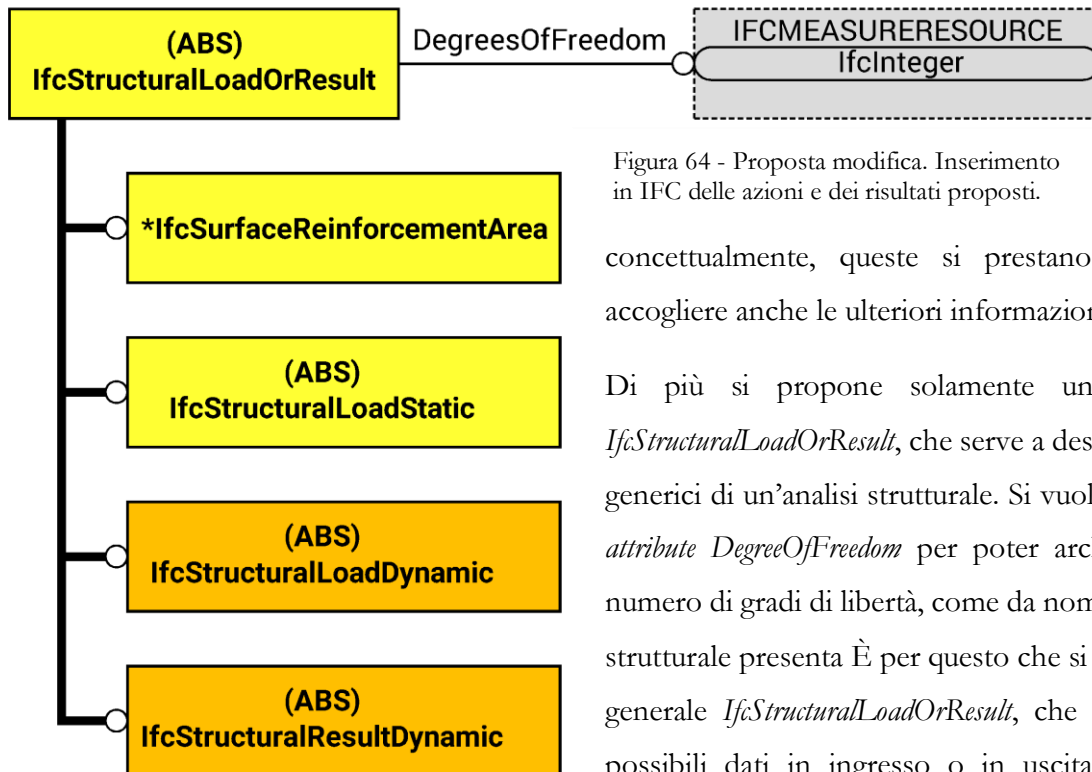


Figura 64 - Proposta modifica. Inserimento in IFC delle azioni e dei risultati proposti.

concettualmente, queste si prestano benissimo ad accogliere anche le ulteriori informazioni necessarie.

Di più si propone solamente una modifica di *IfcStructuralLoadOrResult*, che serve a descrivere i risultati generici di un'analisi strutturale. Si vuole aggiungere un *attribute DegreeOfFreedom* per poter archiviare anche il numero di gradi di libertà, come da nome, che il sistema strutturale presenta. È per questo che si considera l'entity generale *IfcStructuralLoadOrResult*, che raccoglie tutti i possibili dati in ingresso o in uscita di un sistema

strutturale, e vi si porta un *attribute* verso un numero intero per poter ricordare quanti siano i gradi di libertà di un sistema: l'*attribute* è *DegreesOfFreedom*. Il grafico mostra ciò, in arancione le *entity* nuove, da noi definite, in giallo quelle già esistenti in IFC.

Tornado ad *IfcStructuralDynamicResult*, questa presenta un unico *attribute*, *Mode*, verso *IfcInteger* che serve a numerare a quale forma modale corrisponda il risultato descritto. Si ricorda, infatti che questa è un'*abstract entity* e che quindi ogni ulteriore definizione sia specificata dalle sue *subentities*.

**-IfcStructuralDynResultDampingFactor:** secondo la formulazione originaria di Softing prevedeva solo l'*attribute Value* verso *IfcReal*, per correttezza è stata sostituita la classe di destinazione, mantenendo lo stesso nome per l'*attribute*, in modo che si faccia riferimento a **IfcViscousDamping**, una nuova classe dello *Measure Resource Layer* che descrive correttamente, come in termini di unità di misura, la grandezza considerata; vengono inoltre aggiunti due *attributes* opzionali per descrivere i coefficienti  $\alpha$  e  $\beta$  di Rayleigh e una regola esplicita per non permettere di compilare solo uno dei due coefficienti;

**-IfcStructuralDynResultEigenValue:** rimane simile nell'impostazione generale a quanto proposto da Spagnuolo e Farello, un'*entity* con l'*attribute Value* collegato ad *IfcReal* per descrivere il valore assunto dall'autovalore. Per soddisfare alle direttive della *National Agency for Finite Elements Methods & Standards* si aggiungono due *attributes* che riportano i corrispettivi valori di pulsazione e frequenza del sistema che l'autovalore porta come risultato. Il primo *attribute* è *Pulsation* e fa riferimento al *derived type IfcPulsationMeasure* che si rende necessario implementare, mentre il secondo è *Frequency* e fa riferimento ad *IfcFrequencyMeasure*;

**-IfcStructuralDynResultEigenVector:** qui avviene la più grande modifica rispetto alla proposta di Spagnuolo e Farello. Dal confronto con lo standard NAFEMS è emersa la necessità di esprimere il metodo di normalizzazione di un autovettore e i valori in questione. Questo per un'ingegneria più trasparente durante tutto il processo, attraversando progetto e verifica. Ai sei *attributes* che riportavano spostamenti e rotazioni del singolo autovettore che Spagnuolo e Farello proponevano, sono stati aggiunti altri sette *attributes*, i primi sei riportano le costanti moltiplicative applicate a ciascun valore, il settimo la regola per la normalizzazione del vettore utilizzata. La regola *ResultFromCalculus* fa sì che i

valori riportati degli autovettori siano espressi prima di ogni normalizzazione. I primi sei *attributes* sono: *DisplacementX*, *DisplacementY*, *DisplacementZ*, *RotationalDisplacementX*, *RotationalDisplacementY* e *RotationalDisplacementZ*. I primi tre sono riferiti ad *IfcLengthMeasure*, i secondi tre ad *IfcPlaneAngleMeasure*. I sei *attributes* per la normalizzazione sono: *MultiplicativeCostantDisplacementX*, *MultiplicativeCostantDisplacementY*, *MultiplicativeCostantDisplacementZ*, *MultiplicativeCostantRotationalDisplacementX*, *MultiplicativeCostantRotationalDisplacementY* e *MultiplicativeCostantRotationalDisplacementZ* e, poiché si tratta di unità adimensionali, fanno tutti riferimento ad *IfcReal*. Infine l'ultimo *attribute*, *NormalizationMethod*, si riferisce a un *enumerated type* per poter riportare la regola di normalizzazione dell'autovettore utilizzata solitamente di tipo matematico o ingegneristico. L'*enumerated type* è *IfcStructuralDynResultEigenVectorNormalizationTypeEnum* e ha quindi quattro enumerazioni: *MATHEMATICAL*, *ENGINEERING*, *USERDEFINED* e *NOTDEFINED*.

**-IfcStructuralDynResultParticipationFactor:** così come proposta dai programmatori italiani è perfettamente funzionante, ma come per la standardizzazione dell'autovettore, NAFEMS richiede l'esplicitazione di ulteriori informazioni. Vengono, in concreto, collegati l'autovettore, la massa partecipante e le condizioni di spostamento rigido della struttura con cui il fattore di partecipazione è ottenuto. L'autovettore risulta già collegato attraverso la *superentity*: avendo catalogato *IfcStructuralDynamicResult* con un *attribute* che specifica la forma modale corrispondente, risultano univocamente collegati autovettore e fattore di partecipazione, rendendo superflua qualsiasi ulteriore specificazione. Sarà, per contro necessario implementare due *attributes*: il primo atto a collegare gli spostamenti del terreno, il secondo per esplicitare le masse partecipanti. Con il primo si coglie l'opportunità di definire una nuova *class*: *IfcDynBoundaryCondition* per esplicitare le condizioni dinamiche dei vincoli. Tornano utili anche per la definizione delle azioni esterne, almeno in fase preliminare, prima che trasformarli in azioni impresse alla struttura, piuttosto che al terreno e quindi effettuare il calcolo effettivo. Si collegherà quindi un'*attribute* opzionale *RigidDisplacement* equivalente allo spostamento rigido impresso alla struttura del terreno, quindi l'*entity* collegata sarà *IfcDynBoundaryDisplacement*. Per il secondo si implementa una semplice *relationship* opzionale

*ParticipantMass* che lista i valori da calcolo; l'*entity* per fare ciò sarà *IfcMassMeasure*. Si è scelto di tenerli opzionali visto che l'informazione è comunque completa, anche senza una loro definizione.

## **7c. un sistema coerente**

Come si è già avuto modo di anticipare, per la proposta che si avanza è stato necessario implementare nuove *classes* e nuovi *types* all'interno di IFC ma in diversi *data schemas*. Questo è anche il motivo per cui ogni ulteriore specificazione di questo standard va attentamente pesata e valutata, rendendo particolarmente delicato ampliare per argomenti ampi, come può essere il mondo del restauro che si confronta con edifici già esistenti di cui si conosce poco.

Anche questa proposta ha dovuto confrontarsi con la delicatezza di intervenire in un sistema ampio: il risultato è stata la necessità di implementare nuovi vari *derived types* all'interno del *Measure Resource*, che colleziona tutti gli *objects* necessari alla misurazione di qualsivoglia quantità, dalle più semplici, come il metro e il kilogrammo, alle più complicate come i volumi d'aria all'ora e i le forza al metro dei momenti.

Questi *type* sono **IfcViscousDamping**, **IfcMassPerLenghtMeasure**, **IfcMassPerSurfaceMeasure**, e **IfcPulsationMeasure**. Sono tutti *subtypes* di *IfcDerivedMeasureValue* e sono formati tutti quanti da un *REAL*, cioè sono espresse da un numero reale. Per rendere la trattazione coerente e corretta è necessario, inoltre, aggiungere le enumerazioni *VISCOUSDAMPING*, *MASSPERLENGTH*, *MASSPERSURFACE* e *PULSATION* ad *IfcDerivedMeasureTypeEnum* dimodoché possano essere utilizzate al di fuori dello *schemas*.

Per la definizione completa *IfcStructuralLoadBySeismogram* è necessario implementare, sempre nel *Measure Resource*, la classe necessaria a descrivere la velocità: *IfcStructuralLoadBySeismogramSelect* si collega, fra gli altri ad **IfcVelocityMeasure** che raccoglie numeri reali con significato di velocità. Di nuovo, anche l'enumerazione in *IfcDerivedMeasureTypeEnum* è stata aggiunta: si tratta di *VELOCITY*. Sarà utile questo *defined type* anche per la definizione di *IfcDynBoundaryVelocity*. Similmente, per descrivere compiutamente *IfcDynBoundaryAcceleration*, si rende necessario implementare **IfcAngularAccelerationMeasure** per descrivere una grandezza di accelerazione angolare, di radianti al secondo quadro. Sarà accompagnata dall'aggiunta di *ANGULARACCELERATION* alle enumerazioni di *IfcDerivedMeasureTypeEnum*.

Guardiamo poi la modifica più importante: **IfcAnalysisTheoryTypeEnum**. Si implementa una modifica così come proposta da Spagnuolo e Farello. Originariamente il *type* presentava le enumerazioni:

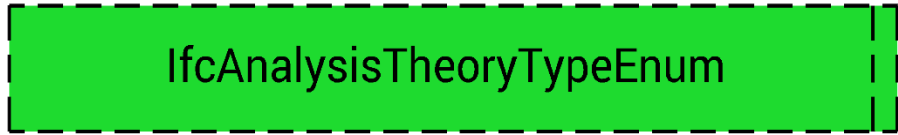


Figura 66 - IfcAnalysisTheoryTipeEnum.

- FIRST\_ORDER\_THEORY*: per analisi statiche tradizionali in cui le deformazioni sono trascurabili;
- SECOND\_ORDER\_THEORY*: per analisi statiche tradizionali in cui le deformazioni sono piccolo ma non trascurabili;
- THIRD\_ORDER\_THEORY*: per analisi statiche nell'ambito delle grandi deformazioni;
- FULL\_NONLINEAR\_THEORY*: per analisi statiche nell'ambito della piena non linearità, sia geometrica, sia dei materiali, quindi in campo elasto-plastico;
- USERDEFINED*: per analisi in cui in generale qualcosa sia diverso rispetto ai casi precedenti e quindi sia l'utente a definire l'ambito;
- NOTDEFINED*: per analisi in cui in generale non sia stato ancora specificato l'ambito di applicazione.

La software house italiana implementa quattro nuove enumerazioni, per descrivere vari tipi di analisi dinamiche e, soprattutto i metodi di sovrapposizione dei risultati:

- MODAL\_THEORY*: per analisi in ambito dinamico classico;
- MODAL\_SUPERPOSITION\_SRSS\_THEORY*: per analisi in ambito dinamico in cui la sovrapposizione dei risultati avvenga secondo il modo SRSS;
- MODAL\_SUPERPOSITION\_CQC\_THEORY*: per analisi in ambito dinamico in cui la sovrapposizione dei risultati avvenga secondo il modo CQC;



*-MODAL\_SUPERPOSITION\_CQC\_ASCE\_THEORY*: per analisi in ambito dinamico in cui la sovrapposizione dei risultati avvenga secondo il modo CQC-ASCE.

Come si vedrà nel successivo capitolo, questa enumerazione verrà ulteriormente ampliata.

# **8. ulteriori sviluppi**

Siccome IFC è in continua crescita e ben lontano ancora dall'essere completato, verranno fatte alcune considerazioni per poterlo estendere ancora e per poter arrivare in un secondo momento anche a trattare adeguatamente sotto ogni punto di vista anche edifici dal sistema strutturale in muratura portante ed edifici storici. Si vuole, in pratica, gettare le basi per un ulteriore completamento delle *Industry Foundation Classes* assolutamente necessario all'ingegneria italiana, i quale è decisamente necessario, nonostante i tempi, si guardi soprattutto ai software, non siano ancora completamente maturi. Finora si è cercato di poter descrivere compiutamente un qualsiasi modello agli elementi finiti e un qualsiasi tipo di azione sismica.

Allo stato attuale è molto importante che sia possibile descrivere un qualsiasi sistema edilizio. Difatti il dominio per le analisi strutturali è completo e perfettamente funzionante per descrivere un sistema strutturale in calcestruzzo armato, in acciaio o in legno. A parte la standardizzazione dell'informazione da rifinire, per quanto riguarda soprattutto l'ingegnerizzazione delle connessioni e quindi la micro-modellazione, allo stato attuale IFC è pronto per non sfigurare in descrivere ciò che si propone.

Ciò che però manca è la possibilità di descrivere un sistema portante costituito da muratura, sia essa nuova o storica. È senza dubbio necessario revisionare completamente lo standard perché ciò sia possibile. Difatti, poiché le *Industry Foundation Classes* vogliono descrivere l'industria delle costruzioni in maniera totale, si rivela assolutamente necessario poter includere anche ciò. Sia che si voglia realizzare un nuovo edificio in muratura, sia che, in generale, si voglia approntare in ambiente BIM un restauro, sia architettonico che strutturale, dati gli indiscussi vantaggi.

Sicuramente da riconsiderare il fatto che, ad esempio, non sia possibile, perché attualmente non necessario, descrivere un elemento strutturale analiticamente tridimensionale. Quando, invece, valutiamo il ribaltamento fuori piano di una porzione di muratura portante, consideriamo l'intero volume, non tanto nelle sue connessioni con il resto dell'organismo edilizio, ma piuttosto nella definizione della linea su cui si sviluppa la cerniera bidimensionale, dato dello schiacciamento al piede che è casualmente disposta sulla superficie di base. Può quindi rivelarsi necessario implementare ancora nello *Structural Analysis Domain* membri strutturali tridimensionali. Il concepire questo particolare elemento e la sua verifica può essere ricondotto al bidimensionale, con baricentri e vincoli

eccentrici, ma un membro strutturale tridimensionale rimane un argomento da non scartare a priori, soprattutto per dover descrivere elementi particolarmente massivi, si pensi ai macroelementi delle chiese storiche, per cui una schematizzazione analitica in curve e superfici può rivelarsi limitante. Se ricondursi ad una geometria analitica bidimensionale, è forse da preferire nell'ambito dei dissenti in atto o presunti di un edificio e della loro standardizzazione, rimane comunque utile implementare *entities* atte a raccogliere i dati di elementi strutturali tridimensionali per descrivere al meglio particolari eventualità riscontrabili nell'immenso settore delle costruzioni. Appunto, si pensi, a titolo di esempio ad alcuni pilastri di chiese storiche, con dimensioni in pianta che raggiungono e superano i 4 metri quadri: il loro comportamento deformativo può essere molto limitatamente descritto come monodimensionale.

Continuando sembra quindi utile aggiungere delle enumerazioni ad *IfcAnalysisTheoryTypeEnum* che possano descrivere ulteriori tipi di analisi dato che, allo stato attuale, sono descritti solamente tipi di analisi che prevedono comportamento unitario della struttura. Difatti molte analisi su edifici esistenti sono condotte per parti, data la concreta possibilità di collasso per meccanismi locali. Questo perché, quando i software che svolgono questo tipo di analisi si confronteranno con IFC, devono avere a disposizione un sistema coerente. Infatti finora IFC si è confrontato con problemi di realizzazione *ex novo*, e descrive molto bene il problema il sistema strutturale su cui si ritiene verificato un importante presupposto, che, spesso, è tutt'altro che rispettato negli edifici esistenti: appunto che il comportamento manifestato dalla struttura sia un meccanismo d'insieme, con solai rigidi capaci di trasmettere gli sforzi agli elementi verticali. Molto più frequente negli organismi edilizi datati, è il manifestarsi di meccanismi locali, in cui parti della struttura, scollegate dalle altre, raggiungono singolarmente il collasso, come, ad esempio, il caso di un muro di facciata che si ribalta lasciando il resto dell'edificio pressoché intatto.

Ed è infatti sulla base di questo presupposto che, nel confrontarsi con un edificio già realizzato, sia la normativa vigente, sia i ricercatori in materia, prescrivono indagini e interventi per sapere se questo comportamento spaziale dell'organismo strutturale abbia a verificarsi. Negli edifici con struttura in muratura portante questo è chiamato "comportamento scatolare", poiché le pareti murarie, tra loro variegatamente ortogonali o parallele o genericamente intersecate, possano collaborare e trasmettere

gli sforzi tra parete e parete e tra pareti e solai in modo simile ad una scatola. Viene detto “ammorsamento” la continuità tensionale tra le varie coppie di elementi strutturali.

Confrontando le necessità relative agli interventi sull'esistente, molto importanti nel nostro Paese data sia la quantità e l'importanza storica dell'esistente in muratura, appare necessario poterlo mettere in sicurezza. Ci appare imprescindibile, quindi, che anche le informazioni relative allo stato di fatto di un edificio già esistente debbano poter essere inserite in un file IFC, a cominciare dagli indicatori tipologici, soprattutto in una prospettiva di lungo termine, con la volontà di trasmettere ai posteri questo patrimonio.

Una prima idea che si vuole proporre è un nuovo *data schema* che racconti lo stato delle costruzioni storiche e descriva al meglio l'effettività dei presupposti per l'analisi dinamica, cioè, nei sistemi edilizi in muratura la presenza del cosiddetto comportamento scatolare. In questo nuovo *data schema* deve, verosimilmente, essere possibile governare, attraverso set di informazioni come “ammorsamento tra muri” il vincolo lineare che si viene a creare tra due murature ortogonali e che si sviluppa in altezza o, similmente, il vincolo orizzontale tra muratura e solaio attraverso “ammorsamento muro-solaio”. L'idea di fondo è che, nel caso in cui in una porzione di struttura ci sia il concreto rischio di attivazione di meccanismi locali, sia possibile collegare i vincoli interni agli elementi strutturali attraverso questo set di informazioni e viceversa. Concretamente vorrebbe dire, ad esempio, implementare un type che governi con *attribute* verso l'entity *IfcStructuralCurveConnection*, ad esempio, dell'*IfcStructuralAnalysisDomain*. Tutto ciò si collega intrinsecamente a una classe che contenga gli interventi di messa in sicurezza, a una che descrive le vulnerabilità e i dissesti, in atto o stabilizzati, quadri fessurativi, indagini. Questo potrebbe essere un unico grande *data schema* che informatizza i dati relativi al restauro. Ad esempio un'entity basata su di una polilinea può evidenziare l'allineamento delle aperture. Di per sé queste informazioni sono già inseribili all'interno di un qualsiasi software BIM, però è necessario che siano informazioni alla portata di tutti, che siano interoperabili, quindi standardizzate. Sarebbe il punto ultimo poter conservare in un file IFC anche qualità e stato di conservazione dell'esistente, indagini e valutazioni di sicurezza. Questo *data schema* prevede un lavoro molto consistente, per cui si preferisce rimandare la sua stesura, anche iniziale, così che chi lo faccia possa avere una prospettiva globale.

Tornando alla revisione di IFC, appare molto urgente, poi, descrivere un tirante e tutti gli elementi strutturali di un intervento tipici del restauro: è abbastanza sciocco pensare di poter descrivere completamente l'intera vita dell'edificio se non vengono aggiunte classi atte a descrivere gli interventi di restauro e consolidamento o, più in generale ancora le valutazioni di sicurezza dell'esistente. Ad esempio con una modifica di *IfcWall* dovrà descrivere un particolare tipo di muro: il contrafforte, che può essere anche un frenello poiché strutturalmente ha lo stesso comportamento; mentre nuove classi dovranno descrivere, appunto i tiranti e i vari paletti possibili. Se le barre d'acciaio sono già descritte all'interno di IFC, diversamente si deve dire per i vari fibrorinforzati, sia a matrice polimerica, sia naturale che vetrosa, anch'esse da standardizzare all'interno delle *Industry Foundation Classes*.

Volendo poi descrivere in IFC una struttura in muratura è da valutare molto negativamente il fatto che, allo stato attuale, tra i *Propriety Set* che descrivono un *IfcMaterialResource* si debba descrivere una muratura portante, anche nuova, attraverso un generico *Pset\_MaterialMechanical*, cioè un generico materiale di cui è possibile descrivere una viscosità dinamica, un modulo elastico, un modulo a taglio, un coefficiente di Poisson e un coefficiente di espansione termica. In concreto vuol dire che gli edifici in muratura non possono essere caratterizzati nei materiali costitutivi all'interno di IFC. Appare ovvio che questo set di proprietà debba essere o ampliato, o, alternativamente e preferibilmente, che debba essere costituito un set alternativo a sé stante per descrivere un elemento strutturale in muratura. Infatti non appare, ad esempio, né una resistenza a compressione, né una deformazione ultima. I vantaggi, poi, di creare un set di parametri per il materiale muratura sarebbero tanto più elevati considerando l'enorme numero di tipologie costruttive e di leggi costitutive presenti che, in alcuni casi, richiedono anche parametri molto diversi tra di loro come l'angolo di attrito alla Coulomb della muratura o la percentuale di vuoti.

In tutto ciò non è ancora possibile interfacciare attraverso IFC un modello di un solutore che non lavori agli elementi finiti, primi fra tutti i modelli a telaio equivalente, che sono i software più diffusi per la verifica e la progettazione di strutture in muratura.

Il lavoro è molto complesso e ampio, e si spera di gettare nel prossimo capitolo almeno le basi per poterlo cominciare, perché si tratta di valutare lo standard IFC in molte sue parti, di differenti *schemas*.

Come si è citato, il descrivere ulteriori materiali strutturali è una modifica che va a toccare il *Material Resource*; il descrivere oggetti architettonici e strutturali come il tirante è una modifica che riguarda lo *Shared Building Element* così come il descrivere un contrafforte. Descrivere le fibre con cui si effettuano gli interventi di rinforzo, invece, è un qualcosa che va a modificare lo *Structural Element Domain*. Per descrivere un modello a telaio equivalente infine si modifica ancora lo *Structural Analysis Domain*.

# **8A. in pratica**



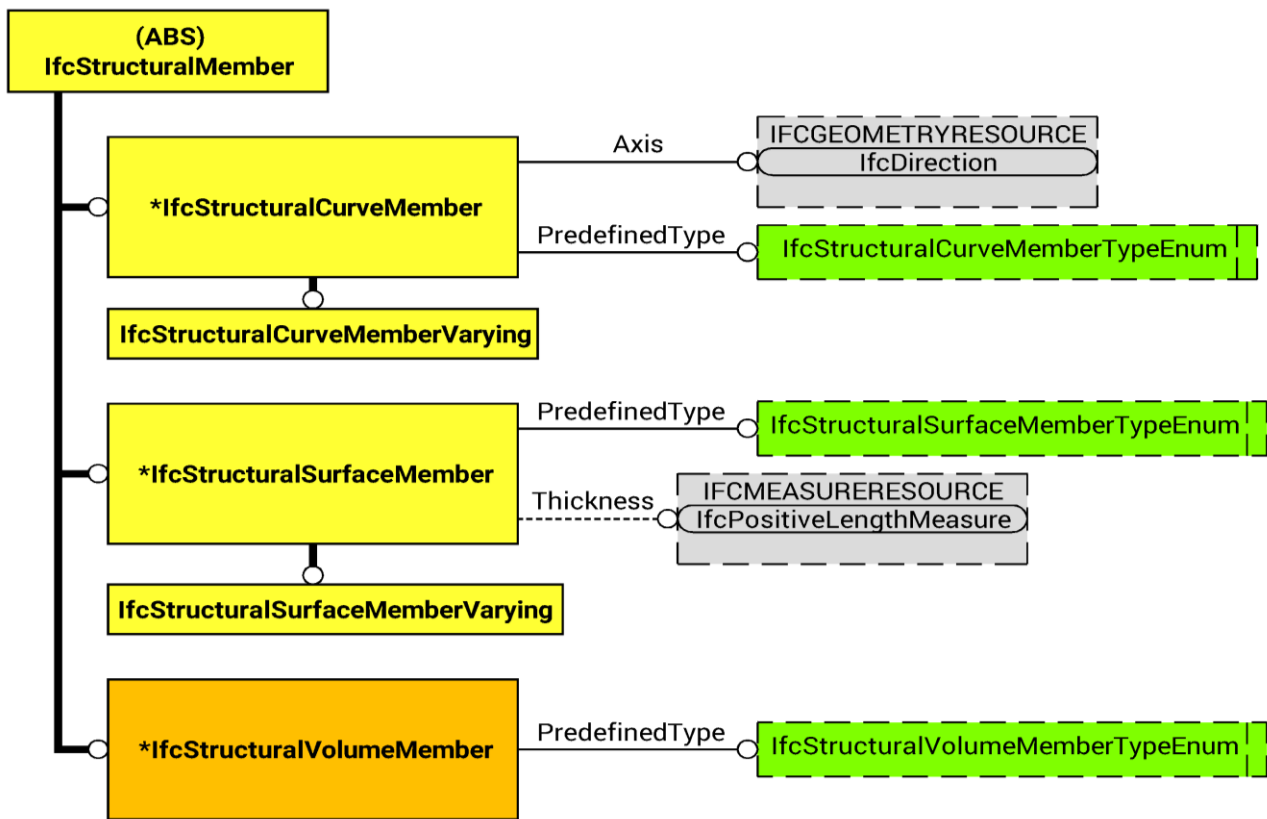


Figura 67 - Proposta modifica. Elemento strutturale analitico tridimensionale. Solita convenzione, in arancione la nuova *entity*, il resto è già presente in IFC.

## IFC STRUCTURAL VOLUME MEMBER

Si definisce **IfcStructuralVolumeMember** per applicazione particolari, senza che si rendano necessarie ulteriori connettività, data l'impossibilità di un collegamento a tre dimensioni non schematizzabili in due dimensioni; cioè non sono necessarie nuove classi per descrivere nuovi tipi di connessioni tra i diversi elementi.

In pratica questa implementazione è molto semplice e costruita similmente alle schematizzazioni mono e bidimensionali: una *subentities* di *IfcStructuralMember*. Non necessiterà di particolari *attributes* che ne descrivano particolarità della geometria, né di rappresentazioni topologiche direzionali. Sarà utile solamente l'*attribute PredefinedType* che ne descrive il tipo di elemento attraverso l'*enumerated type*

**IfcStructuralVolumeMemberTypeEnum.** Simile in tutto e per tutto ad *IfcFEMVolumeElementTypeEnum*, poiché devono, comunque corrispondersi anche per la regola applicata, questo presenterà le sue proprie enumerazioni per assolvere a questo compito:

-*BRICK*: per un elemento dalla geometria solida tradizionale;

-*AXISSYMMETRICAL*: per un elemento dalla geometria complessa, ottenuta per rivoluzione o, comunque, come porzione di un elemento a base circolare;

-*SOLID\_SHELL*: per un elemento analitico tridimensionale in cui le dimensioni prevalenti siano due, ma non sia assimilabile ad un elemento bidimensionale per particolari motivi;

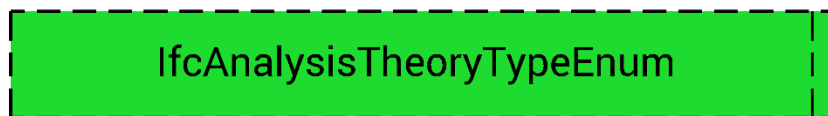
-*USERDEFINED*: per elementi particolari così come definiti dall'utente o dal software;

-*NOTDEFINED*: per elementi non ulteriormente specificati.

Così facendo la definizione degli elementi analitici non potrebbe essere più completa.

## IFC ANALYSIS THEORY TYPE ENUM

Dopo queste considerazioni è parso molto utile estendere ancora l'*enumerated type* *IfcAnalysisTheoryTypeEnum*. Si vuole che *Industry Foundation Classes* siano pronte per poter accogliere modelli e risultati per analisi dei meccanismi locali. Difatti non è infrequente che per edifici esistenti, soprattutto in muratura, avvengano collassi parziali per cause sismiche, spesso per perdita dell'equilibrio di porzioni murarie. Le verifiche con riferimento a questo tipo di azioni possono essere svolti tramite l'analisi dell'equilibrio, se gli elementi strutturali non si disgregano, secondo l'approccio



cinematico e la valutazione dell'azione orizzontale che attiva tale cinematismo.

Figura 68 - *IfcAnalysisTheoryTypeEnum*.

Il vantaggio di questi approcci detti “cinematici” è che permettono di determinare l'andamento dell'azione orizzontale che la struttura è progressivamente in grado di sopportare man mano che il meccanismo di collasso causato dal terremoto si evolve. Si costruisce una curva che è espressa attraverso un moltiplicatore  $\alpha$ , dato da un rapporto tra le forze orizzontali applicate ed i corrispondenti pesi delle masse presenti, rappresentato in funzione dello spostamento  $d_k$  di un punto di riferimento del sistema. Questa curva viene determinata fino all'annullamento di ogni capacità di sopportare azioni orizzontali cui corrisponde il punto in cui  $\alpha$  si azzera. Questa curva può essere trasformata nella curva di capacità di un sistema equivalente ad un grado di libertà, per cui si può valutare la capacità di spostamento ultimo del meccanismo locale, da confrontare con la domanda di spostamento richiesta dall'azione sismica.

Per ogni possibile meccanismo locale ritenuto significativo per l'edificio, si applica questo metodo, iniziando con il trasformare una parte della costruzione in un sistema labile, ossia con il costruire la catena cinematica del meccanismo, individuando i corpi rigidi in gioco, definiti da piani di frattura ipotizzabili per la scarsa resistenza a trazione della muratura, in grado di ruotare o scorrere tra loro; quindi prima valutando il moltiplicatore orizzontale dei carichi  $\alpha_0$  che comporta l'attivazione del meccanismo, poi l'evoluzione del moltiplicatore orizzontale dei carichi  $\alpha$  via via che lo spostamento  $d_k$  di un punto di controllo della catena cinematica aumento, solitamente scelto in prossimità del

baricentro delle masse, fino all'annullamento della forza sismica orizzontale. Dopodiché si trasforma la curva così ottenuta nella curva di capacità, ovvero in accelerazione  $a^*$  e spostamento  $d^*$  spettrali, valutando lo spostamento ultimo per collasso del meccanismo attraverso le verifiche di sicurezza per i vari Stati Limite, controllando sia la compatibilità degli spostamenti che le resistenze richieste dalla struttura.

Di solito si considera che la muratura in questione abbia resistenza nulla a trazione e che non vi sia attrito, quindi resistenza allo scorrimento, tra gli elementi coinvolti nel cinematisimo. È opportuno, a volte, considerare approssimativamente l'attrito tra i blocchi, la resistenza delle connessioni tra gli elementi murari, la presenza di tiranti e, soprattutto, la presenza di paramenti non collegati fra di loro, ad esempio, per mancanza di diafani trasversali. Molto spesso si valuta una resistenza a compressione non infinita della muratura, valutando l'arretramento al piede, rispetto cioè allo spigolo della sezione, della cerniera attorno alla quale si evolve il meccanismo.

Le nuove enumerazioni che in IFC individueranno analisi condotte da software secondi questi metodi sono:

*-LINEAR\_CINEMATIC\_THEORY*: per esplicitare modelli e analisi condotte con il metodo delle analisi cinematiche lineari. Queste vogliono ottenere come risultato un moltiplicatore orizzontale  $d_k$  dei carichi che porta all'attivazione del meccanismo di collasso locale. È un metodo che prevede l'applicazione di forze a degli elementi strutturali, schematizzati come corpi rigidi. Queste saranno date dai pesi propri e applicate nel baricentro, dai carichi verticali portati dagli stessi, come i carichi dati da solai, copertura ed elementi non strutturali, e da un sistema forze orizzontali proporzionali ai carichi verticali portati, se non trasmesse ad altre parti dell'edificio. Infine considerano eventuali forze interne. Si applica, quindi, il principio dei lavori virtuali, assegnando una rotazione virtuale  $\theta_k$  al blocco che si sta considerando, determinando gli spostamenti causati dalle diverse forze applicate nella loro direzione. Si eguaglia di seguito il lavoro totale eseguito dalle forze esterne ed interne applicate in corrispondenza dell'atto di moto rigido virtuale. Così facendo si ottiene il moltiplicatore orizzontale  $d_k$  e lo si può confrontare con la massima accelerazione attesa al suolo;

*-NONLINEAR\_CINEMATIC\_THEORY*: per descrivere modelli e analisi condotte con il metodo delle analisi cinematiche non lineari. Si vuole conoscere la capacità di spostamento della struttura, per il meccanismo considerato, dall'attivazione al collasso e il relativo moltiplicatore dei carichi, valutato sia all'attivazione del cinematismo ma anche su varie configurazioni della catena cinematica, tali che siano rappresentative dell'evoluzione del meccanismo e descritte dallo spostamento  $d_k$  di un punto di controllo del sistema. Si conduce l'analisi dell'evoluzione del cinematismo fino alla configurazione cui corrisponde l'annullamento del moltiplicatore dei carichi orizzontali; a questo punto si redige il grafico della funzione, per punti, così determinata. Se si considerano le diverse azioni costanti all'evolversi del cinematismo si può determinare solamente lo spostamento  $d_{k,0}$  cui corrisponde l'annullamento del moltiplicatore. In questo caso si può esprimere la configurazione attraverso una generica configurazione geometrica variata, funzione della rotazione finita incognita  $\theta_{k,0}$  che verrà determinata attraverso, sempre, il Principio dei Lavori Virtuali. Se si tiene conto del progressivo aumento delle forze esterne con l'evolversi del cinematismo, la curva la si può considerare lineare a tratti, in corrispondenza di spostamenti per cui avvengono eventi significativi.

Arrivati a questo punto si definisce la curva di capacità dell'oscillatore equivalente, come relazione tra l'accelerazione  $a^*$  e lo spostamento  $d^*$ . Si valuta la massa partecipante al cinematismo  $M^*$  considerando gli spostamenti virtuali dei punti di applicazione dei diversi pesi, associati al cinematismo, come una forma modale di vibrazione. Si calcola, poi, l'accelerazione spettrale  $a^*$  moltiplicando  $\alpha$  per l'accelerazione di gravità e dividendo per la massa partecipante, pesata sul fattore di confidenza. Lo spostamento spettrale  $d^*$  dell'oscillatore equivalenti può essere calcolato come spostamento medio dei diversi punti di applicazione dei pesi, pesato sugli stessi. In via approssimata, attraverso lo spostamento del punto di controllo  $d_k$ ,  $d^*$  è definibile con riferimento agli spostamenti virtuali valutati sulla configurazione iniziale. La curva può, alla fine di tutto, risultare essere lineare o, se le forze esterne hanno entità variabile, lineare a tratti.

Lo Stato Limite di Danno sarà valutato per l'accelerazione spettrale corrispondente all'attivazione del meccanismo, mentre lo Stato Limite di Salvaguardia della Vita per lo spostamento minore tra il 40% dello spostamento per cui si annulla l'accelerazione spettrale, valutata su una curva costruita con

solamente le azioni presenti fino al collasso, oppure lo spostamento corrispondente a situazioni compatibili con fenomeni di instabilità locale, se valutabili, come lo sfilamento delle travi dagli appoggi.

Con queste due enumerazioni si spera quindi di poter far interfacciare con IFC e tutta la semantica e filosofia BIM anche i programmi strutturali dedicati al restauro che studiano, appunto, anche meccanismi di collasso di primo modo, i quali, finora, sono sempre rimasti esclusi da tutto questo nuovo mondo.

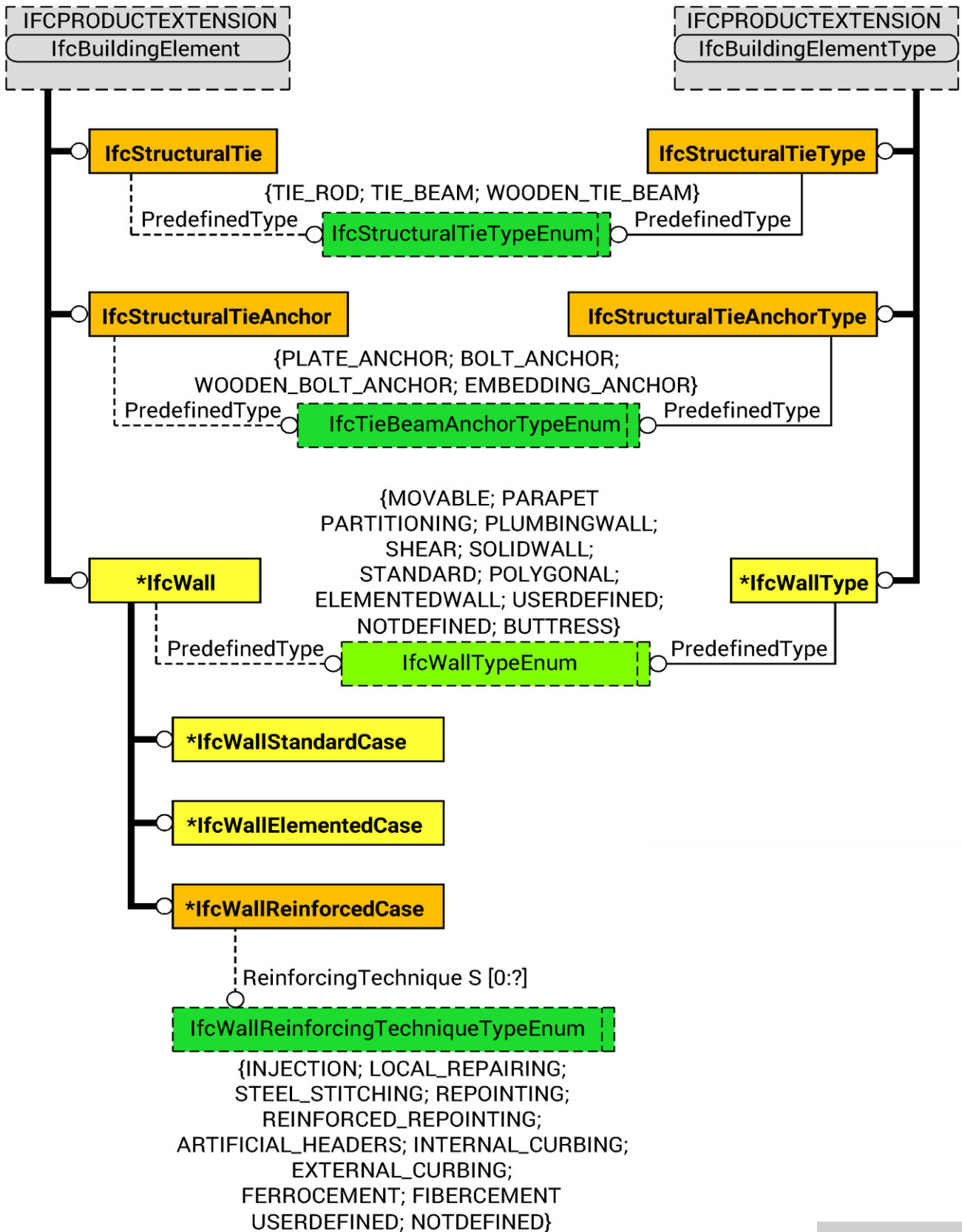


Figura 69 - Proposta modifica. Oggetti del consolidamento.

## OGGETTI DEL CONSOLIDAMENTO STRUTTURALE

La simbologia è quella utilizzata finora, in giallo le *classes* già esistenti, in arancione quelle proposte. In verde scuro i nuovi *enumerated types*, in verde chiaro quello già esistente, su cui viene proposta una modifica.

Come già anticipato, risulta impensabile che IFC possa descrivere tutto il ciclo di vita utile per un edificio, quando è tutt'ora esclusa la possibilità di descrivere gli elementi del consolidamento. È altresì impensabile che in un file BIM questi non rechino con sé le informazioni relative al miglioramento strutturale che introducono, quindi quando si parla di descrivere gli elementi si intende che vanno descritti sia strutturalmente che architettonicamente. Alla data attuale non è possibile nessuna delle due rappresentazioni, quindi si vuole implementare la rappresentazione architettonico di questi oggetti, che sono la conseguenza fisica degli interventi che generalmente si effettuano sull'esistente.

Si tratterà quasi completamente di ragionamenti all'interno dello *Shared Elements Domain* vista la presenza fisica di questi oggetti. La modifica più consistente riguarderà l'entity **IfcWall**. È una *instance* atta a descrivere un generico muro e le sue varie tipologie all'interno delle *Industry Foundation Classes*. Presenta due *subentities*, per poter descrivere muri standard, ossia rettilinei e ad altezza costante, e muri composti da vari sotto elementi: **IfcWallStandardCase** e **IfcWallElementedCase**. È collegato attraverso l'*attribute PredefinedType* all'*enumerated type IfcWallTypeEnum* che permette di identificare meglio la tipologia del muro stesso attraverso undici enumerazioni: *MOVABLE*, *PARAPET*, *PARTITIONING*, *PLUMBINGWALL*, *SHEAR*, *SOLIDWALL*, *STANDARD*, *POLYGONAL*, *ELEMENTEDWALL*, *USERDEFINED*, *NOTDEFINED*. Queste servono a descrivere, rispettivamente, muri semoventi o rimovibili, parapetti, partizioni interne o esterne, appositamente predisposti per contenere impianti idrico-sanitari, setti, cioè muri portanti resistenti anche taglio, muri ciechi, muri standard, muri dal profilo poligonale e dallo spessore variabile, muri composti, infine muri particolari e non ulteriormente definiti. Da notare due enumerazioni che sostituivano le *subentities* prima che queste fossero introdotte, che non sono state ancora oggi cancellate.



Già guardando quest'enumerazione ci si rende conto che debba essere ampliato, come già anticipato: dobbiamo poter categorizzare un muro come contrafforte o come frenello, due degli interventi più efficaci, diffusi e suggeriti per contenere le spinte orizzontali di archi e volte. Concettualmente si tratta di una stessa tipologia muraria, dalla stessa funzione e dallo stesso effetto statico, è necessaria quindi solamente una nuova enumerazione: *BUTTRESS*.

Si avanza una proposta di modifica così che nelle informazioni del muro siano inseribili le informazioni relative a un consolidamento. La proposta consiste nell'aggiungere un *subtype* di *IfcWall* che descriva un muro, appunto, consolidato: **IfcWallReinforcedCase**. A questa si aggiunge un *enumerated type* collegato attraverso l'*attribute AppliedReinforce* che lista quantitativamente i valori necessari. Difatti su un muro possono essere condotti più tipi di operazioni di consolidamento e questa scelta riflette questa possibilità. L'*enumerated type* è **IfcWallReinforcingTechniqueTypeEnum** ed elenca i tipi possibili di intervento con le sue enumerazioni. È possibile descrivere così, operazioni di iniezioni, riparazioni locali, scuci-cuci, riparazioni locali e scuci-cuci armati, ristilature dei giunti, ristilature armate dei giunti, aggiunta di diatoni artificiali, inserimento di tirantini orizzontali e verticali, cerchiature, placcaggi in intonaco armato, sia in rete elettrosaldata in acciaio, sia in reti polimeriche o altro. Le enumerazioni che rendono tutto ciò possibile sono:

-*INJECTION*: per descrivere un muro su cui sia stata fatta, o programmata, un'iniezione;

-*LOCAL\_REPAIRING*: per descrivere un muro su cui sia stata fatta, o programmata, una riparazione locale;

-*STEEL\_STITCHING*: per descrivere un muro su cui sia stata fatta, o programmata, una riparazione locale armata;

-*REPOINTING*: per descrivere un muro su cui sia stata fatta, o programmata, una ristilatura dei giunti;

-*REINFORCED\_REPOINTING*: per descrivere un muro su cui sia stata fatta, o programmata, una ristilatura armata dei giunti;

-*ARTICIAL\_HEADERS*: per descrivere un muro su cui sia stato eseguito, o programmato, un inserimento di diatoni trasversali artificiali;

-*INTERNAL\_CURBING*: per descrivere un muro su cui sia stato eseguito, o programmato, un inserimento di tirantini che possono essere disposti lungo le direzioni principali. Si utilizza questa enumerazione per descrivere i tradizionali metodi di rinforzo degli archi applicati all'intradosso o all'estradosso, con qualunque materiale;

-*EXTERNAL\_CURBING*: per descrivere un muro su cui sia stato eseguito, o programmato, un intervento di cerchiatura o di inserimento di tiranti su di un arco;

-*FERROCEMENT*: per descrivere un muro su cui sia stato eseguito, o programmato, un placcaggio in intonaco armato con reti d'acciaio;

-*FIBERCEMENT*: per descrivere un muro su cui sia stato eseguito, o programmato, un placcaggio in intonaco armato con fibre naturali, polimeriche, in vetro, o altro materiale, non acciaio;

-*USERDEFINED*: utilizzato per descrivere tecniche d'intervento su di un muro che non rientrano tra le precedenti;

-*NOTDEFINED*: per descrivere un intervento programmato, ma non ancora definite, né progettato.

Piccola specificazione è necessaria per poter spiegare le enumerazioni *INTERNAL\_CURBING* e *EXTERNAL\_CURBING*. Difatti il lettore avrà notato che questi saranno le enumerazioni che descriveranno anche gli interventi sugli archi perché questi non sono informaticamente un qualcosa di diverso da un muro. L'arco sarà infatti un foro nel muro, e consolidarlo significa dare stabilità alla schematizzazione digitale del "muro". Quindi per ulteriori specificazioni per quanto riguarda gli interventi si dovrà fare affidamento alla classe che descrive il muro, *IfcWall* o alle classi che descrivono gli oggetti di confinamento, come le barre d'acciaio, le fasce di fibrorinforzati, eccetera. Rimane in pratica corretto, si crede, la definizione di questi interventi con questa modalità.

Dopo questa implementazione ne rimangono solamente due, più semplici. Si vuole standardizzare la rappresentazione architettonica degli ultimi oggetti del restauro, dotati di una loro indipendenza fisico-geometrica, che sono ancora esclusi dallo standard: tiranti e rispettivi paletti. La modalità di ciò è decisamente snella e semplice rispetto anche alla descrizione migliorata del muro. Sarà necessario semplicemente aggiungere quattro *entities*, due che saranno sottoposte di *IfcBuildingElement* e le altre

due di *IfcBuildingElementType*, e due *enumerated types*. In pratica costruiamo **IfcStructuralTie** ed **IfcStructuralTieAnchor**, come *subentities* di *IfcBuildingElement*, appunto, e **IfcStructuralTieType** ed **IfcStructuralTieAnchorType** come *subentities* di *IfcBuildingElementType*. Per distinguerne la tipologia di tirante *IfcStructuralTie* e *IfcStructuralTieType* sono collegati attraverso l'*attribute PredefinedType*, opzionale nel primo caso, ad **IfcStructuralTieTypeEnum** che ha quattro enumerazioni:

- TIE\_ROD*: per descrivere un tirante classico in acciaio dalla sezione circolare;
- TIE\_BEAM*: per descrivere, invece, un tirante in acciaio di sezione qualsiasi, generalmente dotato di rigidità flessionale, sebbene spesso non sfruttata;
- WOODEN\_TIE\_BEAM*: per descrivere un tirante in legno;
- USERDEFINED*: come normale, descrive un ulteriore tipo di tirante;
- NOTDEFINED*: come normale, descrive un tirante per cui non siano state fatte ulteriori ipotesi.

Similmente *IfcStructuralTieAnchor* e *IfcStructuralTieAnchorType* sono collegati attraverso l'*attribute PredefinedType*, opzionale nel primo caso, ad **IfcStructuralTieAnchorTypeEnum**, anch'esso dotato di quattro enumerazioni:

- PLATE\_ANCHOR*: per descrivere una piastra d'ancoraggio in acciaio di un tirante;
- BOLT\_ANCHOR*: per descrivere un paletto d'ancoraggio in acciaio di un tirante;
- WOODEN\_BOLT\_ANCHOR*: per descrivere un paletto d'ancoraggio in legno di un tirante;
- EMBEDDING\_ANCHOR*: per descrivere l'inghisaggio di un tirante, ossia come da nome, l'ancoraggio chimico o meccanico;
- USERDEFINED*: per un ancoraggio particolare di un tirante;
- NOTDEFINED*: per un ancoraggio non ancora specificato.

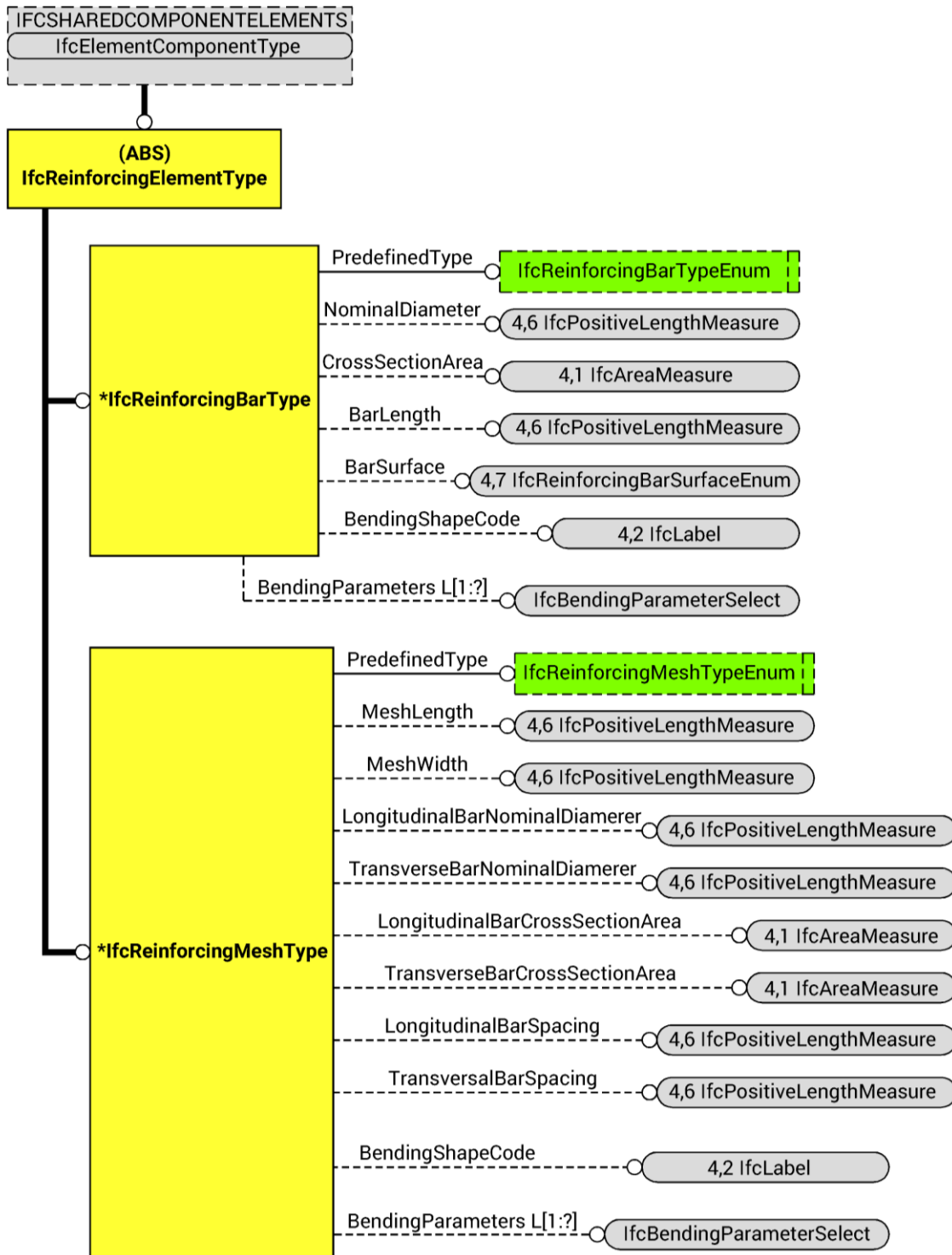


Figura 70 - IFC SED. Barre e reti d'armatura.

Infine, sarà necessario modificare lo *Structural Elements Domain*, il quale descrive oggetti strutturali dotati di una propria identità fisica ma, nonostante questo, non vanno a dialogare con la disciplina architettonica. Sono sistemi fondazionali e armature. In sostanza, quindi, è l'ambito perfetto per poter implementare la modellazione gli oggetti per gli interventi di rinforzo come le fibre di rinforzo. Purtroppo è impossibile descrivere queste con una semplice modifica delle *classes* esistenti, come si può vedere alla pagina precedente, poiché queste sono ampiamente ottimizzate per poter descrivere elementi di armatura in acciaio. Ad esempio gli *attributes* fondamentali implementano diametri, aree trasversali, superfici laterali totali e dimensioni, il tutto riferito sia a barre che a reticolati, sia per calcestruzzi armati ordinati che precompressi; una descrizione siffatta ha poca flessibilità, proprio perché riferita a specifici elementi costruttivi. Sarà quindi necessario implementare nuove *entites* che possano descrivere questi oggetti, basandosi sullo schema già proposto per le armature in acciaio.



Figura 71- Proposta modifica. Descrizione materiali fibrorinforzati. Al solito, in arancione le nuove *entities*, in giallo quelle già esistenti; in verde scuro i nuovi *types*.

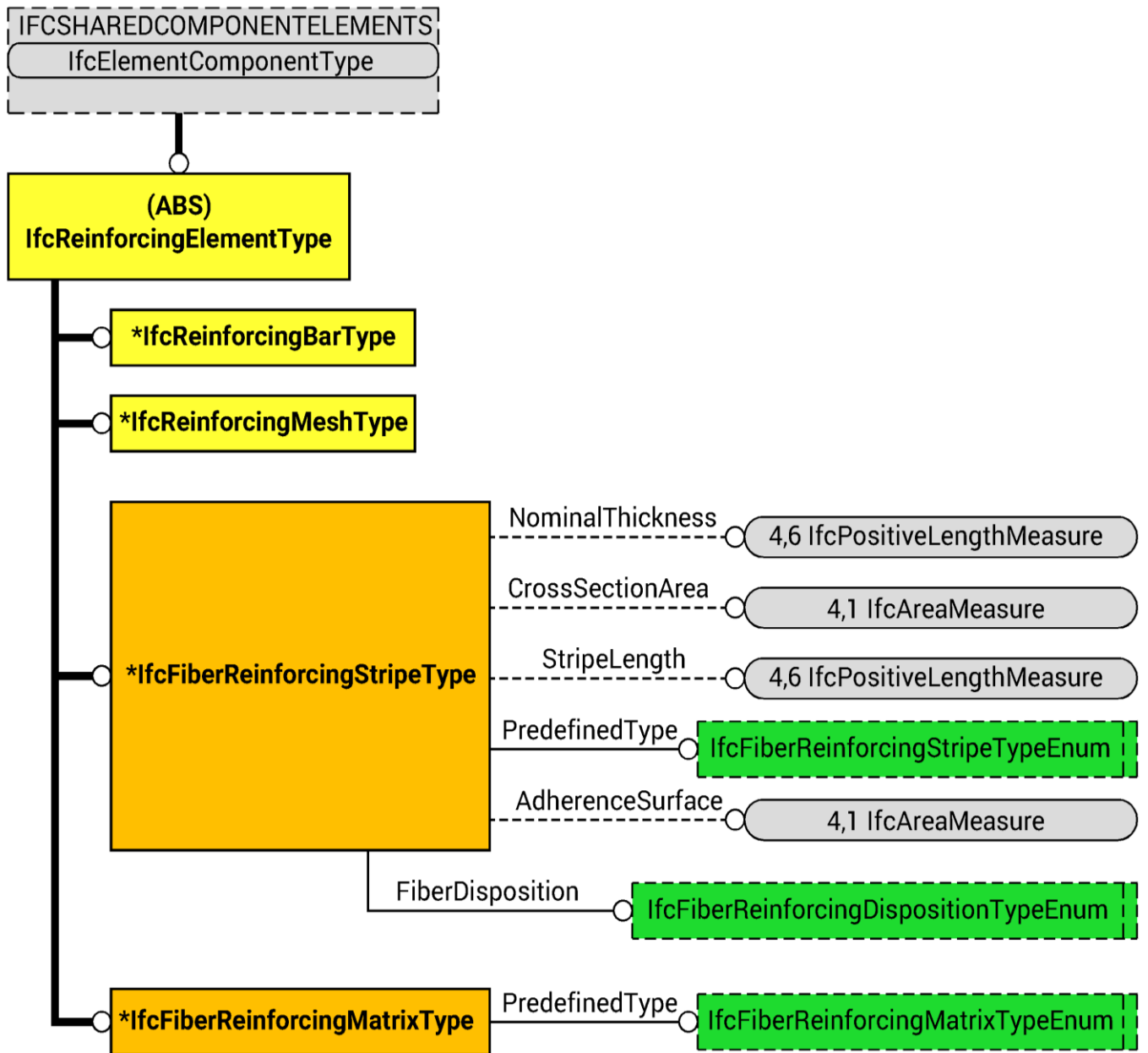


Figura 72 - Proposta modifica. Descrizione tipi di materiali fibrorinforzati. Al solito, in arancione le nuove *entities*, in giallo quelle già esistenti; in verde scuro i nuovi *types*.

In pratica significa implementare quattro *subentities* aggiuntive, una coppia sottoposta di *IfcReinforcingElement* e l'altra sotto *IfcReinforcingElementType* che descrivano strisce e di rinforzo e le matrici di supporto delle fibre di rinforzo<sup>66</sup>.

Sarà necessaria una singola coppia perché non è necessario distinguere tra interventi con fasce di fibre e superfici di fibre. La differenza sarebbe nelle dimensioni prevalenti. Non è necessario questo distinguo per la consuetudine di formare superfici di rinforzo con la sovrapposizione di fasce di rinforzo lungo le linee di tensione principali. Se in un secondo momento fosse necessario descrivere vere e proprie superfici di fibre a rinforzo strutturale, la loro implementazione sarebbe immediata e semplice.

Prima di tutto è necessario poter descrivere il materiale di questi interventi, quindi si aggiungerà ad *IfcReinforcingElement* un *attribute* *FiberMaterial* che, diretto verso *IfcLabel*, permette di descrivere il materiale della fibra. Le due nuove *entities* saranno **IfcFiberReinforcingStripe** e **IfcFiberReinforcingStripeType**, sul modello delle *classes* già presenti. Avranno ciascuna sei *attributes* sostanzialmente identici: *NominalThickness* diretto verso *IfcPositiveLengthMeasure* permette di descrivere lo spessore della fibra utilizzata, *CrossSectionArea* e *AdherenceSurface* diretti verso *IfcAreaMeasure* per descrivere, rispettivamente, l'area trasversale e la superficie d'aderenza totale, *StripeLength* diretto verso *IfcPositiveLengthMeasure* per descrivere la lunghezza della fibra applicata, poi *PredefinedType* diretto verso *IfcFiberReinforcingStripeTypeEnum* che permette di descrivere il tipo di fibra applicata, in base al materiale costituente, infine *FiberDisposition* verso *IfcFiberReinforcingDispositionTypeEnum* per descrivere la direzione, e quindi il comportamento, delle fibre. In questa ultima coppia di *attributes* risiedono le differenze tra i due *types*: tutti gli *attributes* testé elencati sono opzionali, ad eccezione di *PredefinedType* di *IfcFiberReinforcingStripeType* e di *FiberDisposition* di *IfcFiberReinforcingDispositionTypeEnum* che sono obbligatori.

---

66 M.R. Valluzzi, M. Munari, "Dispense del Corso di Restauro e Laboratorio. Tecniche di intervento. I materiali compositi.



La nuova enumerazione, comunque, permette, come già detto, la descrizione del tipo di fibra applicate, in base anche al materiale. Le enumerazioni di **IfcFiberReinforcingStripeTypeEnum** sono:

- CARBONFIBERREINFORCING*: descrive un rinforzo in fibre di carbonio;
- ARAMIDFIBERREINFORCING*: descrive un rinforzo in fibre di aramide;
- GLASSFIBERREINFORCING*: descrive un rinforzo in fibre di vetro;
- POLYEHTYLENFIBERREINFORCING*: descrive un rinforzo in fibre di polietilene;
- POLYVINYLALCOHOLFIBERREINFORCING*: descrive un rinforzo in fibre di polivinil alcol;
- BASALTFIBERREINFORCING*: descrive un rinforzo in fibre di basalto;
- STEELFIBERREINFORCING*: descrive un rinforzo in fibre di acciaio;
- LINENFIBERREINFORCING*: descrive un rinforzo in fibre di lino;
- HEMPFIBERREINFORCING*: descrive un rinforzo in fibre di canapa;
- USERDEFINED*: descrive un rinforzo così come definito dall'utente;
- NOTDEFINED*: descrive un rinforzo non ulteriormente specificato.

L'altra enumerazione descrive, appunto disposizione delle fibre, quindi il loro comportamento meccanico. È **IfcFiberReinforcingDispositionTypeEnum** e ha le seguenti enumerazioni:

- UNIDIRECTIONAL*: per descrivere un rinforzo in fibre con disposizione unidirezionale, lungo l'asse di applicazione;
- BIDIRECTIONAL*: per descrivere un rinforzo in fibre con disposizione bidirezionale, ortogonale e parallelo all'asse di applicazione;
- PSEUDOISOTROPIC*: per descrivere un rinforzo in fibre di con disposizione pseudoisotropa, cioè casualmente per avvicinarsi ad un comportamento isotropo dell'elemento.

L'altra coppia di *entities* implementata come *subtypes* di *IfcReinforcingElement* ed *IfcReinforcingElementType* è quella che descrive la matrice di supporto delle fibre di rinforzo. Sono *classes* molto semplici: **IfcFiberReinforcingMatrix** ed **IfcFiberReinforcingMatrixType**. Entrambe per funzionare hanno bisogno solamente di un *enumerated type* ciascuna, uguale per entrambe. Sarà collegato loro attraverso l'*attribute PredefinedType*. L'*enumerated type* è **IfcFiberReinforcingMatrixTypeEnum** e le sue enumerazioni saranno le seguenti:

- EPOXY*: per descrivere una matrice di applicazione in resina epossidica;
- POLYAMIDE*: per descrivere una matrice poliammidica di applicazione;
- POLYESTER*: per descrivere una matrice di applicazione a base di poliestere;
- GROUT*: per descrivere una matrice a base cementizia, tendenzialmente una malta o ancora più fina, di applicazione;
- BMI*: per descrivere una matrice di applicazione di intonaco di tipo BMI;
- PEEK*: per descrivere una matrice di applicazione a base di polietere etere chetone;
- USERDEFINED*: per descrivere una matrice di applicazione così come specificata dall'utente;
- NOTDEFINED*: per descrivere una matrice di applicazione non ulteriormente definita.

## PROPRIETÀ MURATURA

Un'altra modifica che fin da subito è implementabile e che almeno pone le basi per descrivere un sistema strutturale in muratura è l'implementazione di un *property set* che possa raccogliere le informazioni relative alla muratura come materiale da costruzione. Così facendo diventa possibile standardizzare quest'informazione nelle *Industry Foundation Classes* e, quindi, immagazzinare i dati relativi alla caratterizzazione dei materiali di una struttura siffatta. L'utilità di quest'implementazione è fondamentale per l'utilizzo dei software per analisi, potendo definire i parametri utili alla modellazione strutturale, senza questo *property set* la

Si crea così, all'interno del *Material Resource Layer* questo *property set* che raccoglie parametri della muratura così come richiesti dalle Normative, come gli Eurocodici, le Norme Tecniche per le Costruzioni, le Linee Guida per i Beni Culturali. È stato utile anche un confronto con l'attività di ricerca in materia. Si ricorda che la finalità ultima è la definizione di un qualcosa utilizzabile nei software di calcolo: non sarà presente traccia del processo decisionale che porta alla determinazione dei parametri. Questo rimane uno dei prossimi sviluppi di IFC, sempre nell'ottica di un'ingegneria trasparente, per cui alla fine del paragrafo si elencheranno una serie di proprietà che compaiono nel

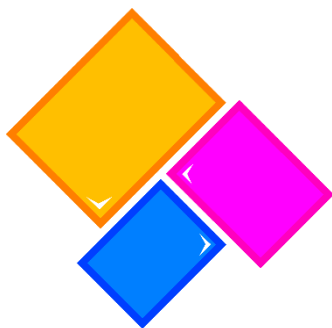


Figura 73 - IFC. Simbologia Property Set.



processo decisionale della caratterizzazione dei materiali strutturali, di cui non rimane però traccia e la cui decisione è tanto più importante quanto più si interviene su edifici esistenti e piccole differenze sui parametri scelti possono avere grosse ripercussioni sul progetto e la verifica<sup>67</sup>.

Il *property set* si chiama **Pset\_MaterialMasonry** e raccoglie i seguenti valori:

-**CompressiveStrength**: per poter descrivere la resistenza a compressione di progetto di una muratura. Per le considerazioni fatte prima il valore

67 - Circolare esplicativa, n. 617/2009, "Istruzioni per l'applicazione delle Nuove norme tecniche per le costruzioni".

che viene riportato è risultato di diverse formulazioni. È espresso come un valore singolo con unità di misura di pressione attraverso *IfcPressureMeasure*;



-**ShearStrength**: per poter descrivere la resistenza di una muratura di progetto. È espresso anch'esso come un valore singolo con unità di misura di pressione attraverso *IfcPressureMeasure*;



-**ModulusOfElasticity**: per poter descrivere il modulo di elasticità tangenziale secante di una muratura. È espresso anch'esso come un valore singolo con unità di misura specifica di un modulo di elasticità attraverso *IfcModulusOfElasticityMeasure*;



-**ShearModulus**: per poter descrivere il modulo di elasticità tangenziale di una muratura. È espresso anch'esso come un valore singolo con unità di misura specifica di un modulo di elasticità attraverso *IfcModulusOfElasticityMeasure*;



-**MassDensity**: i riferimenti normativi riportano, tra i parametri utili, il peso specifico della muratura. Per inserirsi adeguatamente all'interno dell'architettura di IFC e preferendo mantenere un'impostazione il più possibile coerente con il sistema internazionale di misura, si preferisce esprimere questa misura come densità, in ogni caso facilmente riconducibile al peso specifico. Nuovamente un singolo valore esprimerà adeguatamente questo parametro e lo farà attraverso il *derived type IfcMassDensityMeasure*;



-**ParallelCompressiveStrength**: ritorna utile in più di una circostanza, sebbene sia di estremamente difficile valutazione, il valore di resistenza a compressione della muratura per sforzi paralleli alla giacitura dei giunti. Questo soprattutto nella verifica di quelle che sono le fasce di piano di un sistema a comportamento unitario, in cui questi elementi in muratura fungono da elementi di accoppiamento tra due elementi a taglio. Sarà sempre espresso, nel nostro *property set* da un singolo valore attraverso, di nuovo, *IfcPressureMeasure*.



-**BendingDrift**: questo e il successivo sono parametri espressi come percentuali rispetto l'altezza di spostamento. Sono gli spostamenti per cui, da normativa, l'elemento considerato raggiunge lo Stato Limite Ultimo per il parametro della sollecitazione considerato. In

questo caso lo spostamento orizzontale in sommità causato da pressoflessione non può essere superiore a un certo valore percentuale dell'altezza. Lo si esprimerà come un valore singolo, attraverso *IfcPositiveRatioMeasure*, al fine di mantenerne l'espressione come percentuale dell'altezza;



**-ShearDrift:** come al punto precedente, esprime una percentuale di altezza da assumere come spostamento ultimo possibile orizzontale dell'elemento in muratura causato da sforzi di taglio. Come al caso precedente lo si esprimerà come un valore singolo, attraverso *IfcPositiveRatioMeasure*, al fine di mantenerne l'espressione come percentuale dell'altezza;

Questi sono i parametri necessari alla caratterizzazione di un elemento murario all'interno di un software strutturale. Sono, in pratica, i valori finali del processo decisionale, che ci permettono di svolgere i calcoli delle verifiche<sup>68,69</sup>. Però, come già anticipato, è tanto più necessario mantenere traccia del processo decisionale che ha portato alla determinazione di questi valori. È per questo che è necessario che in IFC rimanga standardizzata la possibilità di esprimere:

**-MortarCompressiveStrength:** ossia la resistenza a compressione della malta della muratura. Spesso influenza pesantemente il comportamento dell'intero paramento murario la sola resistenza della malta costituente la muratura. Siccome, poi, è necessario poter riportare quali parametri migliorativi e peggiorativi vengono considerati nella caratterizzazione dei materiali, verosimilmente il parametro "malta di buone caratteristiche" può essere unificato a questa enumerazione, magari facendo sì che sia possibile inserire due valori qui;

**-ConfidenceFactor:** per descrivere il fattore di confidenza considerato per la valutazione dei parametri della muratura. Questo è strettamente correlato al livello di conoscenza che si ha della struttura, in un futuro più o meno lontano si spera che, attraverso il BIM, si possa arrivare a fattori di confidenza sempre più bassi, dato che si può avere conoscenza completa del sistema edilizio. Rimane in ogni caso utile implementarlo, racconta molto delle valutazioni preliminari del progettista;

---

68 - Decreto Ministeriale 14/01/2008, "Norme Tecniche per le Costruzioni". Cap. 4.5.6.

69 - Circolare esplicativa, n. 617/2009, "Istruzioni per l'applicazione delle Nuove norme tecniche per le costruzioni". Cap. C8A.2.

**-VoidPercentage:** questa voce è necessaria per lo più per descrivere una muratura di nuova concezione, infatti oggi giorno gli elementi in laterizio sono, nella stragrande maggioranza dei casi, forati o semipieni, con percentuali variabili di fori e di vuoti, ma che influenzano il comportamento globale. Nella muratura storica invece i laterizi sono tradizionalmente pieni, anche se, ancora oggi sono utilizzati, sebbene in misura decisamente minore;

**-CoulombFrictionAngle:** sebbene compaia raramente in software dedicati alle specifiche analisi di elementi murari rimane molto utile per descrivere il processo decisionale sotteso all'ingegnerizzazione di una struttura in muratura. Si esprime, così, l'angolo di attrito alla Coulomb per la muratura, utile soprattutto per descrivere elementi storici, che il progettista valuta;

**-NoCompressionShearStrength:** dev'essere lasciata traccia di questo parametro che descrive la resistenza caratteristica a taglio in assenza di azione assiale di una muratura. Può essere espresso anch'esso come un valore singolo con unità di misura di pressione attraverso *IfcPressureMeasure*. Serve anche e soprattutto per raccontare le scelte progettuali effettuate nell'arco di vita dell'edificio con struttura muraria;

**-CompressionShearStrength:** similmente al parametro precedente, sarà necessario descrivere la resistenza caratteristica a taglio della muratura, questa volta in presenza di carico verticale. Sarà necessario, che il precedente valore *CoulombFrictionAngle* sia riportato per mostrare l'origine di quest'informazione. Di nuovo sarà espresso come un valore singolo attraverso *IfcPressureMeasure*. La sua utilità è, come sempre, nell'ambito di un'ingegneria trasparente;

**-ReductionResistanceCoefficient:** esprime il coefficiente di riduzione della resistenza del materiale, così come espressamente dimensionato dal progettista, considerate snellezza convenzionale dell'elemento strutturale e del coefficiente di eccentricità considerato<sup>70</sup>;

**-CorrectiveCoefficient:** all'interno di questo parametro, che sarà necessariamente fatto di valori multipli per la scelta dei coefficienti correttivi in questione, saranno appunto riportati i coefficienti

---

70 - Decreto Ministeriale 14/01/2008, "Norme Tecniche per le Costruzioni". Cap. 4.5.6.2.

migliorativi o peggiorativi scelti e il valore ad essi associato. Non sono tutti i valori previsti da normative, poiché, come già sottolineato, il valore di “malta buona” sarà da associato al valore *MortarCompressiveStrength*, il valore dato dai placcaggi in pareti armate può essere descritto agevolmente in un muro per strati, per cui i valori di resistenza e rigidezza nuovi sono dati dall’insieme degli elementi del muro, piuttosto che dall’omogeneizzazione dei parametri meccanici dell’elemento murario<sup>71</sup>. In ogni caso le voci necessarie saranno tutte quante dei valori singoli espressi con *IfcPositiveRatioMeasuring*. In dettaglio saranno:

-**ThinJoint**: esprimerà la presenza di giunti sottili di malta e il valore migliorativi assunto per questa particolare situazione;

-**MasonryRecurring**: per murature in pietra è qui che possiamo descrivere la presenza di ricorsi o listature in laterizio e il relativo valore assunto per il coefficiente migliorativo.

-**HeadersPresence**: esprimerà la presenza, sin dall’origine o in seguito ad interventi, di diaconi trasversali atti a collegare i vari paramenti murari e, di più, il relativo valore di coefficiente migliorativo assunto per il calcolo;

-**MediocreCore**: esprimerà un nucleo sia ampio, sia scadente e il relativo valore del coefficiente peggiorativo assunto nel determinare i parametri meccanici della muratura;

-**Injections**: sarà utile a descrivere un muro consolidato con iniezioni di miscele leganti e, soprattutto, il valore di coefficiente migliorativo assunto in seguito ad intervento di questo tipo.

Nel caso di interventi di consolidamento è bene creare un nuovo *property set* che descriva le caratteristiche della muratura all’indomani degli interventi. Rimane così traccia delle valutazioni migliorative fatte, mantenendo consultabili i valori precedenti. L’informatizzazione degli interventi in sé rimane estraneo a questo set di dati, come si è già potuto vedere.

---

71 - Circolare esplicativa, n. 617/2009, “Istruzioni per l’applicazione delle Nuove norme tecniche per le costruzioni”.  
Cap. C8A.2

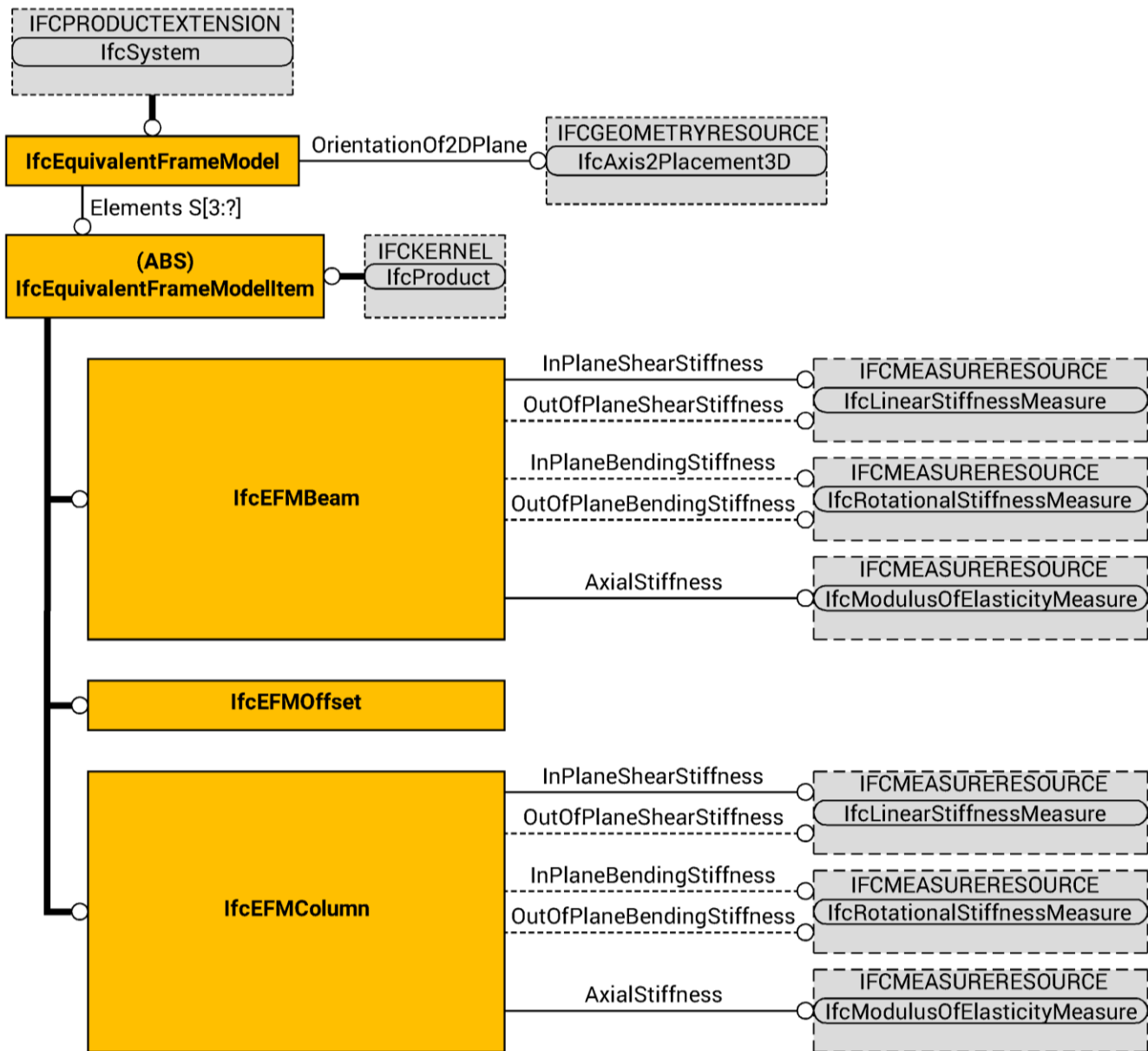


Figura 74 - Proposta modifica. Modello a telaio equivalente.



## MODELLO A TELAIO EQUIVALENTE

Arrivati a questo punto si è voluto implementare almeno la descrizione di un modello a telaio equivalente all'interno di IFC. È soltanto a questo punto che si è realizzato ancora di più quanto fosse giusto e necessario mantenere separata la definizione dei modelli analitico e agli elementi finiti, come anche quello a telaio equivalente: si rende evidente nel modellizzare questa schematizzazione strutturale come le dimensionalità dei modelli deve essere necessariamente separata. Difatti il modello a telaio equivalente schematizza i maschi murari e le fasce di piano come travi, pilastri e offset con una precisa deformabilità<sup>72,73</sup>.

Sono vari e diversificati i modelli a telaio equivalente e si differenziano, per lo più, sull'interpretazione dei risultati per descrivere meccanismi di rottura; alcuni si diversificano anche per le deformabilità assegnate ai vari elementi. Secondo i principali modelli a telaio equivalente, pilastri e travi avranno deformabilità assiale e taglio, mentre gli offset, saranno bracci rigidi in cui maschi e fasce di piano si incontrano. Questa schematizzazione deriva dall'osservazione diretta, in cui i meccanismi di rottura delle strutture in murature si concentrano sì nei maschi e nelle fasce di piano, ma mai nei punti di intersezione di questi, a meno di grossolani errori di realizzazione.

Tornando all'implementazione del modello a telaio equivalente, sarà, similmente a quanto definito per un modello agli elementi finiti, necessaria una *superentity* per descrivere gli oggetti del modello in questione, questa sarà **IfcEquivalentFrameModelItem**. Sarà del tipo *abstract* perché definita dai seguenti *subtypes*.

**-IfcEFMColumn:** è una schematizzazione monodimensionale. Sarà un *subtype* dotato di cinque *attributes* di cui tre sono opzionali. Infatti, in base al modello proposto si può o meno considerare la rigidità tagliente fuori dal piano e la rigidità flessionale nel o fuori dal piano. Si trascura praticamente in ogni caso la rigidità torsionale. Con gli *attributes* obbligatori si descrive la rigidità

---

72 - G. Magenes et alii, "Metodi semplificati per l'analisi sismica non lineare di edifici in muratura"

73 -G. Rizzano, R. Sabatino, G. Torello, "Un nuovo modello a telaio equivalente per l'analisi statica non lineare di pareti in muratura"

tagliante nel piano e quella assiale. In pratica gli *attributes opzionali* sono *InPlaneShearStiffness* e *AxialStiffness* collegano, rispettivamente *IfcLinearStiffnessMeasure* e *IfcModulusOfElasticityMeasure*. Difatti si considera sempre, in questi schemi di calcolo, la rigidezza come costante lungo la lunghezza e quindi un unico valore per tutto lo sviluppo della colonna. Così facendo le rigidezze tagliante e assiale avranno la corretta unità di misura. Allo stesso modo, i tre *optional attributes* sono *OutOfPlaneShearStiffness*, *InPlaneBendingStiffness* e *OutOfPlaneBendingStiffness*, il primo che riporta, nuovamente ad *IfcLinearStiffnessMeasure*, mentre gli altri due ad *IfcRotationalStiffnessMeasure* per la stessa motivazione degli *attributes* precedenti. Così facendo si vuole descrivere compiutamente la schematizzazione di un elemento monodimensionale che rappresenta una parete a taglio realizzata in muratura.

Si è voluto mantenere topologicamente separate colonne e travi perché sia possibile descrivere elementi travi il cui comportamento sia influenzato o governato dalla presenza di cordoli in calcestruzzo armato, in muratura armata, o in acciaio;

-**IfcEFMBeam**: similmente ad *IfcEFMPillar*, saranno elementi monodimensionali a cui si riportano deformabilità assiale e a taglio nel piano per descrivere il comportamento meccanico della fascia di piano. Si riporteranno, volendo, anche deformabilità a taglio, fuori dal piano, e deformabilità flessionale nelle due direzioni principali;

-**IfcEFMOffset** rigidi serve a descrivere i bracci rigidi in cui si intersecano pilastri e colonne. Essendo sicuramente rigidi, infatti, non avranno bisogno di particolari *attributes* per descriverne il comportamento, come invece succedeva per i precedenti.

Infine, l'*entity* **IfcEquivalentFrameModel** raccoglie tutti gli elementi del modello a telaio equivalenti modellato, attraverso l'*attribute Element* collegato ad *IfcEquivalentFrameModelItem*. L'*attribute OrientationOf2DPlane* diretto verso *IfcAxis2Placement3D* definisce invece l'orientamento del modello rispetto al sistema di coordinate generali impostato per il progetto complessivo.

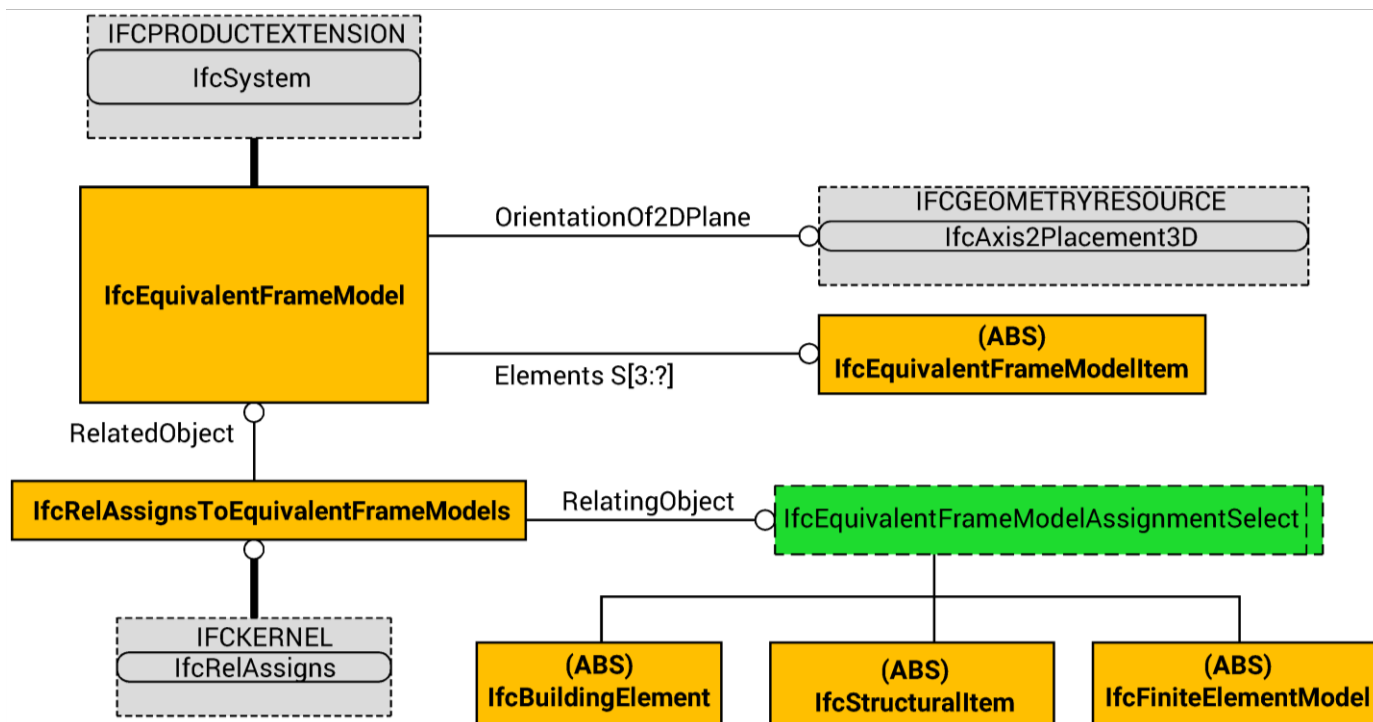


Figura 75 - Proposta modifica. Relazionalità modello a telaio equivalente.

Similmente a quanto definito per il modello agli elementi finiti, si definisce quindi la relazionalità del modello a telaio equivalente all'interno del progetto. Sarà implementata un'entity **IfcRelAssignsToEquivalentFrameModel** che avrà due *attributes*, il primo sarà *RelatedObject* diretto verso *IfcEquivalentFrameModel* che permette così di individuare il modello a telaio equivalente scelto che si sta mettendo in relazione, il secondo sarà *RelatingObject* diretto verso **IfcEquivalentFrameModelAssignmentSelect**. Questo *select type* permette di individuare, invece, a quale elemento del progetto il modello a telaio equivalente sia riferito. Sarà possibile mettere in relazione il modello analitico, scegliendo *IfcStructuralItem*, oppure direttamente con l'architettonico attraverso *IfcBuildingElement*, infine con il modello agli elementi finiti nel caso in cui questo costituisca il modello a telaio equivalente, attraverso *IfcFiniteElementModel*.

A questo punto, perché ciò sia completo, sarà necessario che il modello a telaio equivalenti possa accogliere i risultati delle analisi che vi sono condotte, sarà necessario modellare delle *classes* che possano fare questo.

# **9. validazione**

Come si anticipava nel capitolo “metodologia” si è poi voluto a valutare un programma di analisi strutturali sulla base della sua capacità di interfacciarsi con IFC e viceversa, nell’ottica di poter validare quanto già presente nello standard e quanto si propone come estensione dello stesso. Ovvero come, all’atto pratico, le *Industry Foundation Classes* rispondano e possano rispondere alle necessità di scambio informativo di un software FEM. La scelta è, appunto, ricaduta su TNO Diana, un software agli elementi finiti specializzato in analisi dinamiche, sia perché un buon solutore agli elementi finiti, sia perché uno dei pochi software capace di confrontarsi con lo standard buildingSMART in ingresso. Questo permette una discreta, seppur ancora molto limitata rispetto alle possibilità sin ora evidenziate, interoperabilità tra programmi FEM e programmi BIM. L’idea è di poter valutare come le informazioni relative all’ambito strutturale viaggino e siano mantenute, analizzando criticamente i risultati ottenuti. Con la preziosa collaborazione di Softing, sono state condotte le stesse analisi in Nòlian e confrontate. Sulla base di ciò che è stato possibile effettuare con questi software e sulla base dell’attività di normazione svolta sono, infine, state stilate le conclusioni.

## **DATI IN INGRESSO**

Il caso studio particolare su cui si vuole sperimentare tutto ciò è l’edificio denominato “Ex-Fisica Tecnica”, dell’Università degli Studi di Padova, sede di aule, studi e laboratori dell’ateneo patavino, sito in via Marzolo 9. Di questo si dispone di un file di partenza ARCHICAD, piuttosto ben definito e completo. I vantaggi di questo software è l’alta conformità con lo standard IFC, sebbene non assoluta, e la gestione della sola disciplina architettonica. Questo permette di valutare al meglio un file in ingresso in un software strutturale, perché sicuramente completamente privo di ogni possibile informazione di ambito strutturale riguardo il modello; cosa che in Revit non è garantita, visto la possibilità di questo di gestire anche l’ambito strutturale. Dopo un controllo preliminare approfondito del file in nostro possesso, in cui si verificava la corretta modellazione dell’edificio, la corrispondenza con i dati in nostro possesso, nonché le varie informazioni inserite e quelle inseribili, si è proceduto ad esportare in formato “.ifc” il modello dell’Ex-Fisica Tecnica. Un breve controllo attraverso il software “Solibri Model Checker v9.6” ha permesso di verificare il mantenimento dei parametri inseriti e della geometria di partenza.

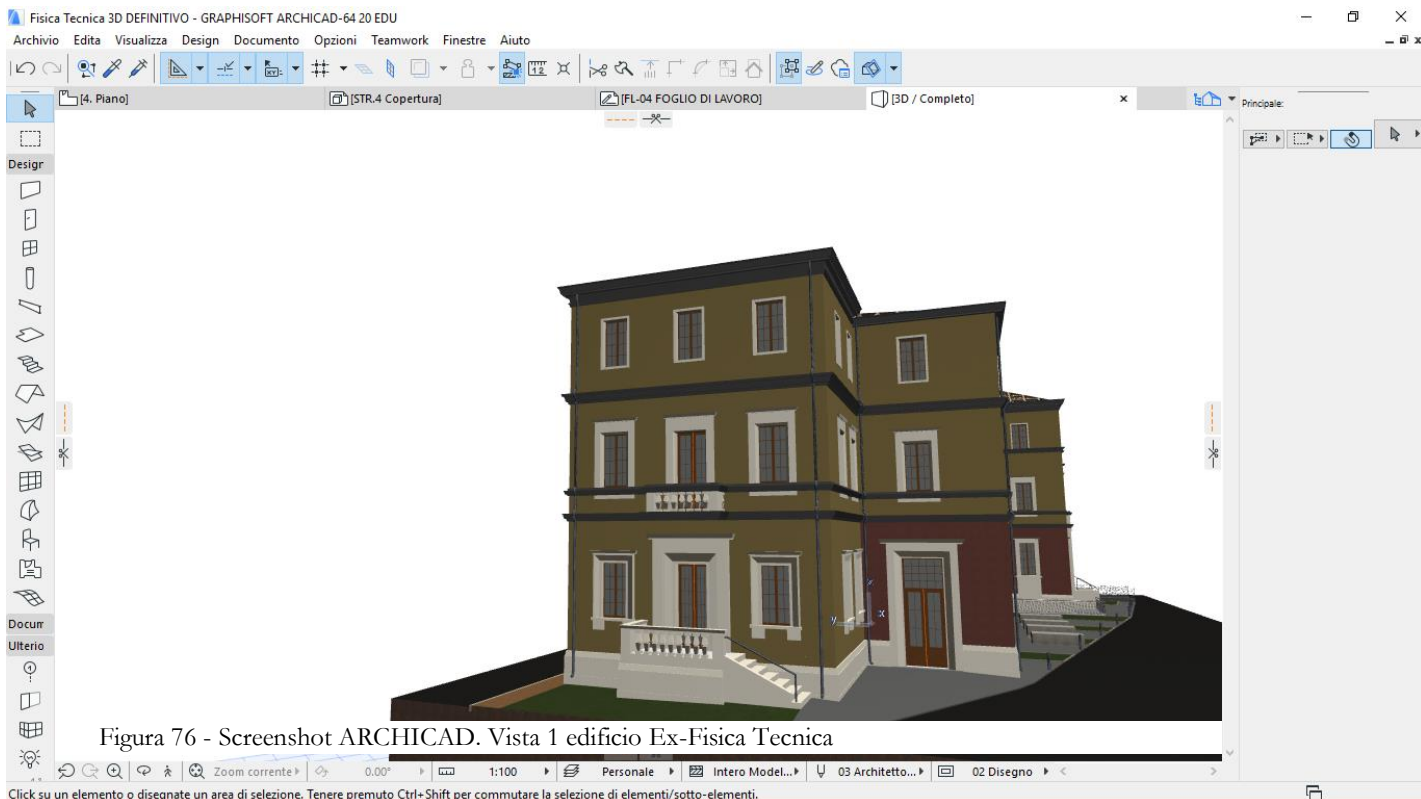


Figura 76 - Screenshot ARCHICAD. Vista 1 edificio Ex-Fisica Tecnica

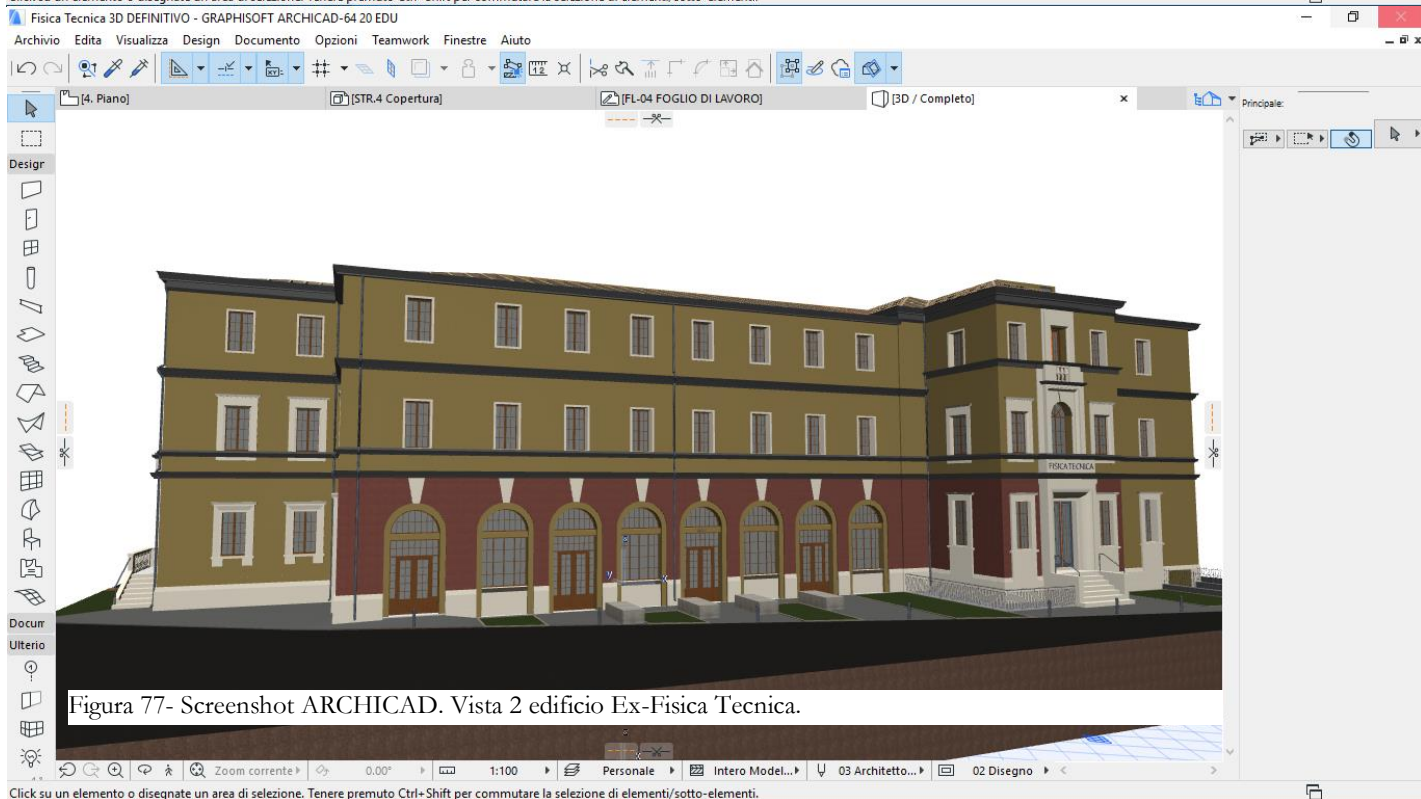


Figura 77- Screenshot ARCHICAD. Vista 2 edificio Ex-Fisica Tecnica.



Figura 79 - Screenshot Solibri Model Checker. Vista 1 edificio Ex-Fisica Tecnica.

### Takeoff

#### Info

Wall.0.75

Hyperlinks		ArchiCADProperties		ArchiCADQuantities		BaseQuantities	
Identification	Location	Quantities	Material	Profile	Relations	Classification	
Pset_ManufacturerTypeInformation				Pset_WallCommon			
Valori NTC							

Property	Value
Massima Accelerazione Orizzontale al Suolo Ag SLD	0.415
Massima Accelerazione Orizzontale al Suolo Ag SLV	0.927
Massimo Accelerazione Orizzontale al Suolo Ag SLO	0.345

Figura 78 - Screenshot Solibri Model Checker. Proprietà personalizzate in IFC.





Per curiosità si è provato ad ampliare lo sguardo e si è verificata la possibilità, come mostrato nelle immagini precedenti di esportare proprietà non ancora previste da IFC. Non solo, infatti, è stato possibile individuare i muri portanti e quelli non portanti, la loro data di realizzazione e altre informazioni base, ma si è anche potuto inserire, sebbene solamente come stringa di testo, i valori di accelerazione al suolo da considerare nella valutazione di vulnerabilità sismica dell'edificio, in base ai differenti Stati Limite considerati. Di per sé non è molto, poiché qualunque software strutturale non sarebbe in grado di recepire questo dato, però ci permette di capire che il formato IFC è un qualcosa dotato di già abbastanza flessibilità da poter essere uno strumento utile alla gestione di un intervento sul costruito la cui problematica principale è la gestione delle informazioni. Così facendo, sebbene solo come stringa di testo, lo si ribadisce, è possibile assegnare ad un singolo modello tutte le informazioni che gli sono proprie, utili e meno utili, che raccontano la storia dell'edificio, dalla sua creazione in poi, per tutto l'arco della sua vita utile, tutti gli interventi che vi sono stati effettuati e, molto di più, tutte le scelte progettuali che gli hanno conferito l'aspetto attuale, permettendo così di avere a portata di mano qualunque informazione possibile. Poiché ciò che riguarda il restauro e la messa in sicurezza dei luoghi storici è in sommo grado legato al grado di conoscenza che si ha dell'edificio e si capisce bene come lo strumento del *Building Information Modeling* possa essere assolutamente prezioso per questa disciplina.

Tornando a noi, si è provveduto poi a tentare di importare il file IFC del nostro edificio all'interno di TNO Diana, ma dopo svariati tentativi di importazione non andati a buon fine, i quali hanno tutti condotto al un crash del software, si è deciso di restringere la sperimentazione ad una singola parete significativa dell'edificio, creando un file IFC separato fisicamente e privo degli altri elementi del progetto, se non quelli utili a definire carichi e vincoli. La scelta è caduta sulla facciata Est dell'edificio, interessante perché inizialmente concepita con un portico a un piano, è stato poi ampliata in pianta e, in un secondo momento, persino in prospetto di due piani fuori terra. Un'analisi di questo elemento murario, anche solo dell'andamento tensionale, rimane indubbiamente interessante. Non si dimentichi, in ogni caso, che il nostro obiettivo è la validazione di un metodo, quindi ciò che è importante è che funzioni fin dai casi più semplici, altrimenti rimane inutile. Si è voluto quindi condurre una semplice analisi *pushover*, che permette di verificare come la struttura reagisce a un carico



statico equivalente ad un'azione sismica, un'analisi dinamica per la determinazione degli autovalori per i modi di vibrare della struttura. In ogni caso, poi, va ricordato che la filosofia BIM permette che le informazioni così ottenute dal singolo elemento strutturale siano comunque archiviabili all'interno dello stesso file IFC.

Si è quindi ricostruita la geometria semplificata della parete in esame e la si è, di nuovo, esportata in IFC. Dopo il controllo preliminare attraverso Solibri Model Checker si è potuto importare il modello all'interno di TNO Diana.

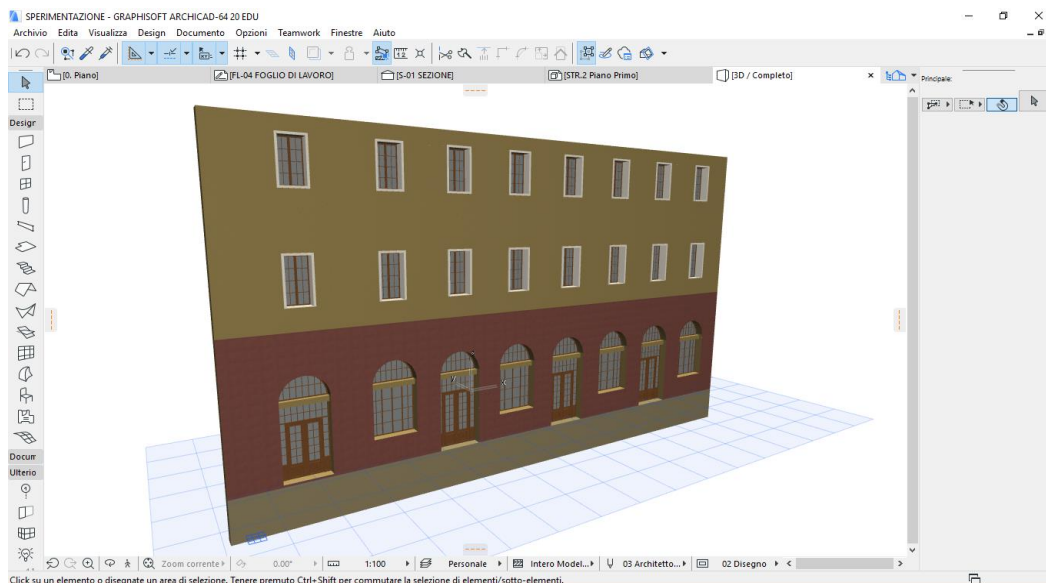


Figura 80 - Screenshot ARCHICAD. Vista muro isolato facciata est.

Figura 81 - Screenshot Solibri Model Checker. Vista muro isolato facciata est.

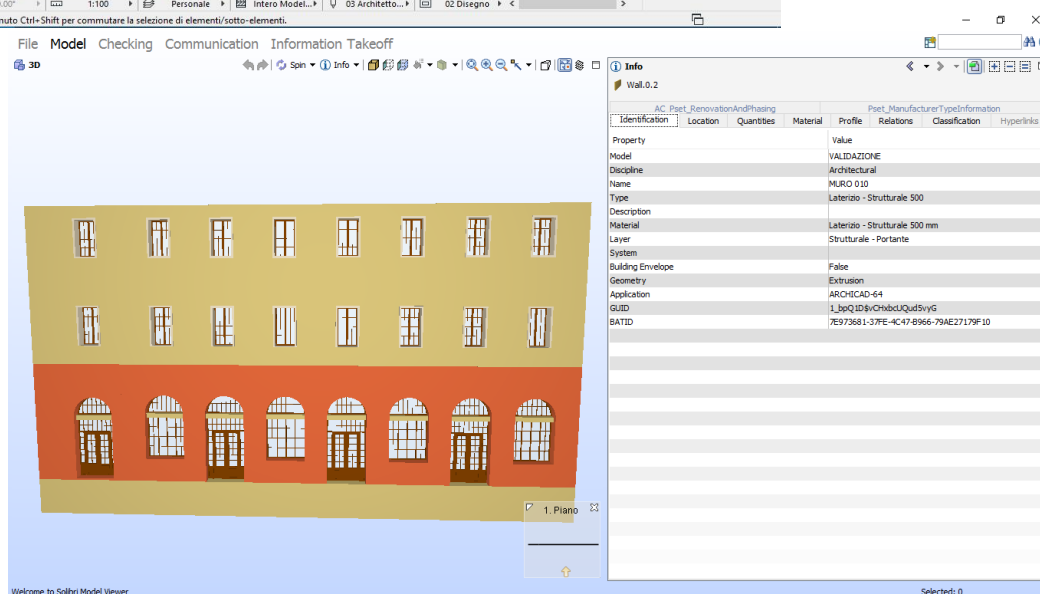


Figura 82 - Screenshot pannello importativo TNO Diana.

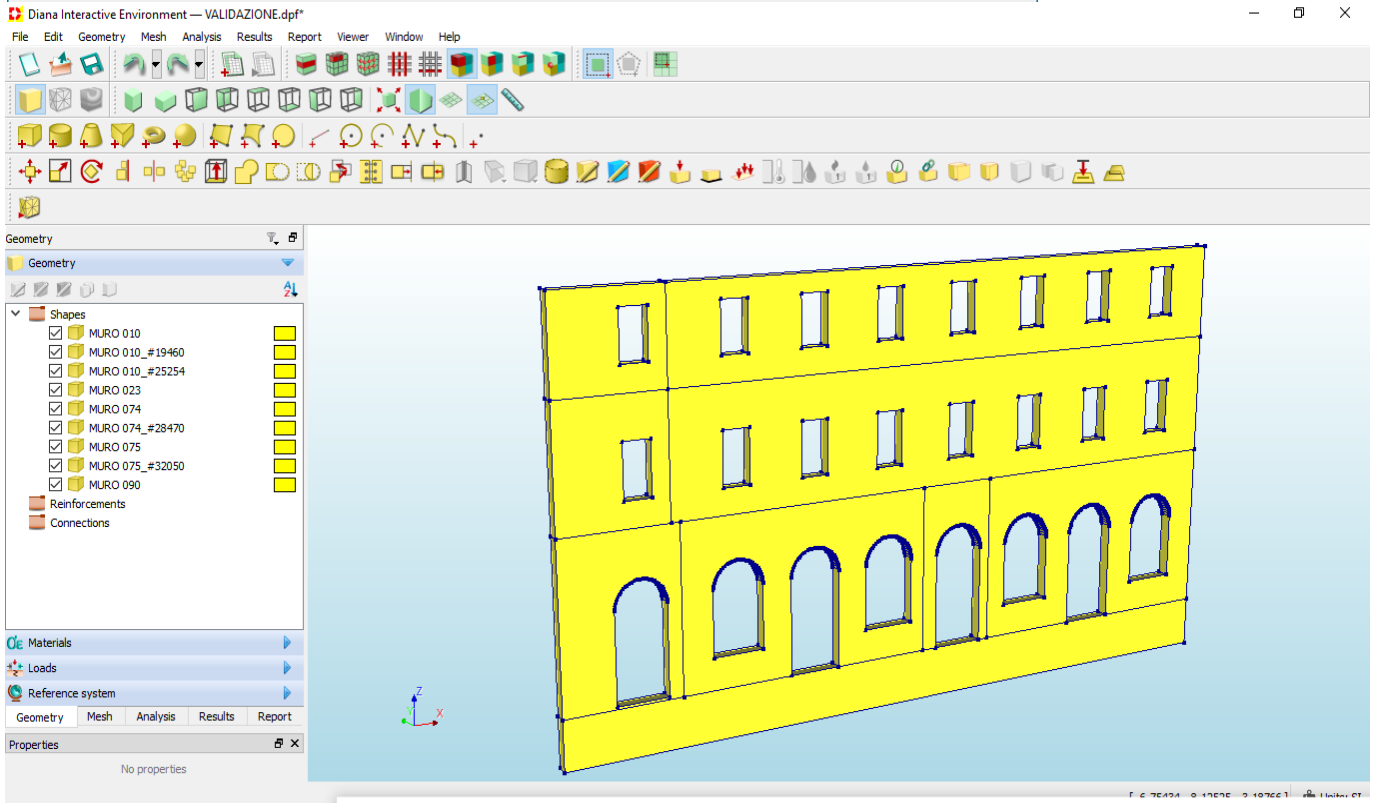
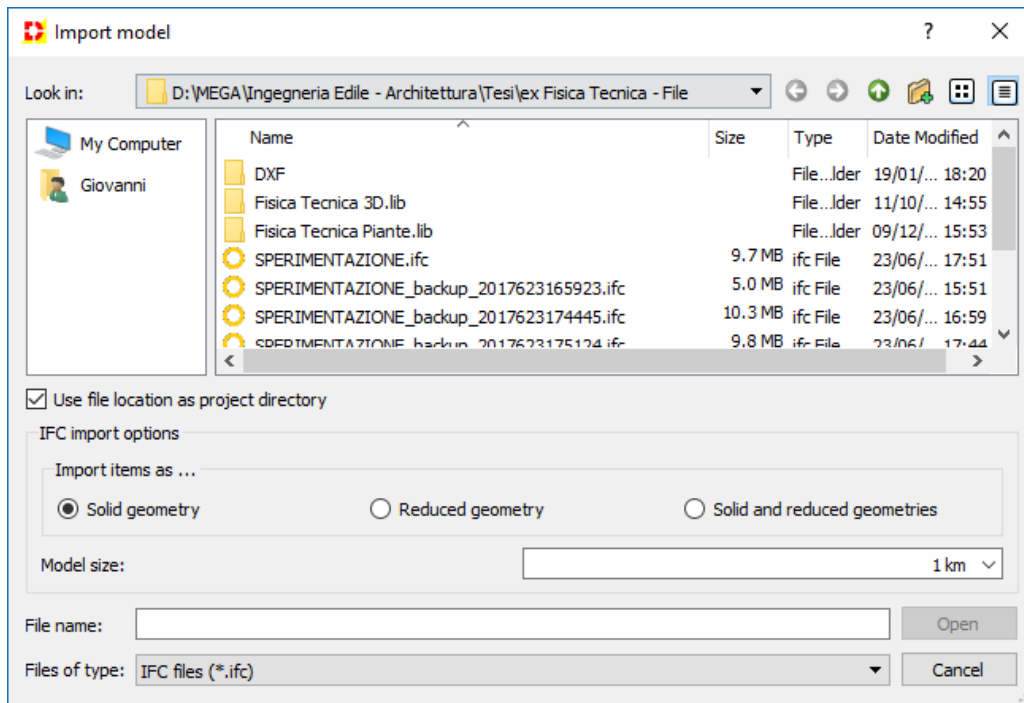


Figura 83 - Screenshot TNO Diana. Muro solido importato.

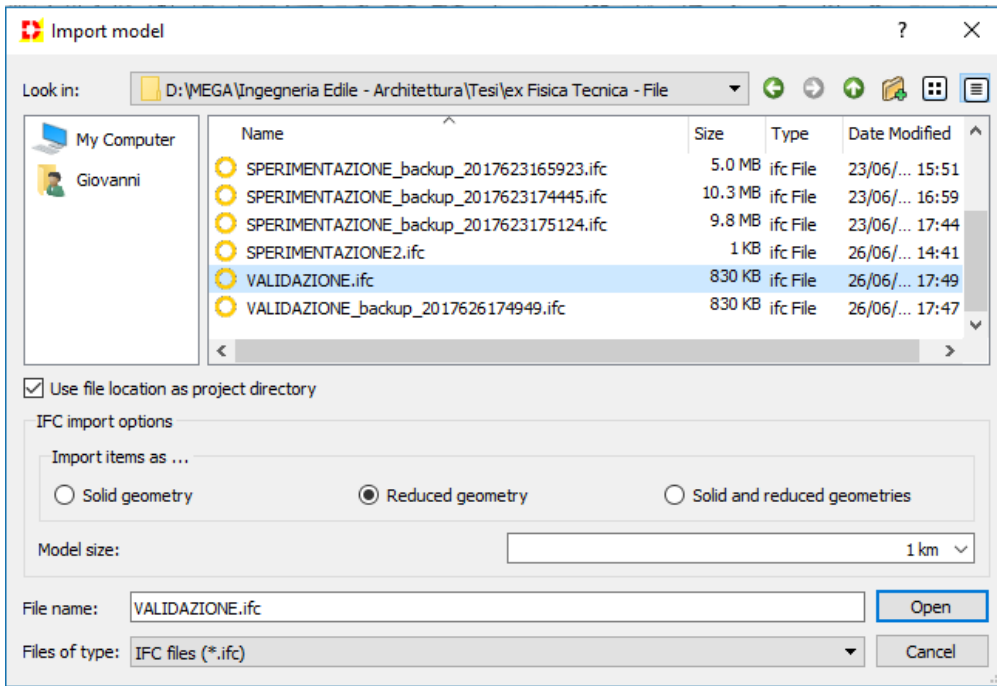


Figura 84- Screenshot pannello importativo TNO Diana.

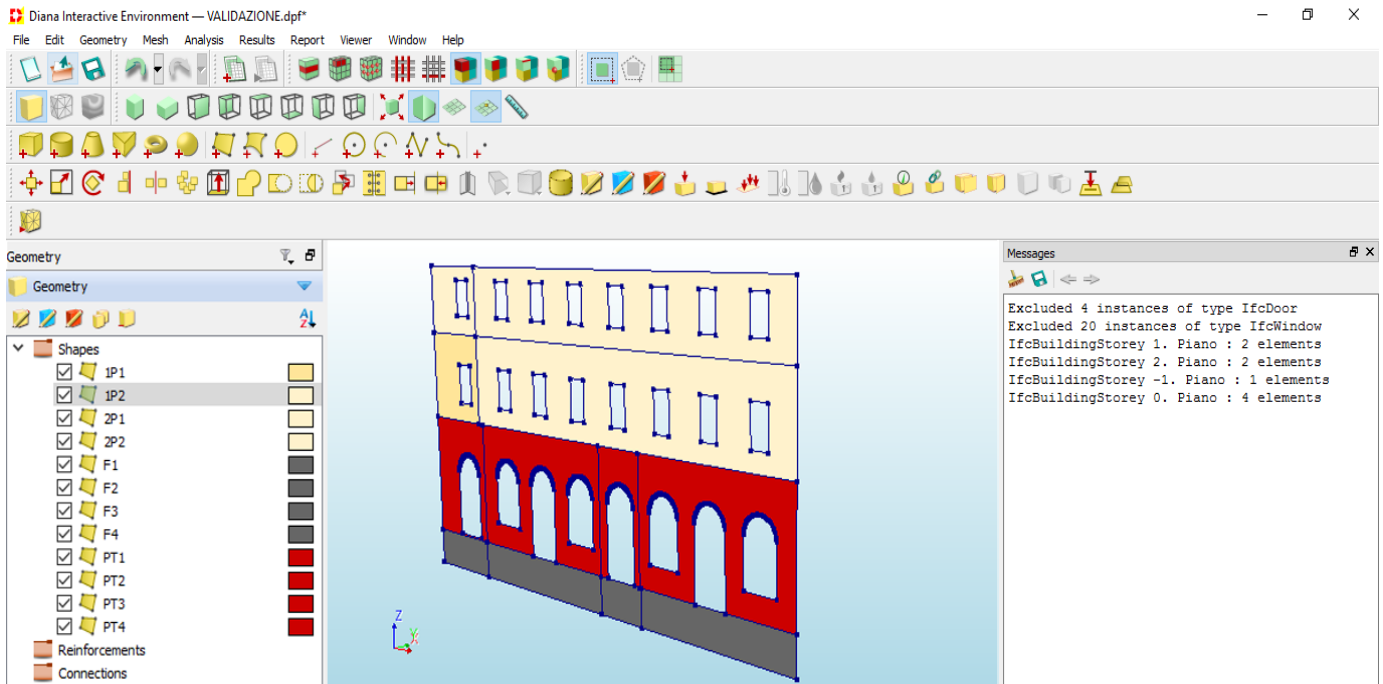
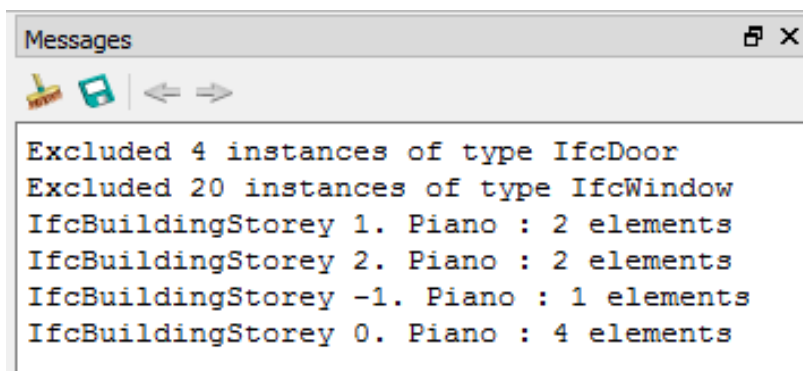


Figura 85 - Screenshot TNO Diana. Muro bidimensionali importato.

Questa volta, in breve tempo, è stato possibile importare il file IFC senza particolari problemi all'interno dell'ambiente Diana, che permette due possibilità di scelta all'importazione, più una terza che le combina: ossia la possibilità di importare la geometria solida del modello o la geometria "ridotta", oppure entrambe allo stesso tempo. Ci si è accorti di una cosa: il software TNO Diana quando si va a confrontare con i file IFC, ad esempio, lo fa sia con la Model View Definition di Coordination View 2.0, leggendo gli elementi fisici della struttura, quelli condivisi nello *Shared Element Domain*, forse anche quelli appartenenti allo *Structural Element Domain*. A questo punto dà questa possibilità di scelta che si traduce nel confrontarsi con elementi analitici iniziali tridimensionali nel caso si importi la geometria solida, oppure con elementi analitici mono e bidimensionali, nel caso in cui si importi la geometria "ridotta". In pratica il software provvede ad attivare il modello analitico della struttura automaticamente, basandosi sulla geometria che ha a disposizione. Ciò è evidente poiché, importando una geometria e non un modello analitico, ci si ritrova una modellazione di tutto ciò che abbiamo importato come elementi solidi. Risulta allora evidente il perché il software crashasse: il gestire una geometria, per quanto semplice, molto ricca, come soli elementi solidi, è un qualcosa che richiedono una capacità computazionale decisamente elevata, soprattutto se è richiesto che sia il software, come in questo caso, a formare il modello automaticamente sulla base di quanto riceve in input, per di più, se possibile, assegnando preliminarmente anche un comportamento e dei gradi di libertà ai vari elementi e, di più, cercando di assicurare la congruenza. Anche nel caso in cui ci si ritrovi a importare una geometria ridotta il problema sussiste: si tratta di schematizzare analiticamente un numero molto elevato di elementi, creando non poche difficoltà.

Figura 86 - Screenshot TNO Diana. Report importazione.



```
Messages
Excluded 4 instances of type IfcDoor
Excluded 20 instances of type IfcWindow
IfcBuildingStorey 1. Piano : 2 elements
IfcBuildingStorey 2. Piano : 2 elements
IfcBuildingStorey -1. Piano : 1 elements
IfcBuildingStorey 0. Piano : 4 elements
```

Superato questo problema e importato il modello dell'elemento murario di interesse, si va a verificare come sia la situazione generale. Innanzitutto, con il controllo di possibili messaggi d'errore, poi si continua controllando quelli che sono i le geometrie e i materiali. Nei messaggi viene,

correttissimamente, riportato il numero di oggetti IFC letti, quelli esclusi e quelli considerati e le relative tipologie. È fin troppo evidente come un programma agli elementi finiti, in genere, non svolga analisi strutturali su porte e finestre. Verificando le geometrie, queste sono correttamente schematizzate dal software, complice una modellazione preliminare all'interno di ARCHICAD molto pulita, corretta e non ridondante. Come da più rosee aspettative è possibile assegnare sin da subito a questi elementi proprietà quali il tipo di elemento finito che si utilizzerà, la sua geometria, la

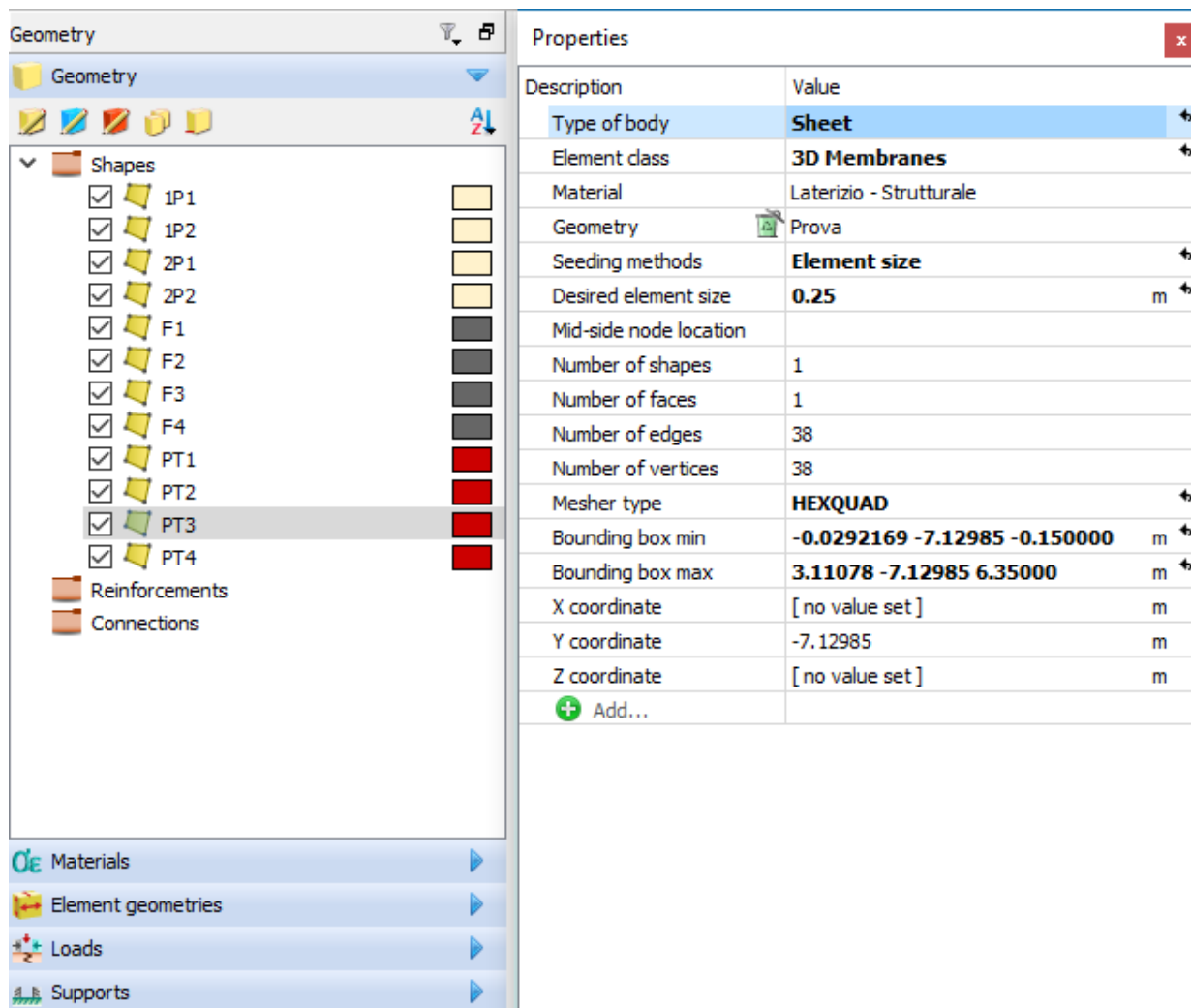


Figura 87 – Screenshot TNO Diana. Proprietà elementi analitici.

suddivisione e il metodo di suddivisione. Si provvede quindi a farlo. Si è scelto una superficie che agisce nel tridimensionale per valutare le analisi dinamiche, in modo che sia rivelato, almeno

parzialmente, il comportamento deformativo fuori dal piano della muratura. Nel caso in cui fossero state condotte analisi esclusivamente statiche lineari sarebbe ovviamente stato sufficiente l'elemento *shell*.

Per quanto riguarda i materiali, anche se precedentemente assegnati, all'interno di ARCHICAD, rimane solamente un nome di questi: ogni caratterizzazione è da effettuare, come ci si aspetta. Il software recepisce correttamente le *entities* che esprimono i materiali, ma, essendo questi stati definiti in ambito architettonico e non strutturale, questi non presentano nessuna informazione a riguardo del loro comportamento meccanico. Ovviamente ciò che riguarda i materiali è l'unico aspetto che ci si aspetta possa essere già implementato in automatico da TNO Diana, difatti ogni caratterizzazione riguardo la definizione dei vincoli, dei carichi, la suddivisione della mesh degli elementi è, giustamente, completamente assente. Sarebbe, però, possibile che la caratterizzazione dei materiali fosse già

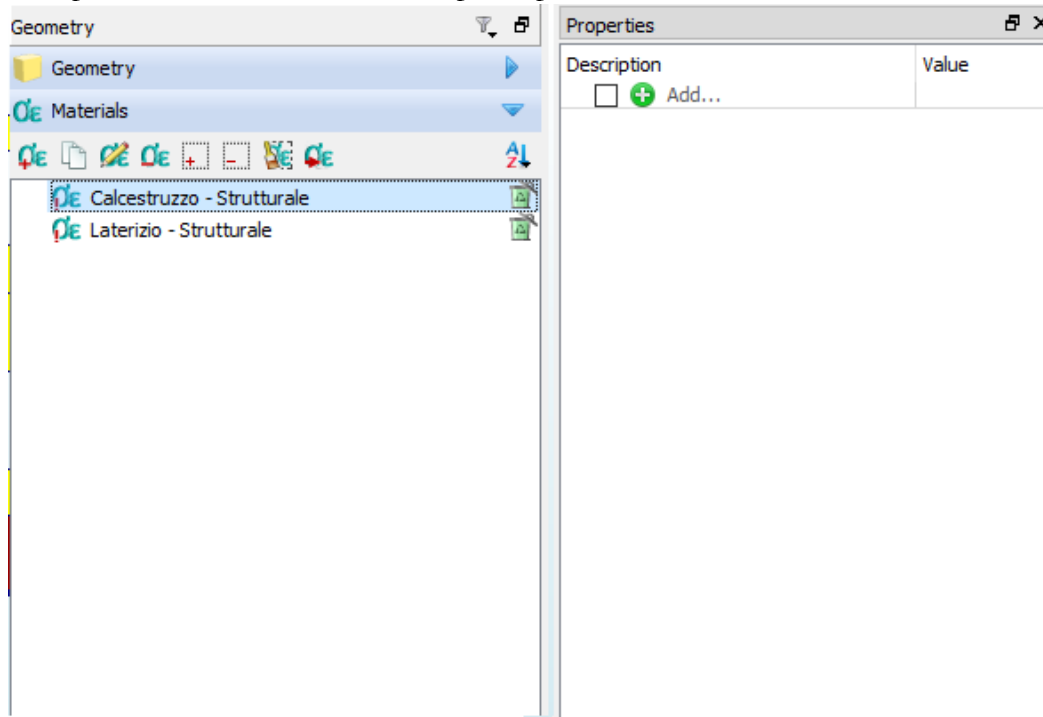


Figura 88 - Screenshot TNO Diana. Caratterizzazione materiali.

presente all'interno di un software FEM: se entrambi i programmi facessero riferimento correttamente ad IFC potrebbero infatti, entrambi attingere a

un'informazione standardizzata anche per quanto riguarda la libreria dei materiali. Questa ovviamente ricalca altri standard, che si occupano, appunto, dei materiali costitutivi, permettendo all'uno e all'altro

un'informazione già precisa ma ancora accessibile e modificabile. Vista la situazione sono stati caratterizzati i materiali basandosi sulle Norme Tecniche per le Costruzioni e la loro Circolare applicativa, soprattutto per quanto riguarda la muratura portante. Le ipotesi fatte sono basate sul lavoro di indagine già fatto, contemporaneamente alla modellazione ARCHICAD dell'edificio. Si è

Figura 89 - Screenshot TNO Diana. Caratterizzazione muratura.

Description	Value
Aspect	[ ]
Linear material properties	
Elasti	
Young's modulus	2250000 kN/m <sup>2</sup>
Poisson's ratio	0.33000000000000002
Mass	
Mass density	1.8 T/m <sup>3</sup>
Total strain based crack model	
Crack orientation	Fixed
Crack bandwidth specification	Rots
Tensile behaviour	
Tensile curve	Linear-ultimate crack strain
Tensile strength	0.16 kN/m <sup>2</sup>
Ultimate strain	0.004000000000000001
Residual tensile strength	0 kN/m <sup>2</sup>
Poisson's ratio reduction	
Reduction model	No reduction
Compressive behaviour	
Compression curve	Elastic
Shear behaviour	
Shear retention function	Constant
Shear retention	0.01

ipotizzata una muratura in mattoni pieni e malta di calce, con livello di conoscenza LC1 ed FC=1.35, con i

Name: Calcestruzzo - Strutturale

Aspects to include:

- Total Strain crack model
- Shrinkage
- Rayleigh damping
- Creep
- Heat flow

Eurocode 2 EN1992-1-1

Concrete type: Normal weight

Concrete class\*: C16/20

Aggregate type: Sandstone

Cement type: Class N

Material safety factors (ULS)

seguenti:  $f_m=240$  N/cm<sup>2</sup>;  $t_0=6$  N/cm<sup>2</sup>;  $E=1,5$  kN/mm<sup>2</sup>;  $G=500$  N/mm<sup>2</sup>; dotata di connessioni trasversali e malta buona. Il risultato è una muratura con  $E=2250$  N/mm<sup>2</sup>;  $G=750$  N/mm<sup>2</sup>; peso specifico=kN/m<sup>3</sup>;  $f_m=468$  N/cm<sup>2</sup>;  $\tau=11.7$  N/cm<sup>2</sup>;  $f_k=242.67$  N/cm<sup>2</sup>; Drift di Taglio=0.4%; Drift da Flessione=0.6%. Per

Figura 90 - Screenshot TNO Diana. Caratterizzazione calcestruzzo.

quanto riguarda il muro di fondazione, a trave rovescia, la caratterizzazione è svolta in automatico da TNO Diana una volta ipotizzato un calcestruzzo di classe C16/20.



Il passo successivo, come sempre, è stato l'inserimento dei vari carichi e dei vincoli. Per quanto riguarda i primi, considerando anche i carichi strutturali e non portati dai solai sulla muratura, poiché questi non sono stati modellati per la semplificazione. Per cui i carichi propri gravanti sui muri ai vari piani dati dai solai sono, dal pianterreno a salire: 32,5, 30, 22,5 e 15 kN/m; mentre gli accidentali portati da questi sono, nello stesso ordine: 11,5, 11,5, 7,5 e 2 kN/m. Sono state considerate azioni

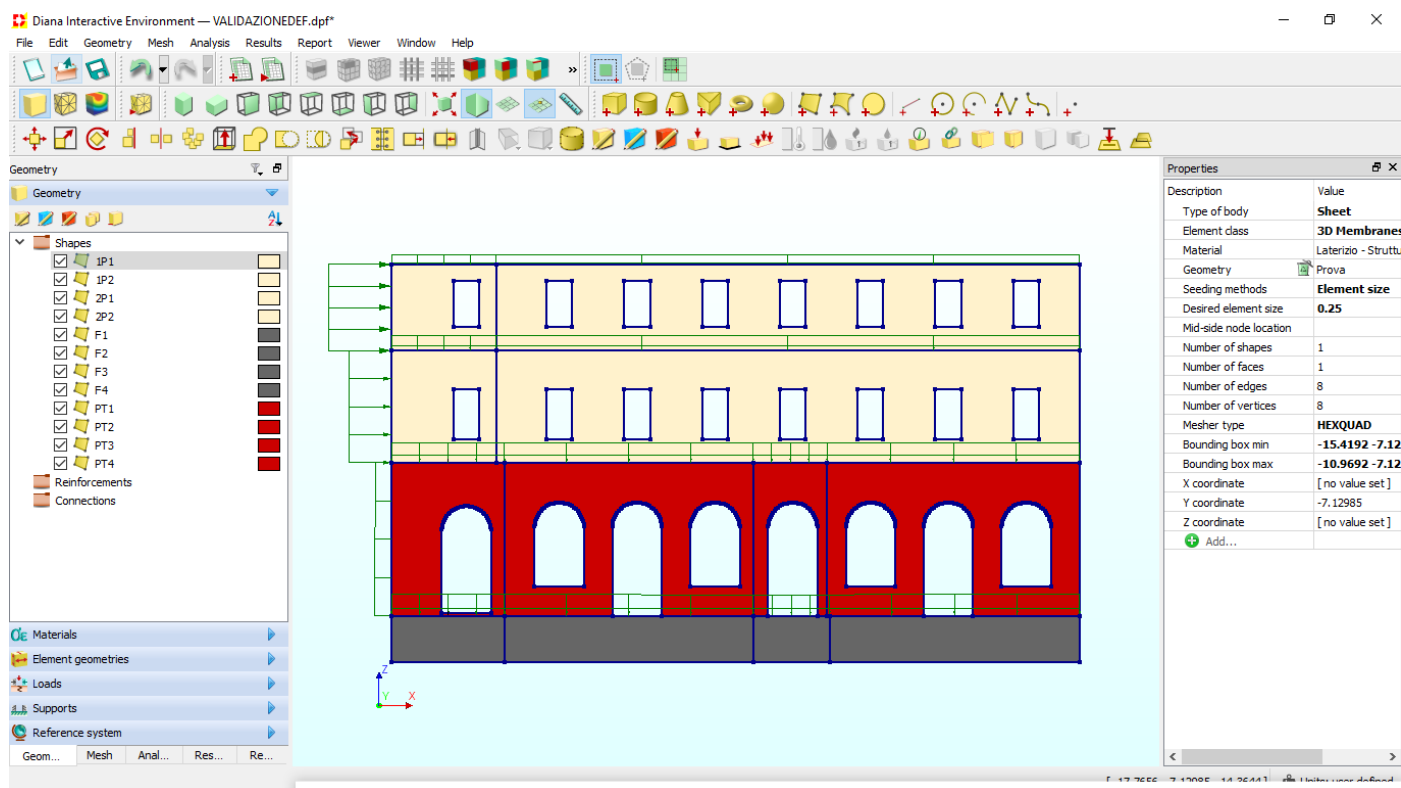


Figura 91 - Screenshot TNO Diana. Definizione dei carichi.

come da normativa per un'analisi statica lineare. Per semplicità sono stati modellati come carichi distribuiti, in modo che l'equivalente carico concentrato applicato nel baricentro abbia gli stessi effetti; questo perché TNO Diana non dà la possibilità di modellare un'azione concentrata in un punto diverso da un nodo. Questo può essere limitante, le stesse *Industry Foundation Classes* prevedono la possibilità di descrivere un carico puntuale posizionato su una curva o una superficie. Comunque le azioni considerate sono di 155 kN al baricentro del primo livello, 300 kN al secondo, 340 kN al terzo. Sono stati modellati come carichi distribuiti dal valore di 25, 65 e 95 kN/m rispettivamente.

Conseguentemente, al momento di modellare i vincoli si è scelto di riportare l'ammorsamento tra le murature ortogonali con un vincolo monodirezionale che trattenesse la muratura in direzione ortogonale al suo piano di giacitura. Oltre a questo sono stati aggiunti i vincoli di fondazione, scegliendo però di mantenere la modellazione della trave rovescia di fondazione visto l'altezza complessiva considerevole pari 1.85 m.

Dopo tutto ciò si è provveduto a creare la mesh e a verificarne la correttezza, nonché la presenza di vincoli e carichi come definiti.

Una volta verificata la correttezza di questa sono state lanciate le analisi: un'analisi statica lineare, un'analisi dinamica per determinare l'autovalore del modo fondamentale e una seconda analisi dinamica per determinare la risposta del primo modo fondamentale.

Figura 93 - Screenshot TNO Diana. Definizione combinazioni di carico.

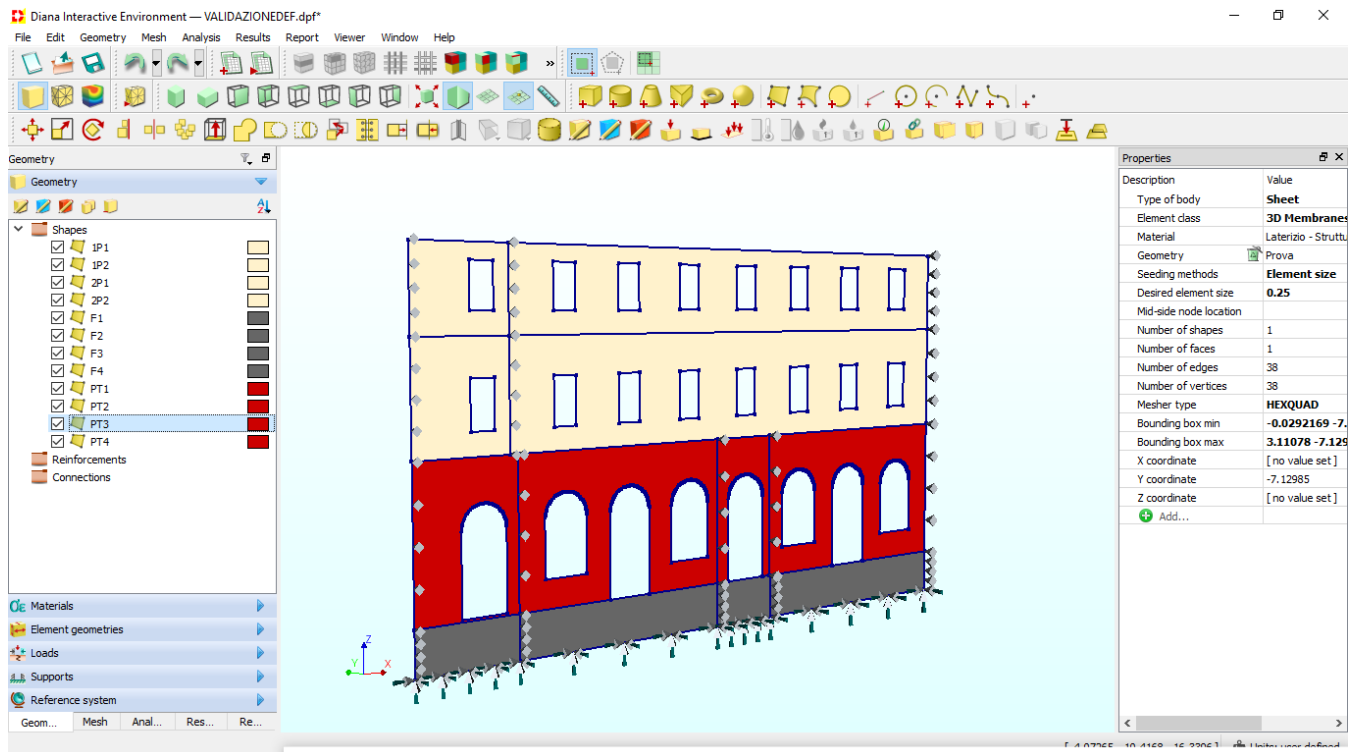
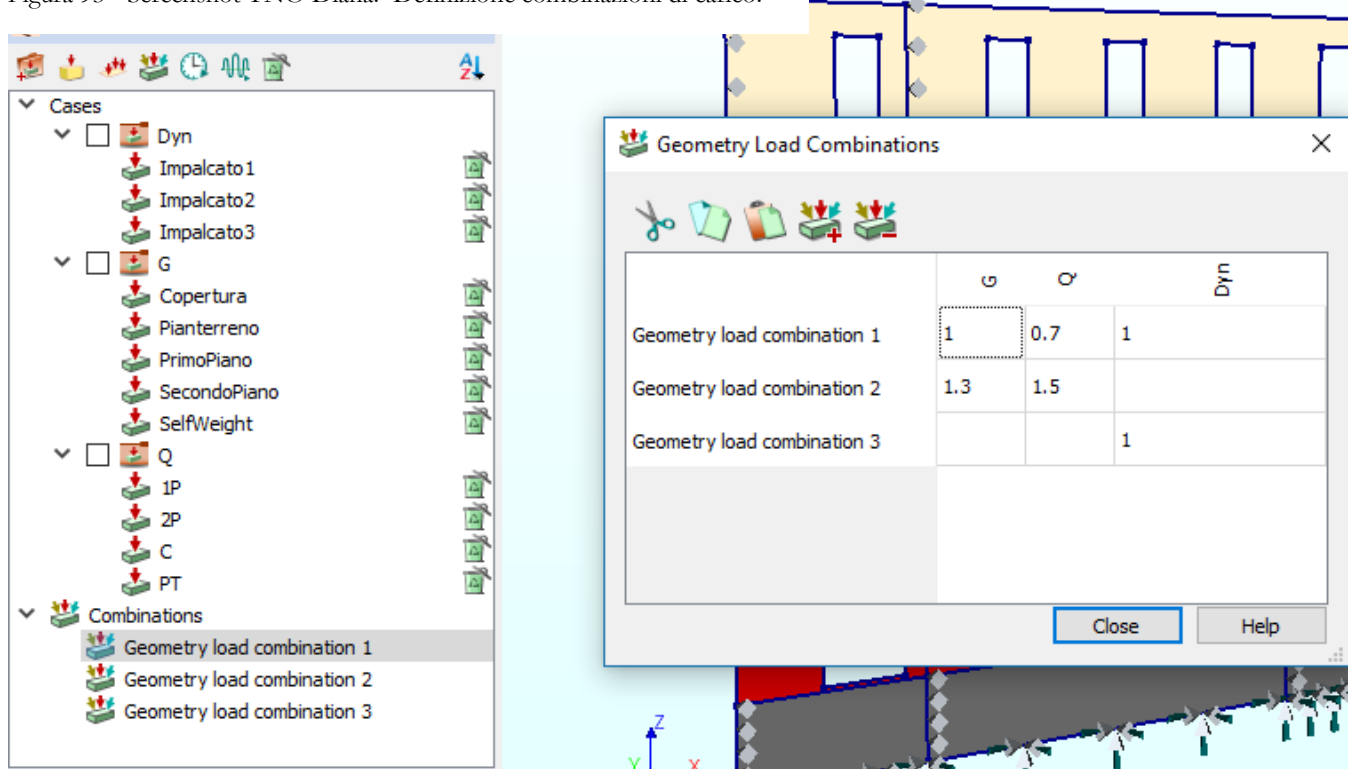


Figura 92 - Screenshot TNO Diana. Definizione condizioni di vincolo.

Figura 95 - Screenshot TNO Diana. Generazione della mesh.

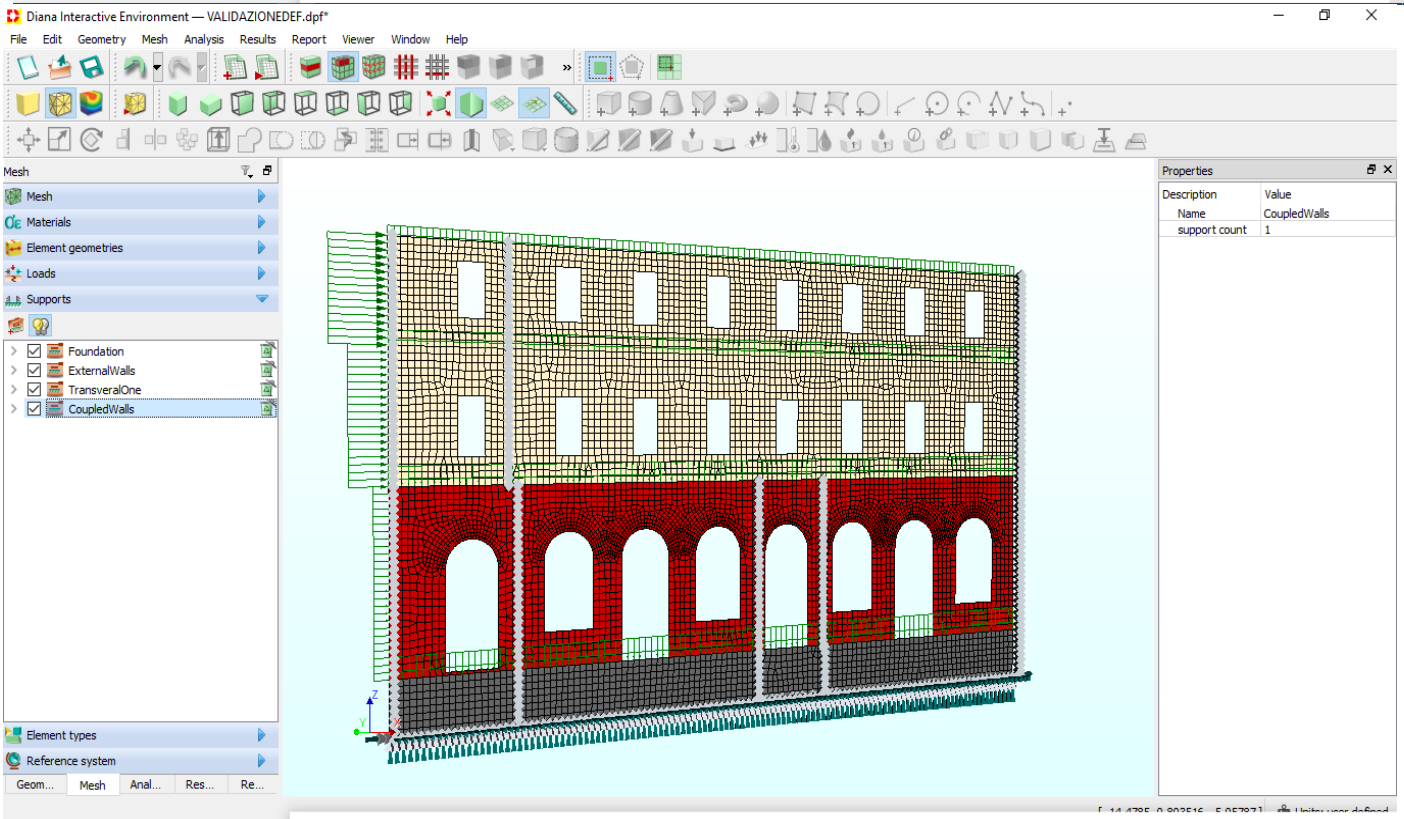
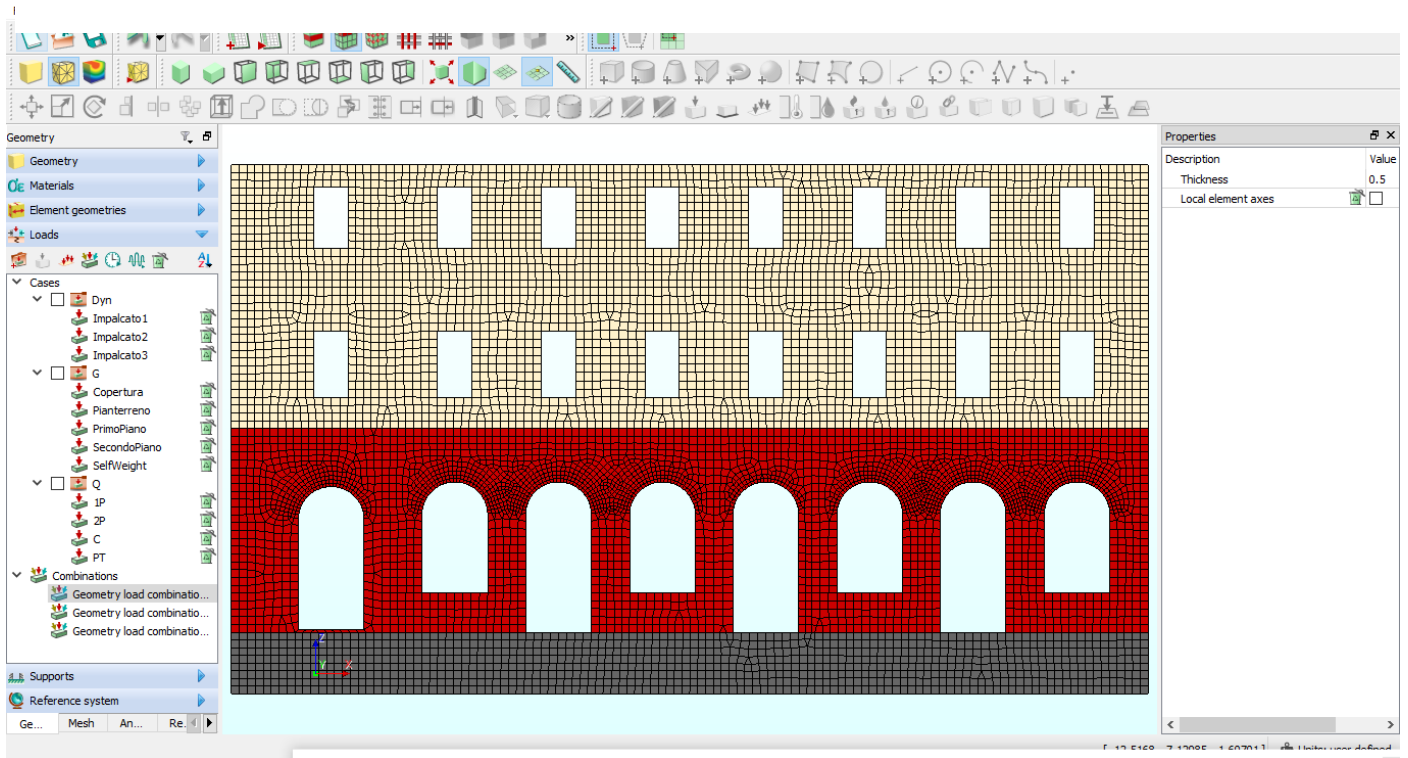


Figura 94 - Screenshot TNO Diana. Verifica carichi e vincoli nella mesh.

## DATI IN USCITA

Vengono di seguito presentati i risultati ottenuti. I primi che troviamo sono quelli relativi all'analisi statica lineare, nell'ordine per la combinazione di carico in caso di sisma, per la combinazione di carico classica allo Stato Limite Ultimo, infine, per i soli carichi statici equivalenti. La prima, appunto relativa alla combinazione pari a  $1*G+0.7*Q+1*Dyn$  è di seguito rappresentata nella sua configurazione deformata. Ottenuta correttamente, si guardano i valori degli spostamenti totali e delle deformazioni elastiche, sono anch'esse corrette e rivelano molto bene il comportamento classico per le strutture in muratura portante: le deformazioni elastiche mostrano un concentrazione di tensioni, visibili nel maggior valore delle deformazioni in corrispondenza dei restringimenti murari. Similmente le deformazioni a taglio si concentrano nei maschi murari e nelle fasce di piano. In entrambi i casi si mostra il classico andamento diagonale a formare il puntone resistente che scarica al suolo.

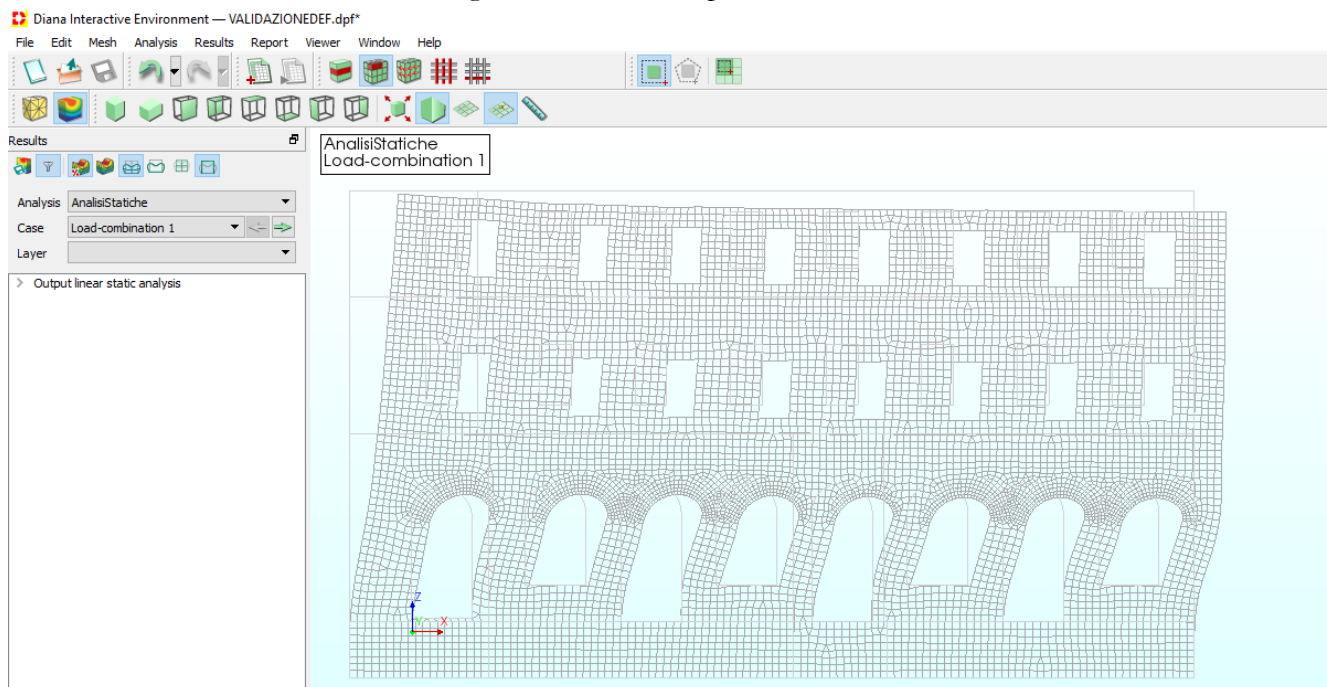


Figura 96 - Screenshot TNO Diana. Configurazione deformata da combinazione di carico n°1.

Results

Analysis: AnalisiStatiche

Case: Load-combination 1

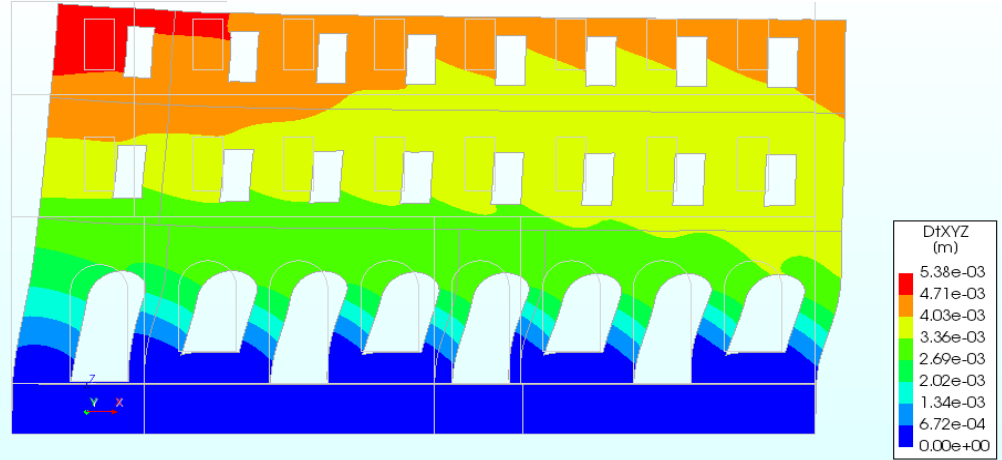
Layer: 1 of 3

Output linear static analysis

- Nodal results
  - Total Displacements
    - DX
    - DY
    - DZ
    - DtXYZ
  - External Forces
  - Reaction Forces
- Element results

AnalisiStatiche  
Load-combination 1  
Total Displacements DtXYZ

Figura 98 - Screenshot da TNO Diana. Spostamento totale da combinazione di carico n°1.



Results

Analysis: AnalisiStatiche

Case: Load-combination 1

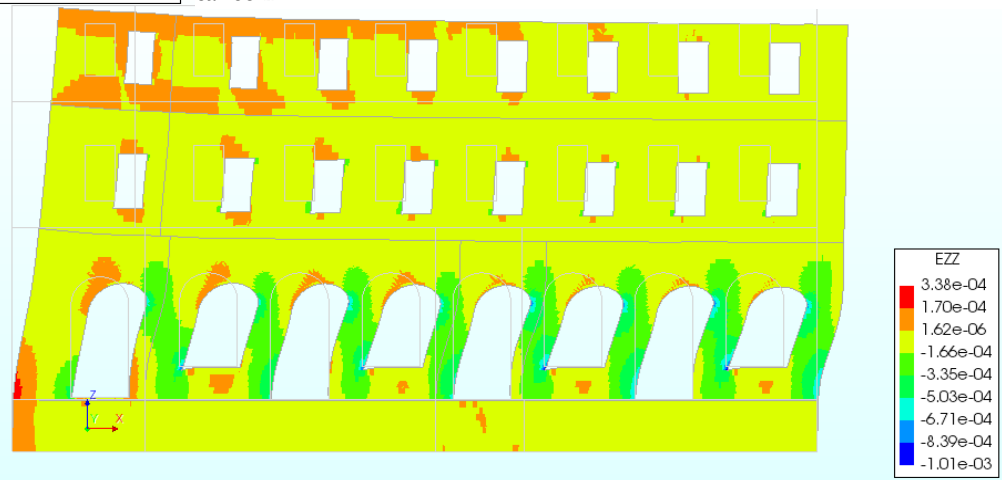
Layer: 1 of 3

Output linear static analysis

- Nodal results
  - Total Displacements
  - External Forces
  - Reaction Forces
- Element results
  - Total Strains
    - EXX
    - EYY
    - EZZ
    - GXY
    - GYZ
    - GZX
  - Cauchy Total Stresses
  - Distributed Forces
  - Distributed Moments

AnalisiStatiche  
Load-combination 1  
Total Strains EZZ layer 1

Figura 97 - Screenshot TNO Diana. Deformazioni verticali combinazione di carico n°1.



Results

Analysis: AnalisiStatiche

Case: Load-combination 1

Layer: 1 of 3

Output linear static analysis

- Nodal results
  - Total Displacements
  - External Forces
  - Reaction Forces
- Element results
  - Total Strains
    - EXX
    - EYY
    - EZZ
    - GXY
    - GYZ
    - GZX
  - Cauchy Total Stresses
  - Distributed Forces
  - Distributed Moments

AnalisiStatiche  
Load-combination 1  
Total Strains GZX layer 1

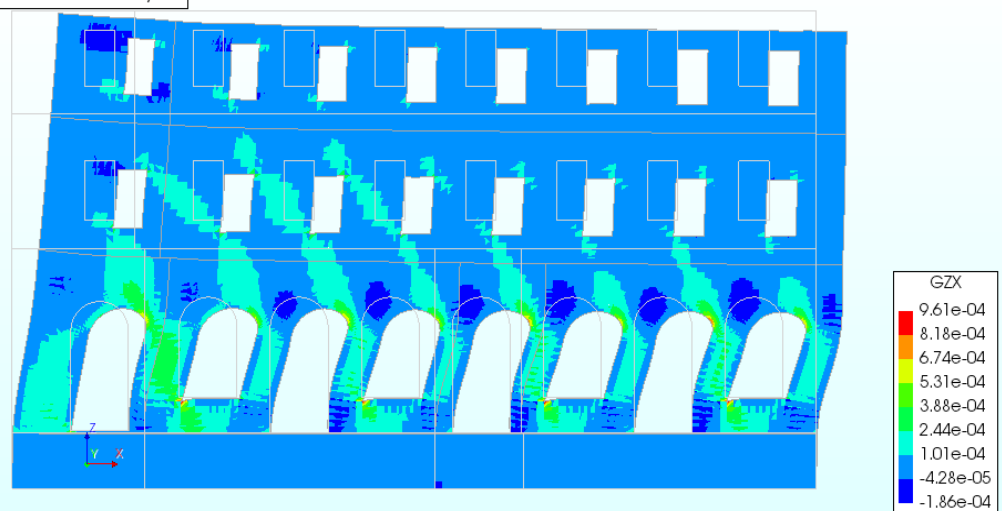


Figura 99 - Screenshot TNO Diana. Deformazioni orizzontali combinazione di carico n°1.

La seconda combinazione di carico statica lineare è quella classica per Stati Limite Ultimi per soli carichi verticali, data da  $1.3 \cdot G + 1.5 \cdot Q$ , ossia massimizzando le combinazioni sfavorevoli. Anche qui chiaramente, si mostra il massimo spostamento per instabilità nei maschi snelli e il concentramento di tensioni nei maschi più stretti.

Figura 100 - Screenshot TNO Diana. Configurazione deformata da combinazione di carico n°2.

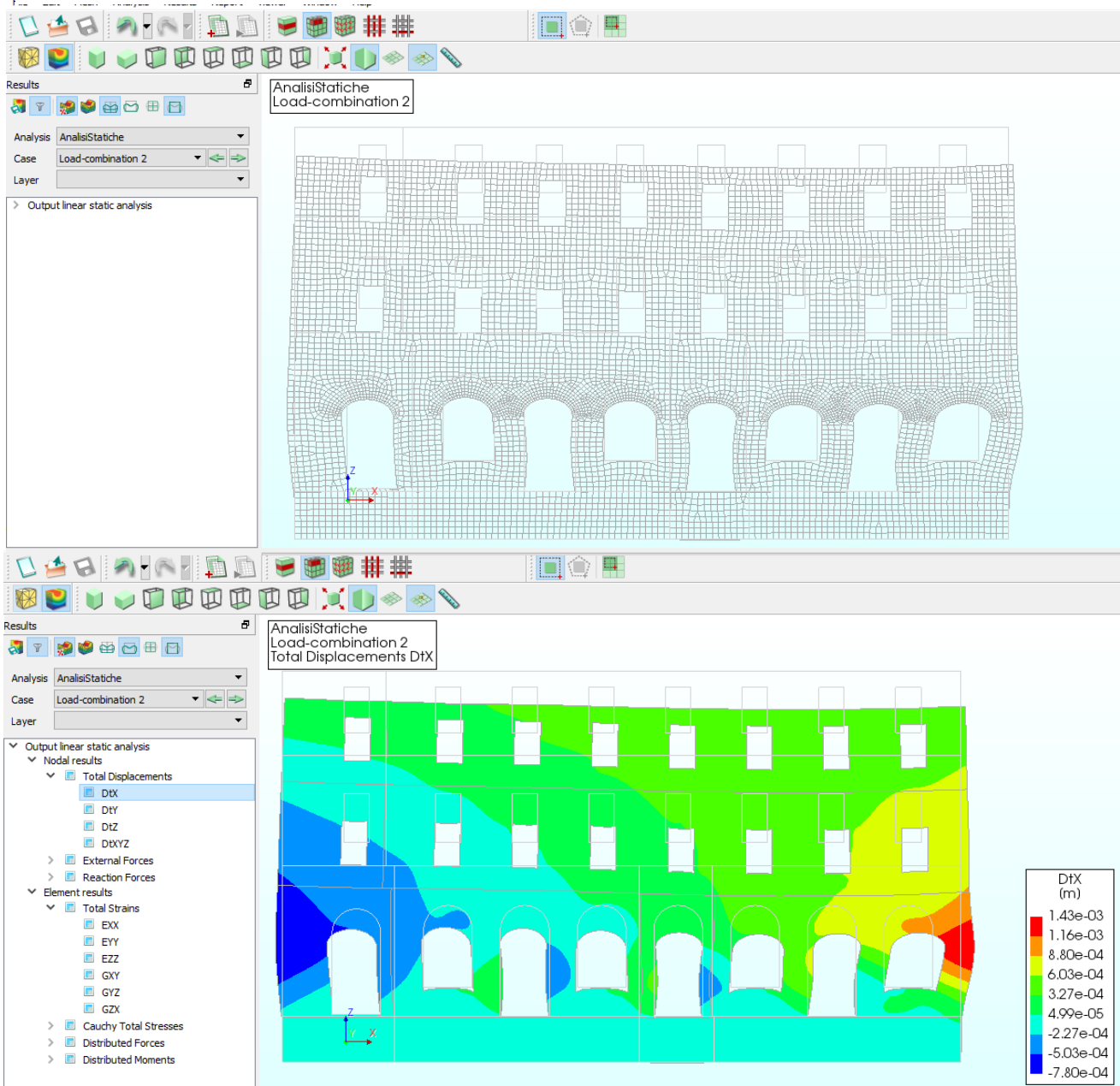


Figura 101 - Screenshot TNO Diana. Spostamento orizzontale da combinazione di carico n°2.

Figura 104 - Screenshot da TNO Diana. Spostamento verticale da combinazione di carico n° 2.

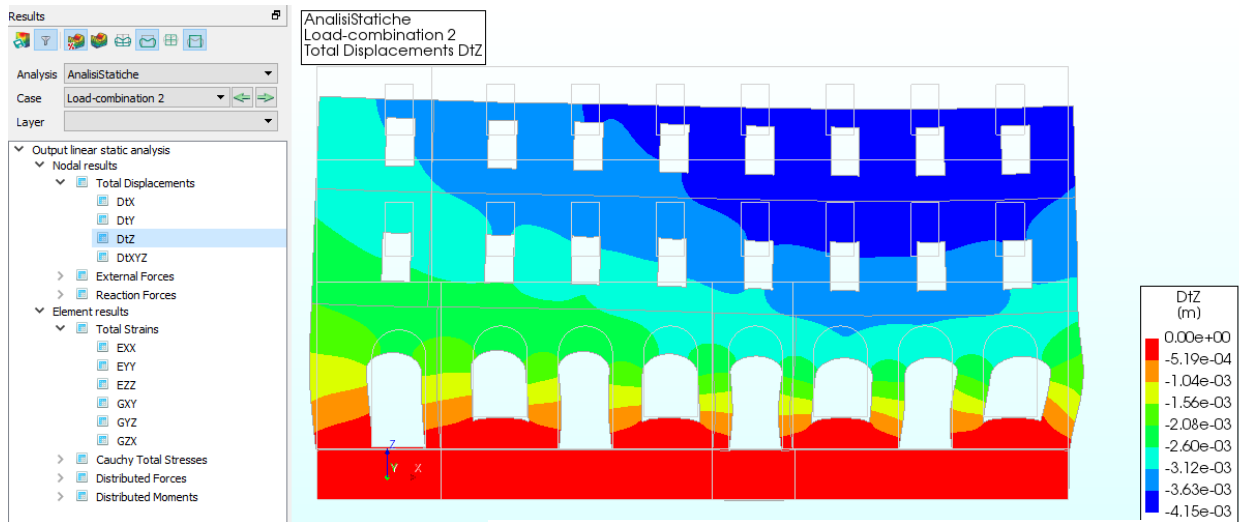


Figura 103 - Screenshot da TNO Diana. Deformazioni orizzontali da combinazione di carico n°2.

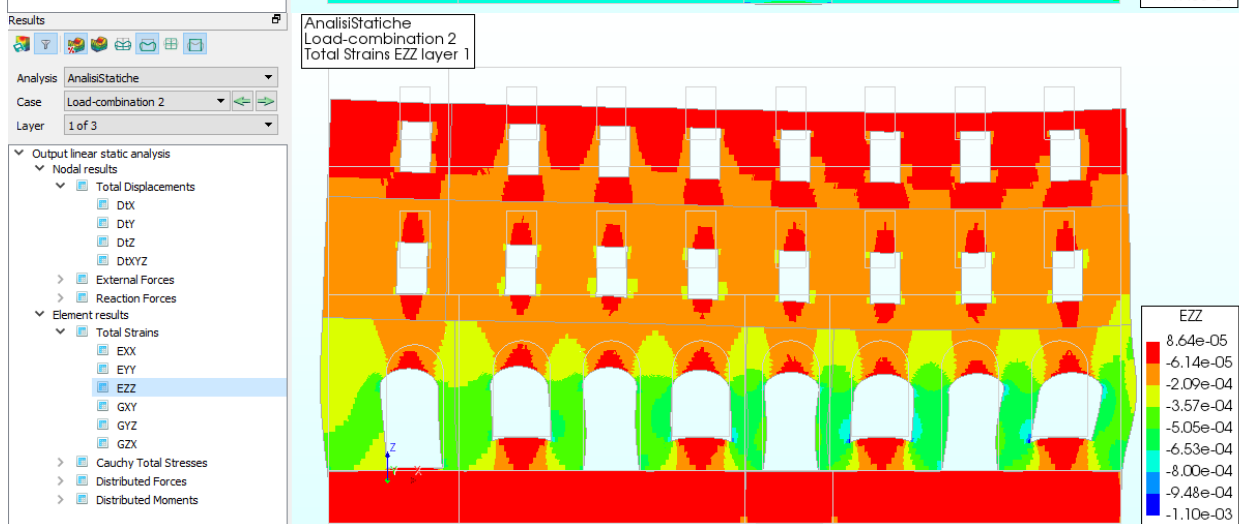
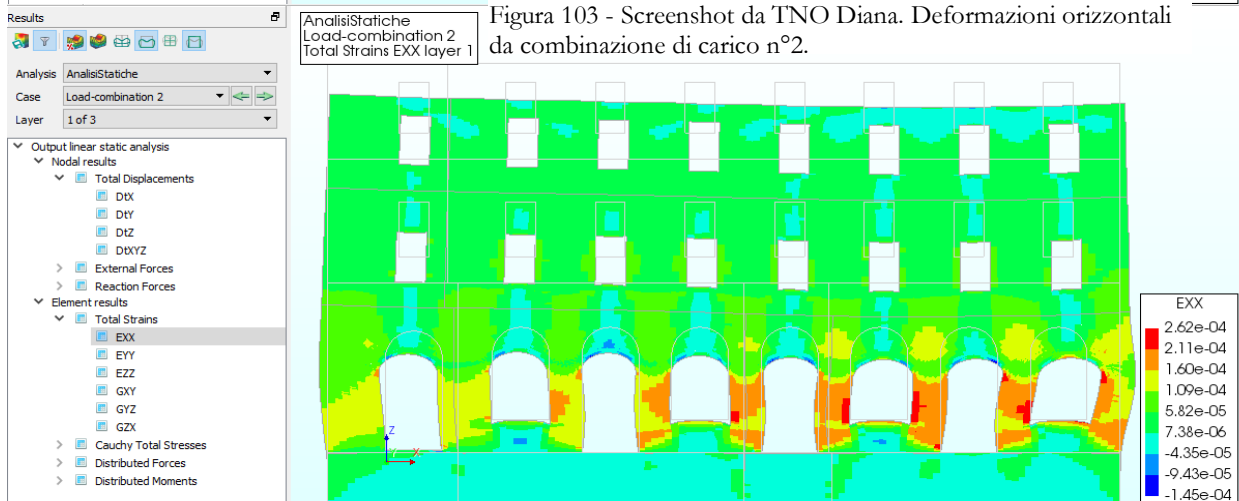


Figura 102 - Screenshot da TNO Diana. Deformazioni verticali da combinazione di carico n°2.



L'ultima analisi è sempre in ambito statico lineare, considerando i soli carichi dovuti all'azione dinamica, non coefficientati. Come il primo caso sono molto evidenti gli andamenti tensionali a puntone che creano le lesioni diagonali.

Figura 106 - Screenshot TNO Diana. Configurazione deformata da combinazioni di carico n°3.

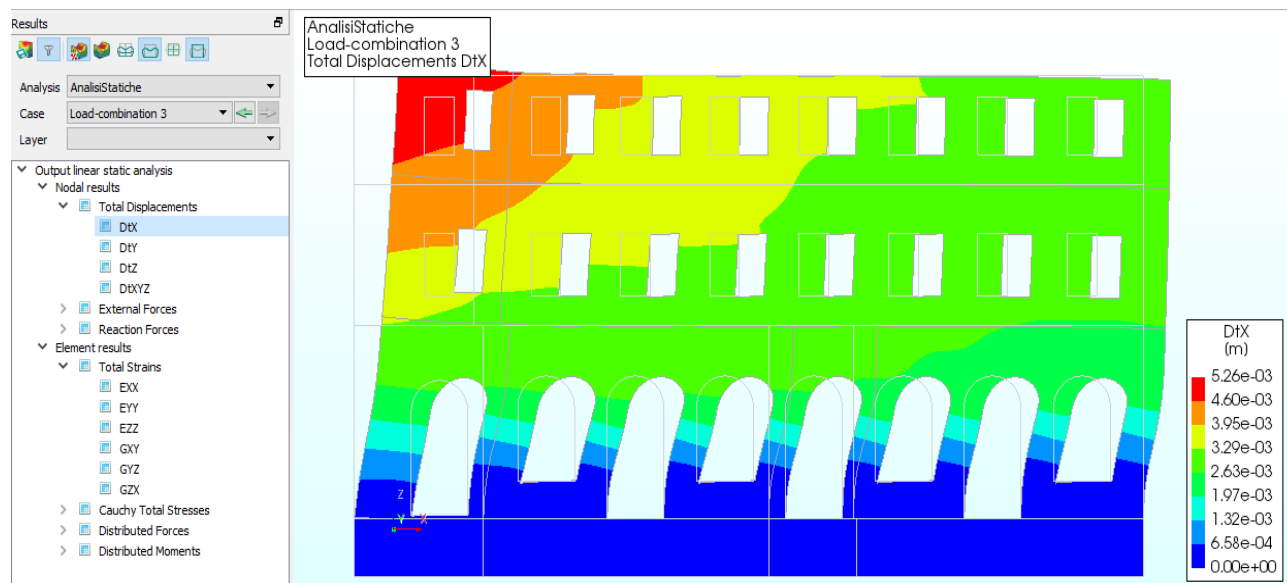
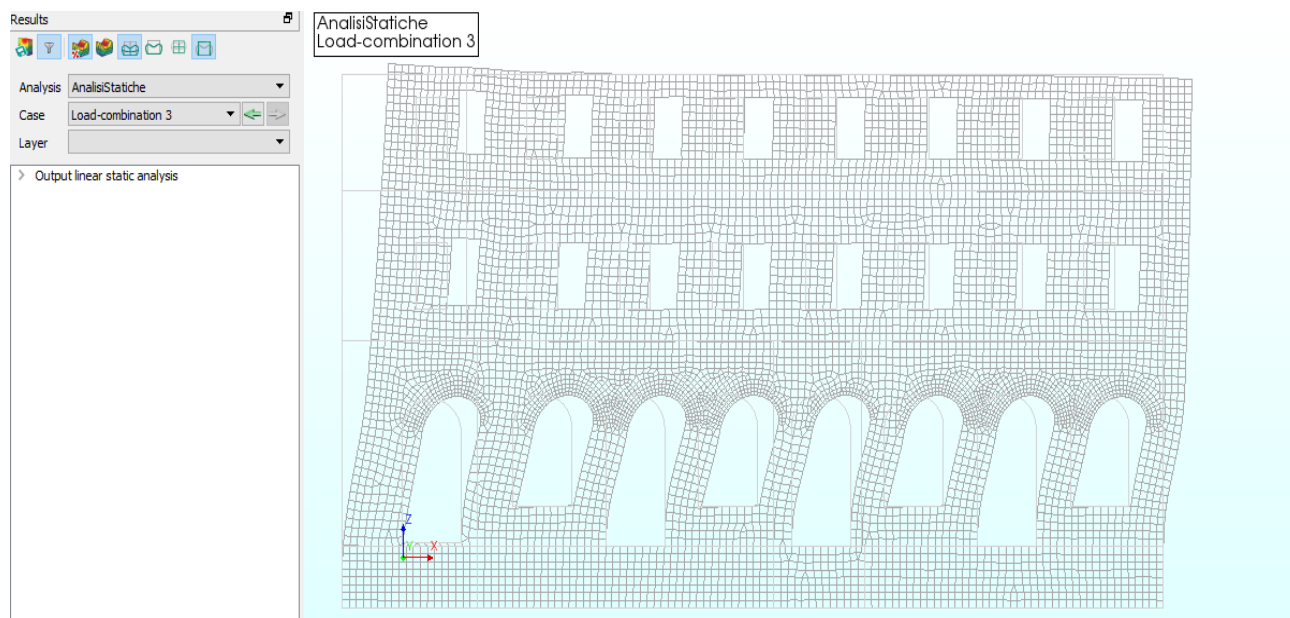


Figura 105 - Screenshot TNO Diana. Spostamenti orizzontali da combinazione di carico n°3.

Figura 109 - Screenshot TNO Diana. Spostamenti verticali da combinazione di carico n°3.

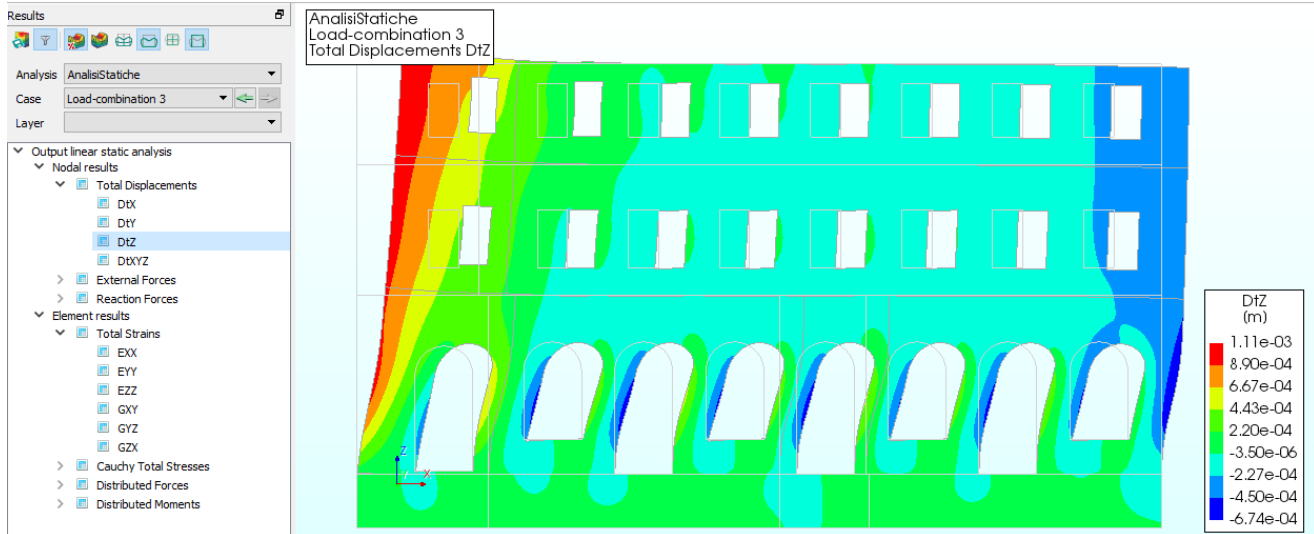


Figura 108 - Screenshot TNO Diana. Deformazioni verticali da combinazione di carico n°3.

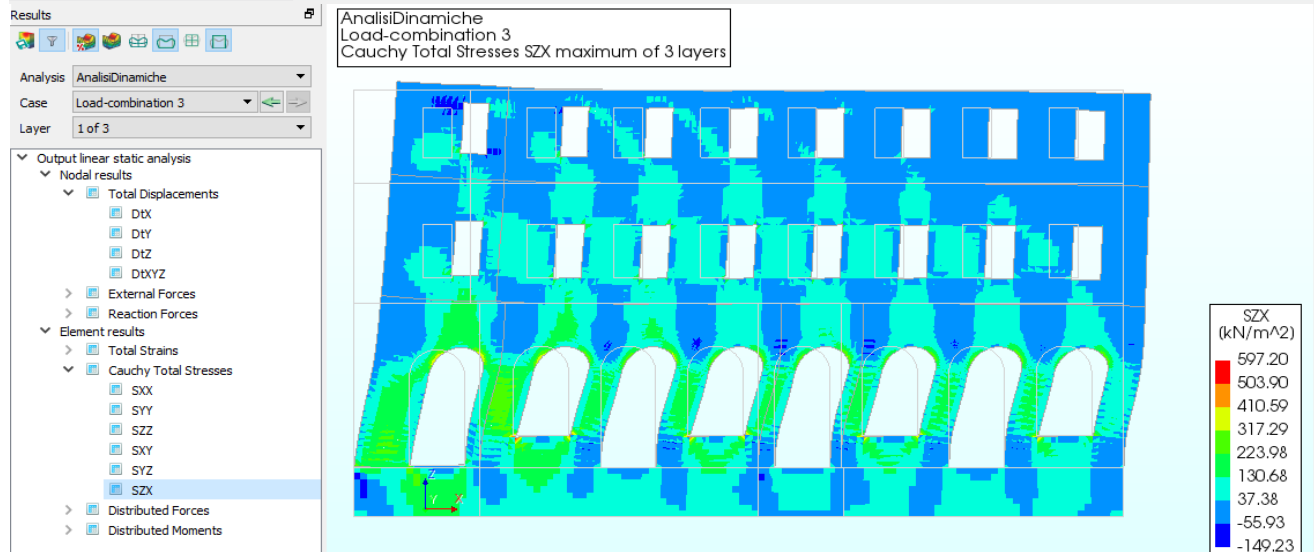
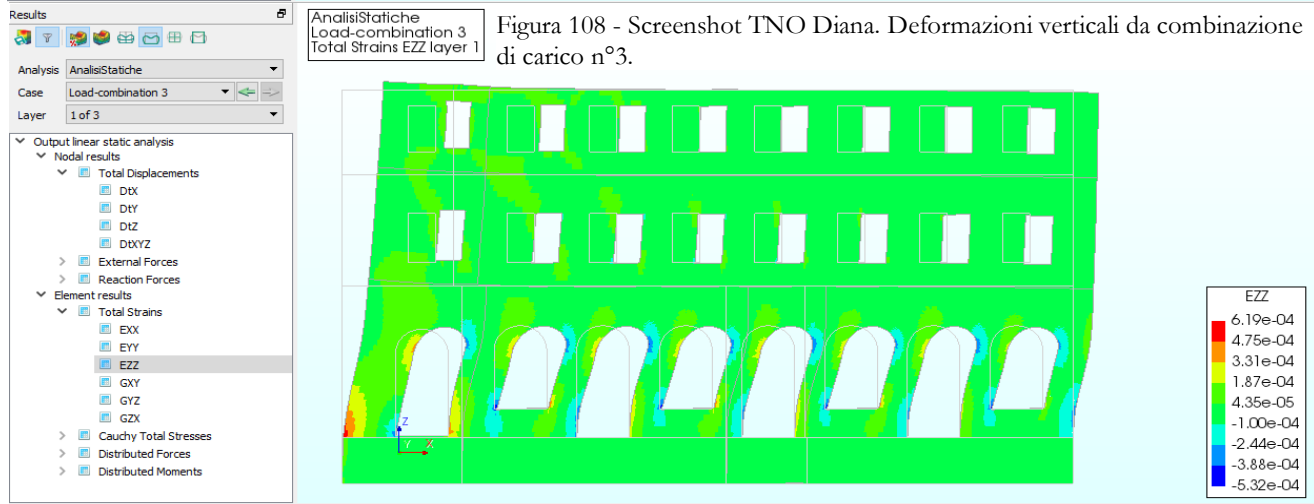


Figura 107 - Screenshot TNO Diana. Sforzi orizzontali da combinazioni di carico n°3.

Infine troviamo le analisi dinamiche. Troviamo prima la determinazione dell'autovalore corrispondente al modo fondamentale di vibrare e ai successivi due modi propri. Poi la determinazione degli spostamenti massimi in corrispondenza del modo fondamentale. È un meccanismo che tende a essere fuori da piano, correttamente quindi è un comportamento corretto quello rappresentato da questo modello. Sarebbe poco interessante verificare il comportamento dinamico nel piano di questo, vista la scarsa applicabilità pratica, se non dopo aver verificato ed eventualmente dopo essere intervenuti per garantire il comportamento unitario della struttura. Come cartina tornasole della bontà di questi risultati si nota che la deformata rispetta i vincoli dati, causati da pareti ortogonali. Si sono considerati i solai, come non offrenti vincolo. In sede di verifica si dovrà controllare il ribaltamento delle porzioni sovrastanti della fascia muraria, sia in corrispondenza della fasce evidenziata da quest'immagine, ma anche in corrispondenza dei maschi murari al piano terra più meno irrigiditi trasversalmente, magari valutando la resistenza a trazione della muratura per stimare la resistenza finita dei pannelli murari trasversali.

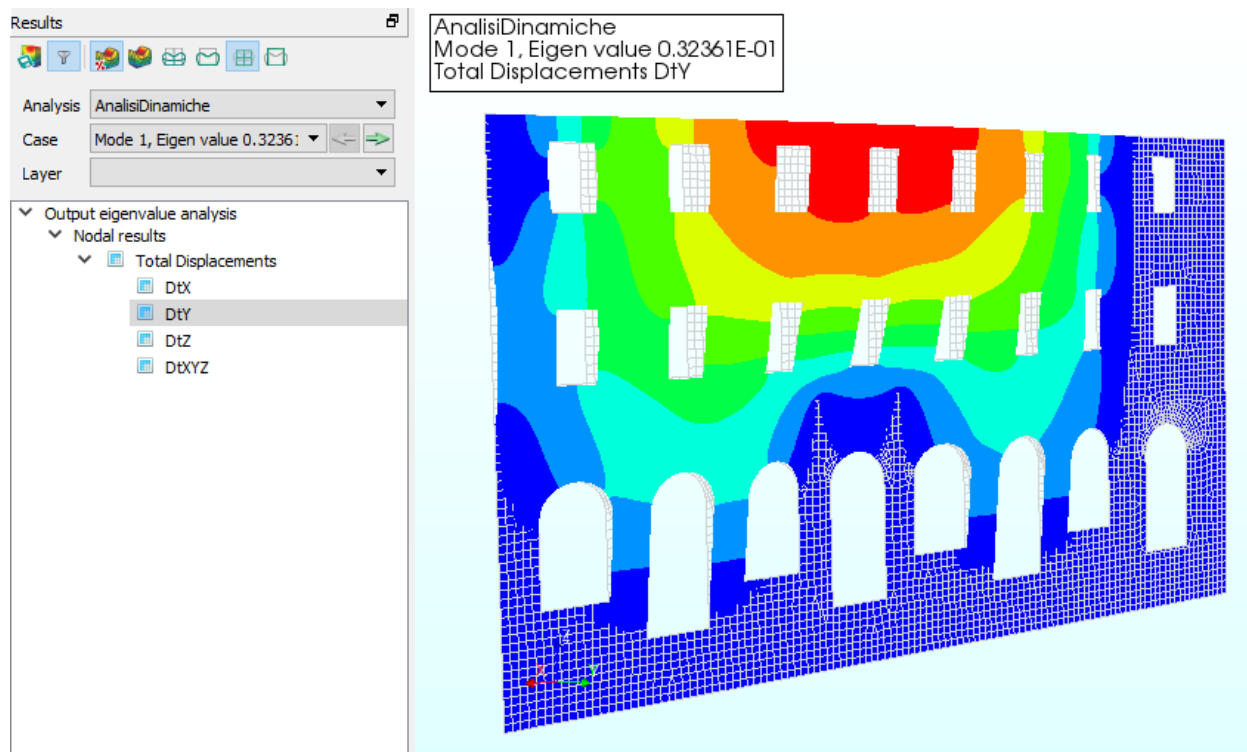


Figura 110 - Screenshot TNO Diana. Determinazione autovalore e deformazione del primo modo di vibrare.

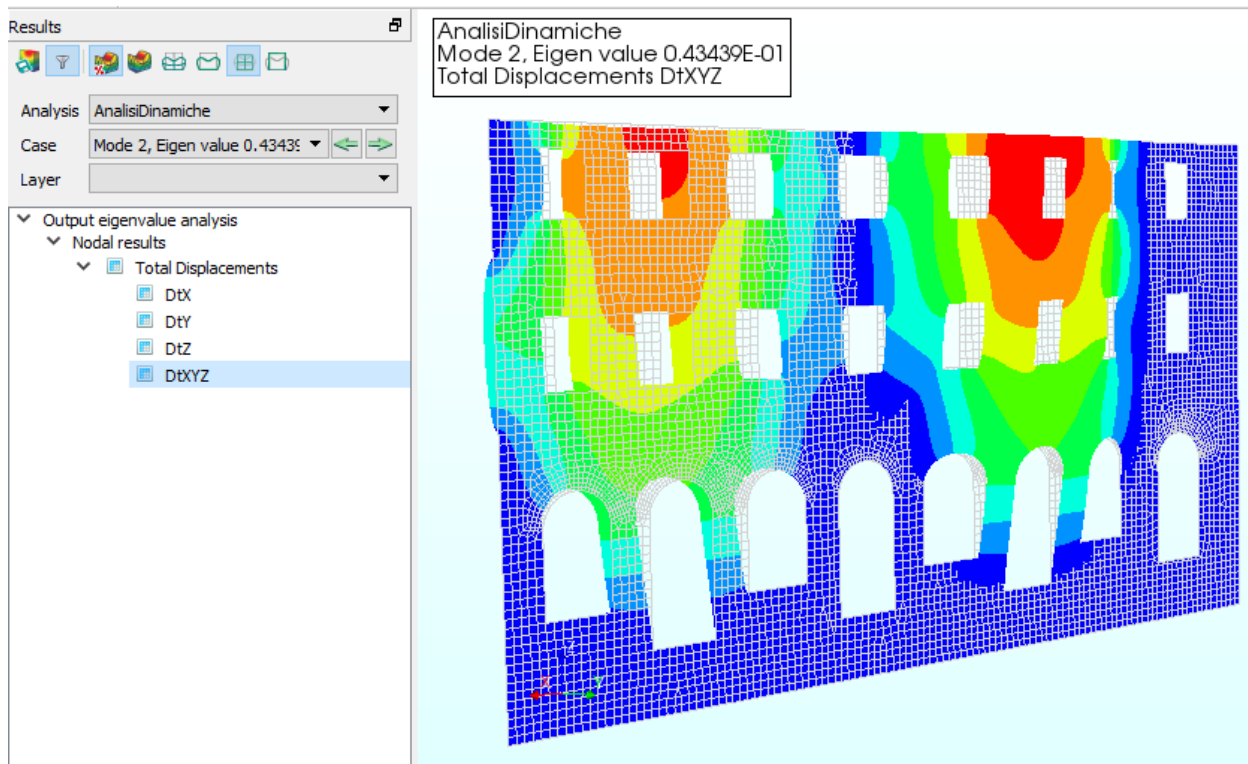


Figura 112 - Screenshot TNO Diana. Determinazione autovalore e deformazione del secondo modo di vibrare.

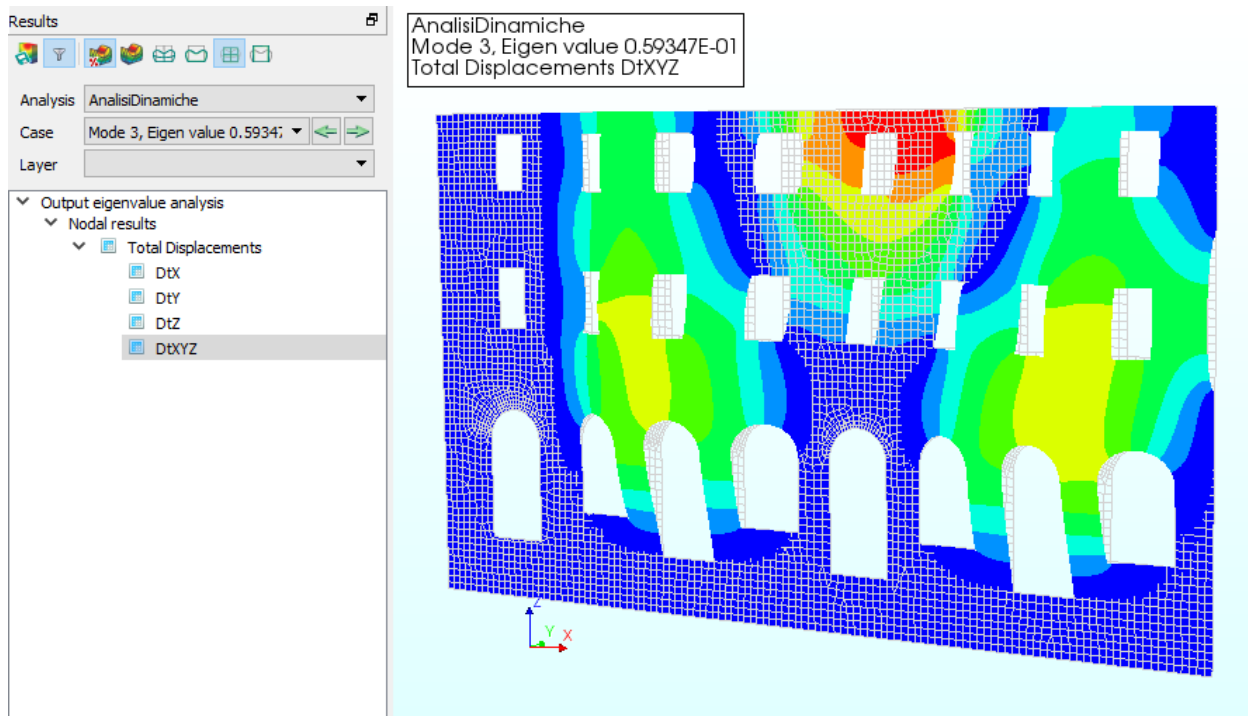


Figura 111 - Screenshot TNO Diana. Determinazione autovalore e deformazione del terzo modo di vibrare.

Figura 113 - Screenshot TNO Diana. Risultati in deformazione dell'analisi modale del primo modo di vibrare.

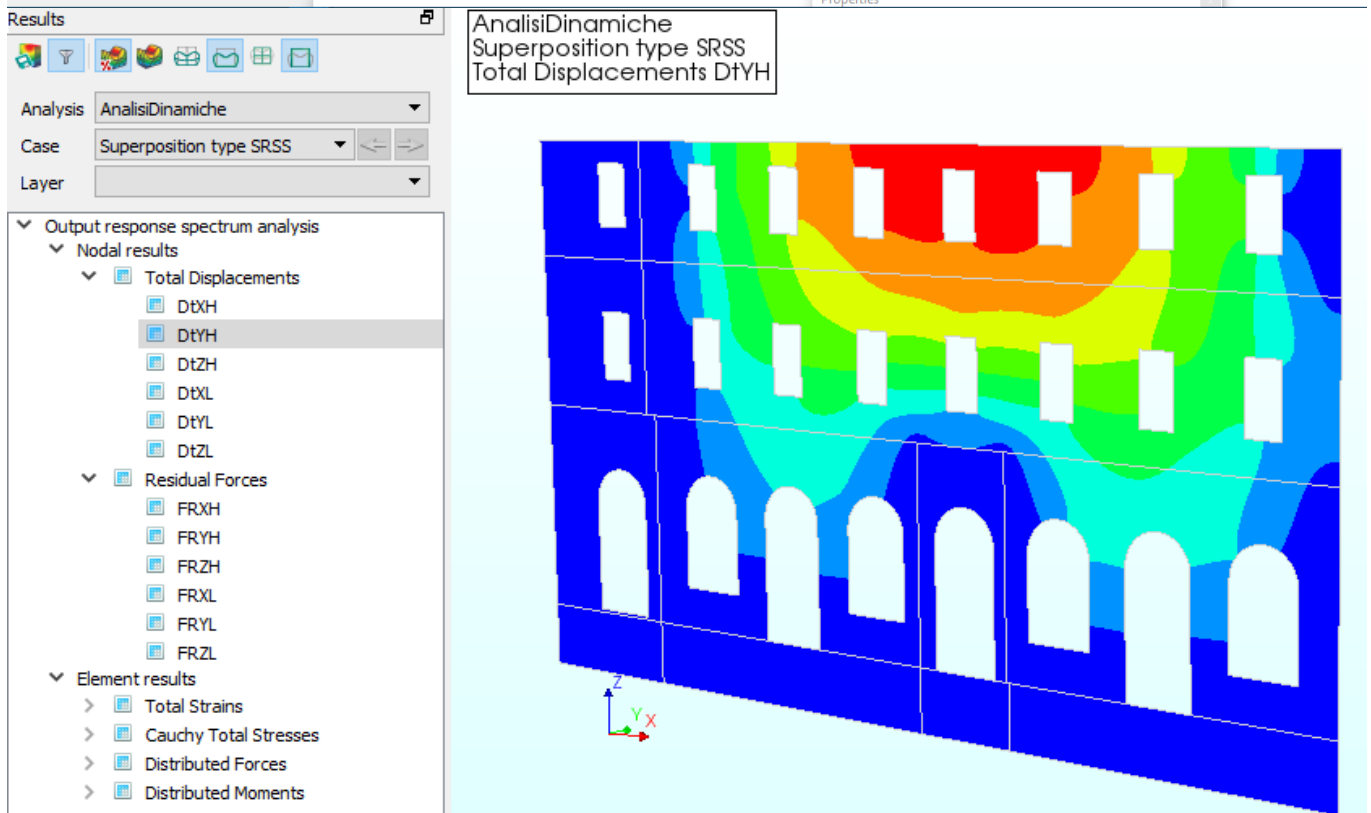
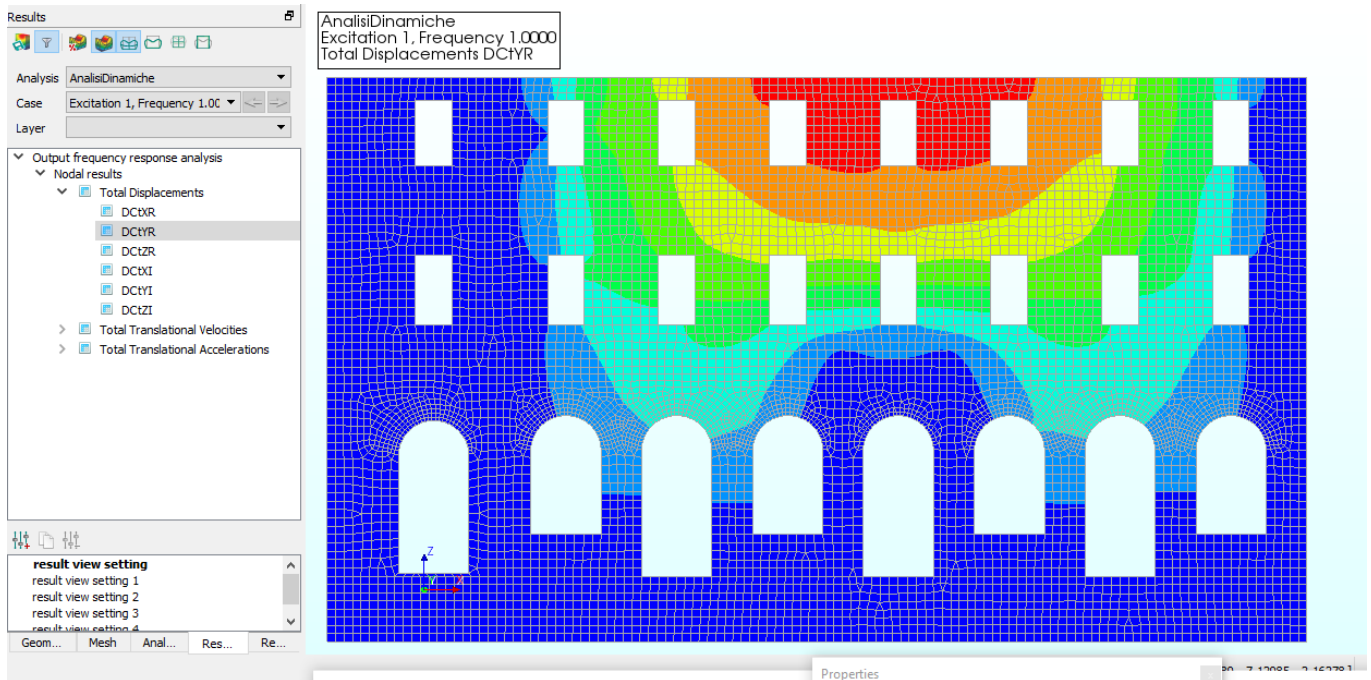


Figura 114 - Screenshot TNO Diana. Risultati dell'analisi dinamica in spostamento dalla sovrapposizione di tipo SRSS.

# NÒLIAN

Con l'assistenza di Softing srl, sono state poi condotte analisi simili in ambiente Nòlian e confrontate con i risultati ottenuti da TNO Diana. Si è partiti dallo stesso modello IFC di partenza e, constatata

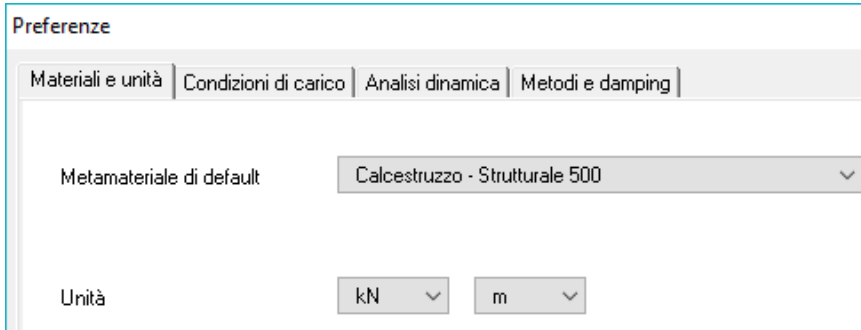


Figure 115 - Screenshot Softing Nòlian. Unità di misura.

## ARCHIVIO MATERIALI

pos	usi	Nome
GLO	0	Calcestruzzo
G+L	973	Calcestruzzo - Strutturale 500
G+L	6785	Laterizio - Strutturale 500

Figure 116 - Screenshot Softing Nòlian. Archivio materiali.

Materiale Calcestruzzo - Strutturale 500

proprietà	valore
Classe calcestruzzo	C16/20
Classe acciaio	B450C
Modulo elastico	28820612.94
Modulo tangenziale	14410306.47
Peso specifico	24.5166250000
Coeff. Dil. Termica	0.000012000

Figure 117 - Screenshot Softing Nòlian. Proprietà Calcestruzzo.

l'importazione corretta e identica a quella effettuata da TNO Diana, si è provveduto, come prima, ad assegnare materiali, carichi e vincoli al modello analitico privo di proprietà. Si è verificato, in prima battuta, la

corrispondenza delle unità di misura e che la libreria dei materiali fosse già presente e assegnata ai singoli oggetti analitici strutturali. Di seguito si sono definite, come da immagine, le caratteristiche meccaniche dei materiali.

Subito dopo, come al caso precedente sono stati assegnati gli stessi carichi, le stesse combinazioni di carico e vincoli. I valori sono identici, a parte per quanto riguarda le spinte sismiche dell'analisi pushover, assegnate alla sommità del muro, per confrontare anche le differenze da questa modellazione.

Materiale 'Laterizio - Strutturale 500'

proprietà	valore
Modulo elastico	2250000.00
Modulo tangenziale	900000.00
Peso specifico	18.0000000000
Coeff. Dil. Termica	0.000012000
Coefficiente sottofondo	49033.2500000000
Proprietà non lineari	nessuno

Figure 118 - Screenshot Softing Nòlian. Proprietà Muratura.



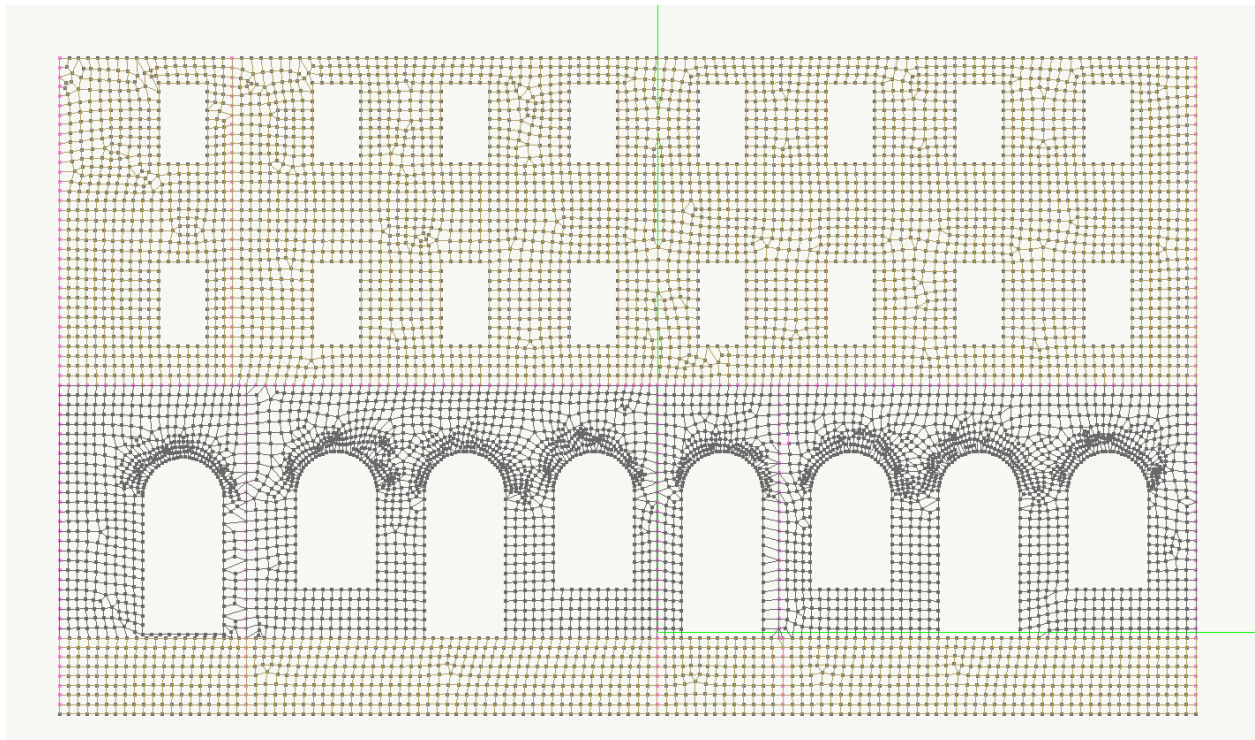


Figure 119 - Screenshot Softing Nölian. Generazione mesh di calcolo.

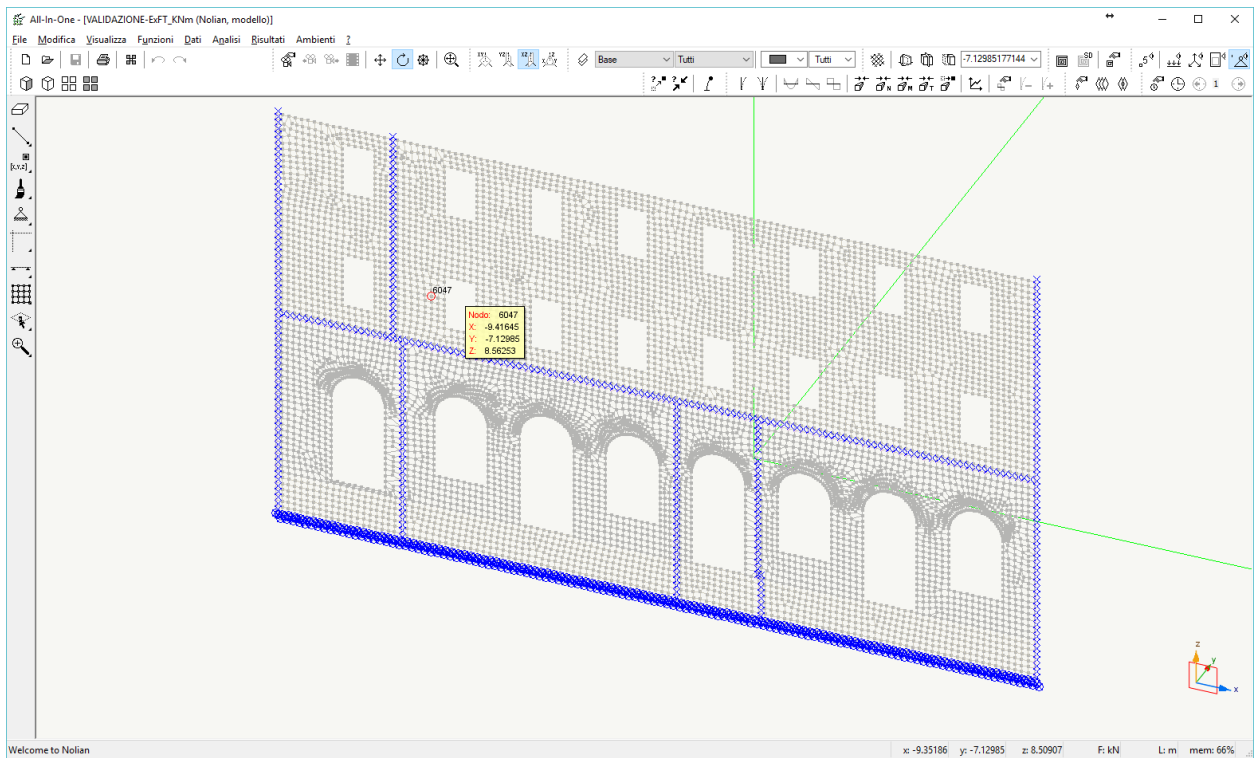


Figure 120 - Screenshot Softing Nölian. Assegnazione vincoli esterni.

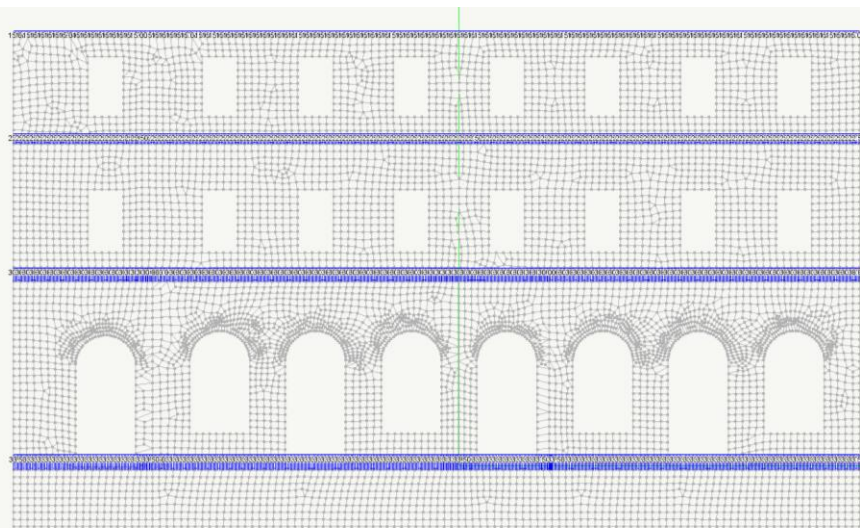


Figure 121 - Screenshot Softing Nòlian. Assegnazione pesi propri portati.

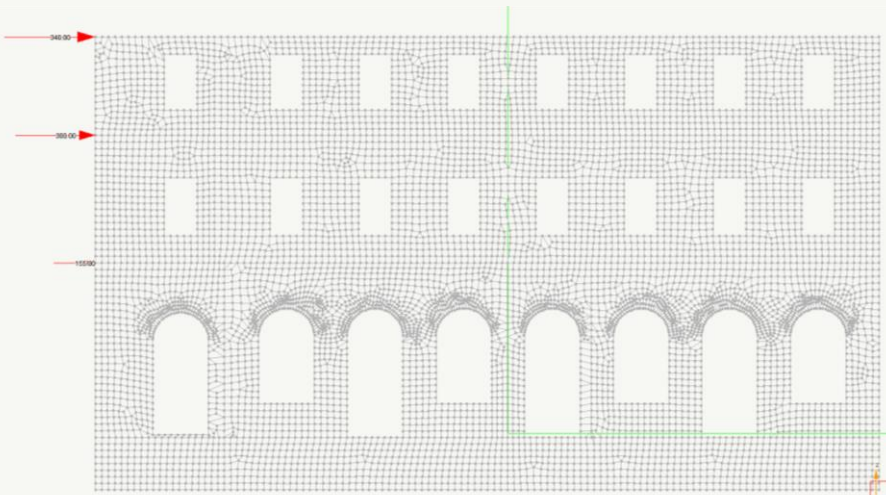


Figure 122 - Screenshot Softing Nòlian. Assegnazione carichi sismici. Rispettivamente 340, 300 e 155 kN.

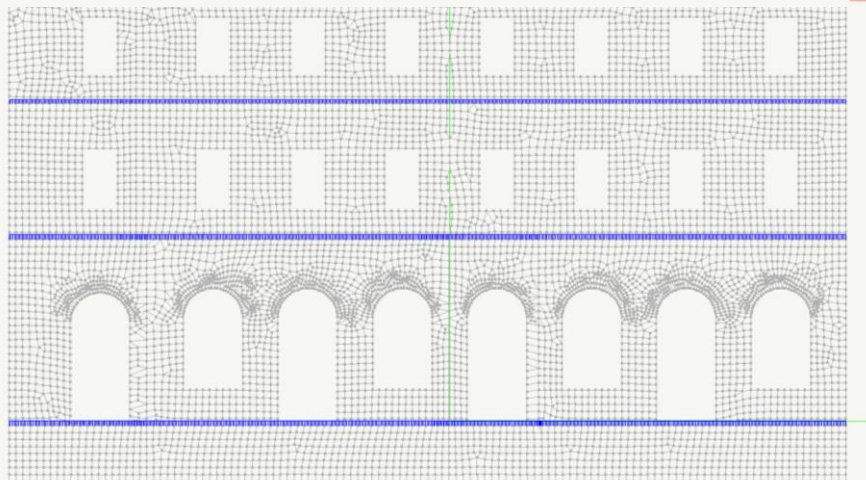


Figure 123 - Screenshot Softing Nòlian. Assegnazione carichi accidentali portati.



Combinazioni										
SLU	Servizio	Danno	Operatività	Geotecniche	Tipi di carico	Coefficienti combinazioni geotecniche				
Dinamica SLVh Y	(1) Dinamica SLVh X	(1) Dinamica SLDh...	(1) Dinamica SLD...	(1) Peso proprio	(1) Permanente	(1) Accidentale	(1) spinta sismica			
0.00	0.00	0.00	0.00	1.00	1.00	0.70	1.00			
0.00	0.00	0.00	0.00	1.30	1.30	1.50	0.00			

Figure 124 - Screenshot Softing Nölian. Creazione combinazioni di carico.

Verificata la correttezza sono state lanciate le analisi, un'analisi statica lineare di pushover, un'analisi statica lineare dei soli carichi verticali, corrispondenti alle combinazioni 1 e 2 utilizzate in TNO Diana. È stata verificata la corrispondenza dei risultati. Sono effettivamente confrontabili e portano alle stesse conclusioni. Difatti sia i valori di spostamento, quanto quelli di deformazione che di sforzi sono dello stesso ordine di grandezza e all'incirca anche dello stesso valore. Le differenze sono minime.

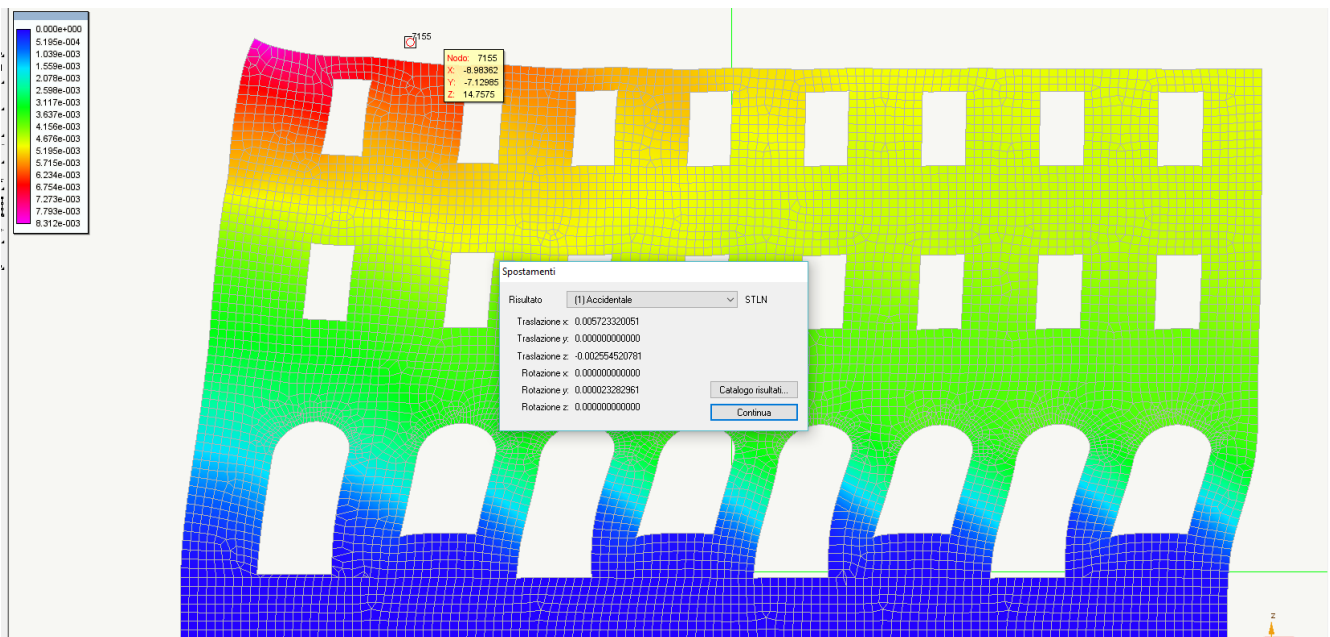


Figure 125 - Screenshot Softing Nölian. Configurazione deformata della combinazione n° 1.

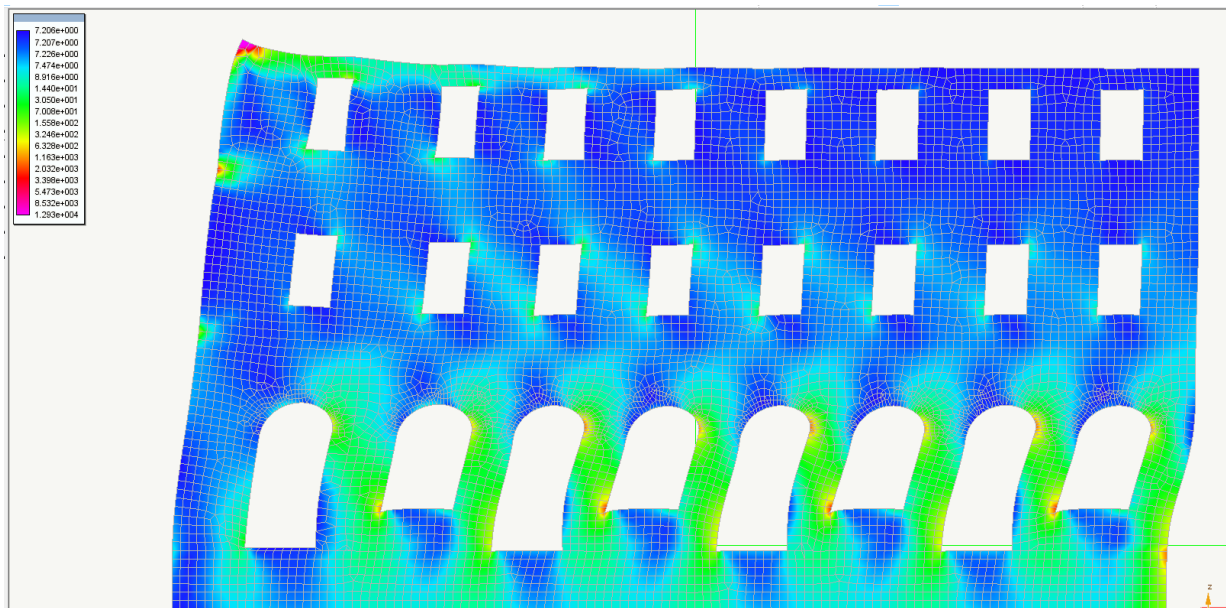


Figure 126 - Screenshot Softing Nòlian. Deformazioni della configurazione n° 1.

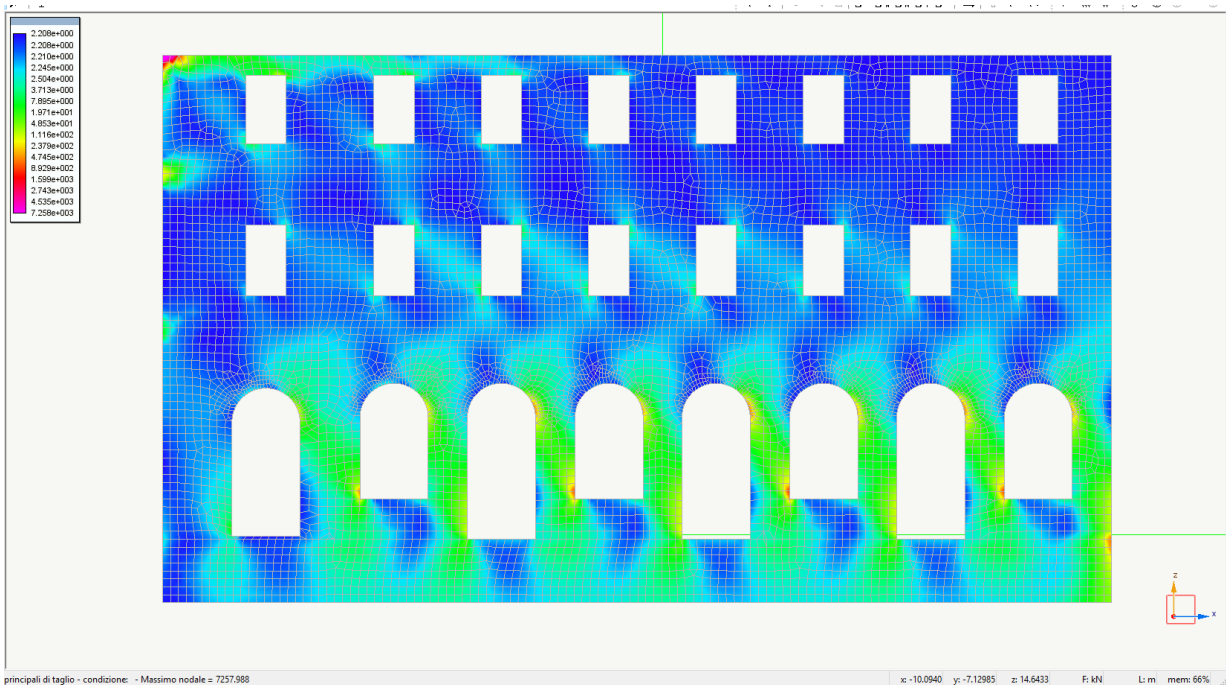


Figure 127 - Screenshot Softing Nòlian. Sforzi principali di compressione della combinazione di carico n° 1.

Figure 128 - Screenshot Softing Nòlian. Configurazione deformata della configurazione di carico n° 2.

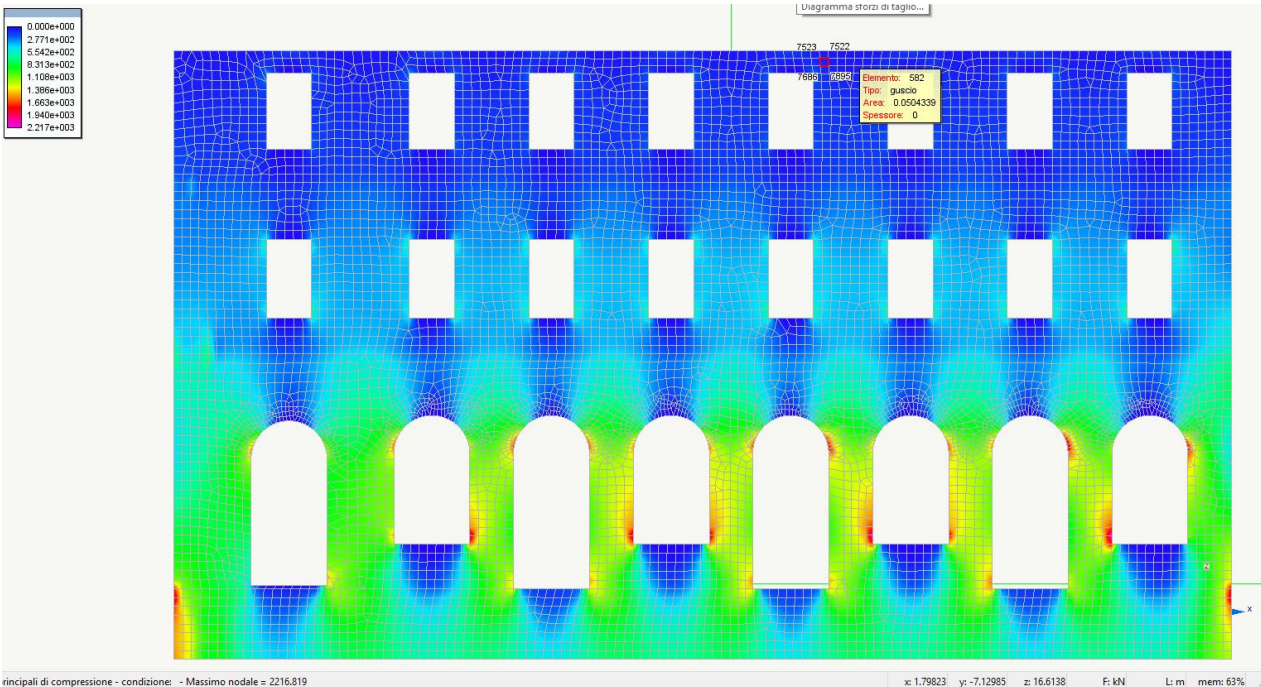
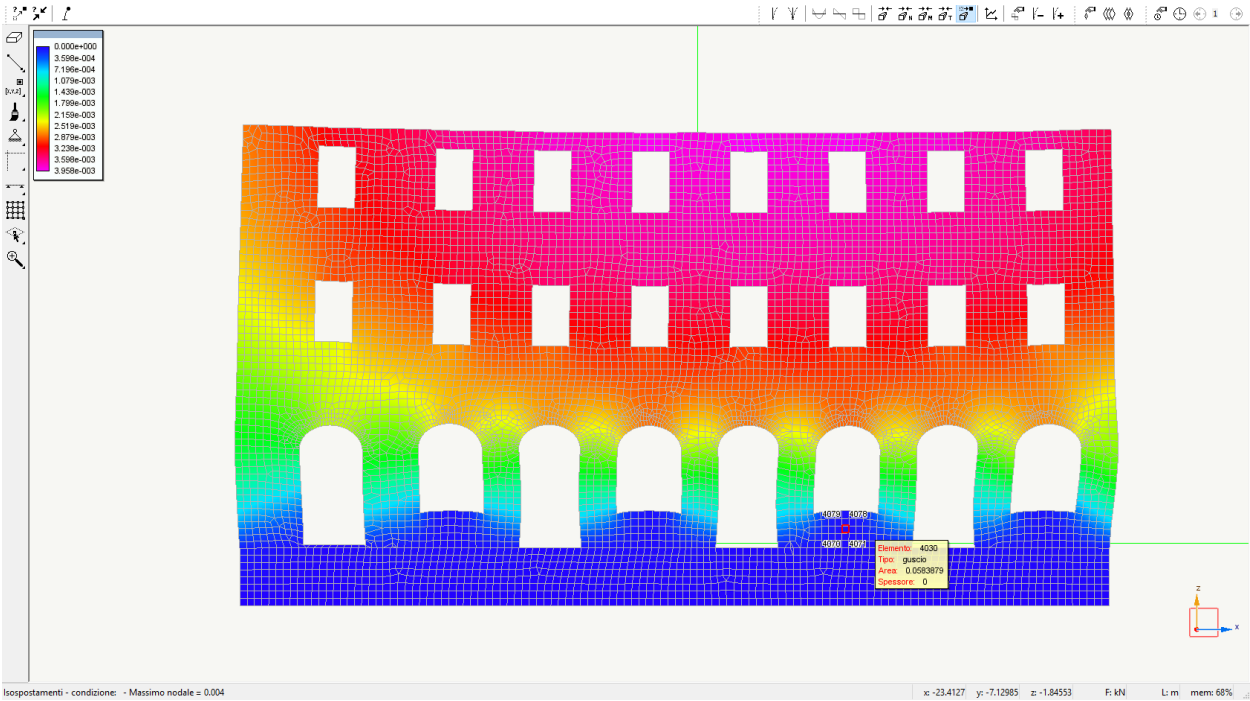


Figure 129 - Screenshot Softing Nòlian. Sforzi principali di compressione della configurazione n° 1.



Di seguito, invece, i risultati dell'analisi modale, che evidenzia il modo fondamentale di vibrare e gli spostamenti attesi in corrispondenza di questo.

Di nuovo si mostra un meccanismo di collasso parziale, vista l'assenza di presidi strutturali che trattengano in sommità la parete, libera di muoversi fuori dal piano. Gli spostamenti, di nuovo attestati all'incirca di un metro, mostrano la necessità di intervenire, se effettivamente il meccanismo si può verificare.

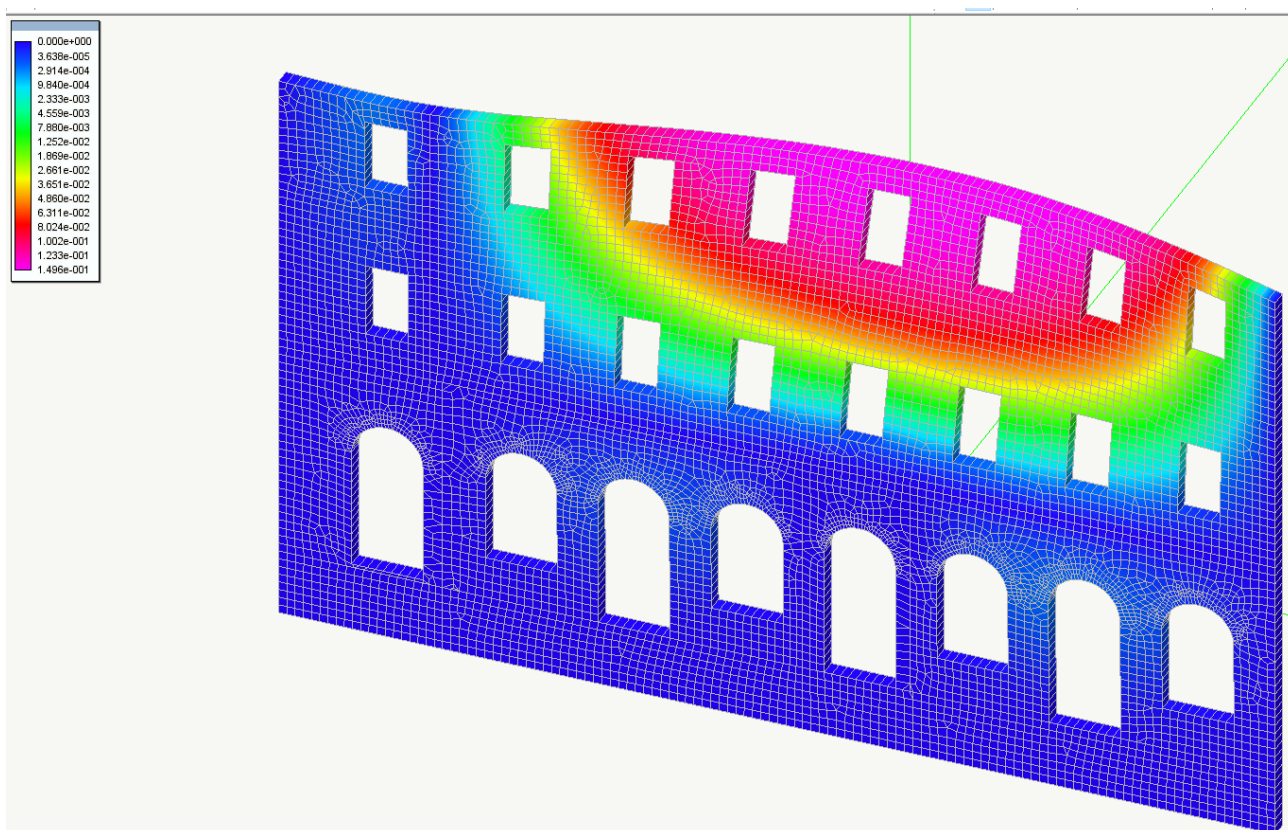


Figure 130 - Screenshot Softing Nòlian. Modo fondamentale di vibrare e corrispondenti spostamenti.



# **9A. risultati**

Arrivati a questo punto possiamo iniziare a ragionare su ciò è stato.

Il software TNO Diana non è riuscito ad esportare in formato IFC i risultati ottenuti, in nessuna maniera. Anzi, presenta una capacità esportativa tutto sommato molto limitata, potendo esportare la geometria in formati CAD e poco altro ancora. Non sembra nemmeno possibile creare un listato, anche in formato excel, per poter consultare esternamente i risultati stessi. Vista la capacità risolutiva non indifferente del software, capace di risolvere velocemente e agilmente, sembra quasi una volontà da parte dei suoi autori di proteggere il loro lavoro, ottimizzando, piuttosto, la consultazione in app, ad esempio.

Tuttavia sia TNO Diana che Nòlina sono riusciti in maniera eccellente a confrontarsi con IFC: probabilmente due dei migliori software in commercio nel recepire le informazioni da questo tipo di file. Decisamente apprezzabile come questi riescano a gestire un modello basato sulla *Coordination View 2.0*, ossia come riescano a recepire gli oggetti architettonici per ricondursi a un modello analitico e, questo, a convertirlo in un modello agli elementi finiti. Ovviamente riescono ad escludere gli oggetti non precipuamente strutturali che compongono il file: molto semplicemente, nel messaggio di importazione vengono riportate anche gli oggetti presenti che non vengono considerati perché appartenenti ad altre discipline.

Per come è stata implementata la cosa, poi, né TNO Diana né Nòlian avranno problemi a confrontarsi con un file IFC che viene da un altro software strutturale, nel caso in cui questo abbia già formato delle informazioni, come quelle relative al modello analitico. Meno immediata, ma facilmente migliorabile la gestione di vincoli e carichi, e combinazioni. In parole povere significa che oggi sono davvero pochi i software capaci di gestire l'informazione strutturale a mezzo IFC, per la possibilità per i programmi di confrontarsi con questi file prodotti da altri programmi similari è piuttosto basse, tuttavia, pare essere possibile e semplice che Diana e Nòlian si possano confrontare adeguatamente anche con questi particolari salvataggi.

Difatti è bene sottolinearlo: non è stata effettuata, nella maniera più assoluta, nessuna modellazione geometrica di alcuna parte della struttura, portando come primo vantaggio una sostanziosa riduzione dei tempi di modellazione. Le uniche difficoltà riscontrate sono nella capacità computazionale, perché,

gestire un file di dimensioni considerevoli è sicuramente complicato, il file iniziale pesava all'incirca 100 MB. Si sottolinea che sia una difficoltà, dovuta in parte anche alla potenza degli strumenti, non un ostacolo insormontabile. Ciò significa che i software FEM si stanno muovendo verso una direzione completamente apprezzabile per la disciplina dell'ingegneria strutturale, quella dell'ottimizzazione della progettazione: è facile immaginare i vantaggi del poter non modellare da zero una struttura, al massimo modificarla, ma solamente controllare la validità di un modello già finito, in cui, magari, vincoli, carichi, combinazioni di carico e caratterizzazione dei materiali sono già implementate o solamente da ritoccare. L'insieme delle informazioni diventa sempre facilmente accessibile senza rischi di perderlo, anzi, di volta in volta arricchendolo e, soprattutto, potendo dedicare le energie alla parte più delicata, ossia la progettazione e la verifica vere e proprie.

Appare evidente però, appunto, come non sia tutto rose e fiori: per queste informazioni vadano conservate e rese accessibili il software deve necessariamente poter produrre un file in uscita in tutto e per tutto simile a quello in ingresso, in cui le informazioni sono intelligentemente catalogate e disponibili a chi le consulta. Si rende così possibile un'efficacissima comunicazione tra le varie parti coinvolte nel processo edilizio, anche solo per la possibilità di riscontri. È a questa circostanza che la Pubblica Amministrazione ha in mente quando parla di software BIM negli appalti pubblici, la possibilità di un'ingegneria e, in generale, di un'edilizia trasparente per la realizzazione e in primo luogo per la gestione delle opere pubbliche.

Ritornando però ai software TNO Diana e Softing Nòlian e confrontandoli con lo standard di buildingSMART, appare addirittura che le *Industry Foundation Classes* implementino una serie di casistiche non contemplate da questi solutori agli elementi finiti: nonostante la semplicità del caso studio è emerso, ad esempio, l'impossibilità di posizionare un carico, uno spostamento impresso, una generica azione o, conseguentemente una reazione, in un punto qualsiasi di uno spigolo o di una faccia del modello analitico, perlomeno in Diana – si badi bene che questo è ben distinto da un modello FEM in cui ciò non è preferibile. Questo ci racconta una cosa e ci dà un'importante conferma: che, a volerlo utilizzare, lo standard IFC raccontato nella ISO 16739:2013 è molto completo e può tranquillissimamente raccontare quasi tutto quello che è il ciclo di vita di un organismo edilizio. La flessibilità e la capacità descrittiva possibili sono davvero molto ampie: si invita a passare in rassegna



quanto incontrato guardando a TNO Diana e Nòlian, sono informazioni solitamente incontrabili in un tipico programma strutturale. Nel sondarlo anche molto approfonditamente sarà difficile incontrare qualcosa di non descrivibile nello standard e, qualora questo non lo sia, sarà verosimilmente oggetto dell'argomento di questa tesi. A titolo di esempio si riporta come ciò che riguarda il completamento delle analisi dinamiche sarebbe facilmente implementabile così come qui iniziato a presentare in questa tesi: gli stessi autovalori e le forme modali possono ottimamente essere aggiunte in un file IFC e, conseguentemente, resi consultabili e aggiornabili in un secondo momento, con la conseguenza che non saranno informazioni che vanno perse. Così come le azioni dinamiche, in qualunque loro tipologia: basti pensare alla comodità che verrebbe dal poter distinguere tra un peso proprio, un carico accidentale e un carico dovuto all'azione sismica: tutto ben distinto e funzionale con enorme vantaggi per quanto riguarda l'utente, poiché sarebbe più semplice accedere criticamente alle informazioni, potendo individuare difficoltà e problemi e potendo intervenire efficacemente nella gestione del progetto e dell'immobile.

Volendo poi pesare anche la parte di proposta avanzata, sembra che così come la si avanza questa possa descrivere compiutamente sia modelli e analisi ad azioni finite, sia, almeno parzialmente, analisi dinamiche. Nel descrivere il modello FEM sono stati considerati tutti gli oggetti possibili, in tutte le loro declinazioni, e con sufficiente flessibilità così da poter includere tutti gli elementi informatici che emergono durante il processo in maniera più o meno esplicita: a partire dagli elementi e i nodi, sulla base della loro dimensionalità, dei loro gradi di libertà e delle loro, conseguenti funzioni di forma, passando per gli specifici elementi per la risoluzione come i punti di integrazione, arrivando alla descrizione di un risultato delle analisi, sia espresso come sforzi e deformazioni di un elemento finito, parametrizzato o meno, sia espresso come sforzi e deformazioni dell'elemento strutturale. Infine, anche l'implementazione delle classi per le analisi dinamiche sembra efficace, efficiente e sebbene da completare: risultati spesso non meramente grafici come autovalori sono implementati di pari passo a risultati come i gli spostamenti e gli autovettori. Anche la modalità di gestione, basata sul concetto di voler trasmettere un'informazione corretta e trasparente ha portato i suoi frutti, poiché parametri spesso trascurati nella trasmissione di un risultato di calcolo, si pensi ai coefficienti di Rayleigh, possono essere gestiti correttamente. In sostanza si pensa che, qualora la proposta diventi parte

effettiva di IFC, possa risultare soddisfacente, potendo gestire i dati di modelli e analisi agli elementi finiti e i dati di analisi, in ingresso e in uscita, dinamiche.

In conclusione, sebbene la strada non sia assolutamente conclusa, né per IFC e buildingSMART, né, soprattutto per i produttori di software, in particolare di TNO Diana e di Softing Nòlian, si può dire senza patema di sbagliare che la direzione presa finalmente dal mondo degli applicativi per l'ingegneria strutturale è quella giusta ed è davvero soddisfacente.



# **10. conclusioni**

L'obiettivo iniziale di questa tesi era l'estensione di IFC per poter descrivere meglio il mondo dell'ingegneria strutturale e come questa intervenga all'interno del ciclo di vita del bene edilizio, dalla progettazione, al recupero fino allo smaltimento. Rimaneva impossibile descrivere i modelli agli elementi finiti e i loro risultati all'interno di un ciclo di vita di un organismo edilizio rappresentato all'interno dello standard, così come le analisi dinamiche condotte, per stessa ammissione di buildingSMART. Ma, molto di più, ancora ampie sono le carenze che lo affliggevano, e continuano ad affliggerlo. Difatti, man mano che si è proceduto con la conoscenza dello stato dell'arte in cui versano le *Industry Foundation Classes* e il mondo delle analisi strutturali appariva via via sempre più evidente come il lavoro da fare, come standardizzazione dell'informazione, fosse ancora molto ampio, nonostante ancora molti software non si confrontino con programmi BIM e nonostante molte discipline dispongano di davvero pochi strumenti informatici dedicati che, men che meno, neanche loro si confrontano con gli applicativi in Building Information Modeling – si pensi per i primi allo strumento del GIS, oltre ai già ricordati solutori a telaio equivalente e solutori per meccanismi locali. Tra l'altro, si spera che il lettore possa aver intuito la mole di lavoro ancora da svolgere e, di più, anche l'urgenza di svolgerlo vista l'esigenza di questo tipo di strumento. Più ci si confrontava con le mancanze di questo standard, più veniva spontaneo cercare di volta in volta di poter descrivere correttamente ciò che si incontrava, per poter completare IFC, sapendo come questo fosse impossibile per questo solo lavoro di laurea.

Per concludere questa tesi va ricordato che manca la validazione effettiva del modello agli elementi finiti e delle analisi dinamiche da parte di buildingSMART, perché questi possano essere recepite all'interno di IFC: si procederà, come le intenzioni iniziali, a sottoporre questo lavoro nel suo corpo principale al capitolo nazionale e quello internazionale dell'associazione, perché possa essere vagliato e, si spera, inserito all'interno dello standard ISO 16739.

Per quanto riguarda, inoltre, la seconda parte del lavoro, come descrizione di oggetti tra cui il tirante, rimane ancora impossibile in molti software BIM e la standardizzazione di ciò rimane subordinata alla possibilità di poter modellare correttamente questi oggetti, altrimenti rimarrà una normativa inutilizzata. Similmente per quanto riguarda l'interfaccia dei modelli a telai equivalenti: praticamente nessuno di questi si confronta davvero con software BIM, rimanendo, come massima espressione di

interoperabilità, nell'ambito dei software CAD, perdendo clamorosamente le informazioni del file e rimanendo esclusi da una gestione intelligente del progetto e dell'edificio in genere. Per cui, per poter validare questi modelli deve prima succedere che questi reciproci software dialoghino gli uni con gli altri. La forte speranza è che, con uno strumento come IFC e una fetta di mercato come quella sempre più ampia del BIM, anche i produttori di questo tipo di software possano trovare facile e conveniente mettere in relazione i loro prodotti con questo mondo. In questo modo, questo lavoro di tesi servirebbe come punto di partenza, come un qualcosa che può indicare la via più semplice da seguire.

Ciò per cui si sente di poter dire che ciò che si è presentato è un lavoro valido è stata l'enorme tempo ed energia spesi per comprendere lo stato dell'arte delle varie discipline che si sono interfacciate. Si ritiene, infatti, che, essendo un'attività prettamente di normazione, fosse davvero necessario, indispensabile e obbligatorio, sapere effettivamente le regole del buon costruire, del buon progettare e del buon programmare, guardando alle esigenze e alle modalità operative di tutti. Cioè come nella pratica agiscono i vari programmi, come nella pratica agiscono i professionisti utilizzandoli e come, nella pratica, svolgano anche fisicamente il loro lavoro.

Ossia, ad esempio perché si relazionassero software FEM e software BIM si è cercato di fare propri il *modus operandi* degli uni e degli altri, comprendendo a fondo, nel primo caso, anche la teoria matematica che ne è alla base e le sue applicazioni pratiche. In questo modo sapendo il funzionamento di entrambi si è potuto capire come entrambi debbano relazionarsi, mantenendo le peculiarità, cioè le finalità e i processi che appartengono a ciascuno. Perché soltanto così il BIM può essere *Building Information Modeling*, se ogni informazione utile del processo è mantenuta e catalogata intelligentemente. Lo stesso è stato necessario per gettare le basi per il modello a telaio equivalente e tutto ciò che viene affrontato all'interno di questa tesi di laurea.



# **11. bibliografia**



## BIBLIOGRAFIA BIM E IFC

-G. Pasqualotto, “Modellazione informative interoperabile del degrado degli edifici. Proposta di un’ontologia IFC/open BIM”, tesi di laurea presso l’Università degli Studi di Padova, discussa, A.A. 2015-2016;

-buildingSMART. (2007). IFC2x Edition 3 Technical Corrigendum 1. Consultato il 11 giugno 2017, da: <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm>;

-buildingSMART. (2010). Information Delivery Manual Guide to Components and Development Methods. buildingSMART, 1–84. BuildingSMART. (2015a). IFC4 - Addendum 1. Consultato il 11 giugno 2017, da: <http://www.buildingsmart-tech.org/ifc/IFC4/Add1/html/>;

-buildingSMART. (2015b). IFCDoc. Consultato il 11 giugno 2017, da: <http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool>;

-buildingSMART. (2016). buildingSMART - Home. Consultato il 11 giugno 2017, da: <http://buildingsmart.org/>;

-buildingSMART, (2017). IFC4 – Addendum 2. Consultato il 10 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html>;

-buildingSMART, (2017). IFC4 – Addendum 2 - Contents. Consultato il 10 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/toc.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 - Forewords. Consultato il 10 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/foreword.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 - Introduction. Consultato il 10 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/introduction.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – Scope. Consultato il 10 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/chapter-1.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – Normative Reference. Consultato il 10 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/chapter-2.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – Terms, definitions and abbreviated terms. Consultato il 10 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/chapter-3.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcStructuralAnalysisDomain. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcstructuralanalysisdomain.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcStructuralItem. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcstructuralitem.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcStructuralActivity. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcstructuralactivity.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcRelConnectsStructuralActivity. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcrelconnectsstructuralactivity.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcRelConnectsStructuralMember. Consultato il 11 giugno, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcrelconnectsstructuralmember.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcStructuralAnalysisModel. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcstructuralanalysismodel.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcStructuralLoadResource. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcstructuralloadresource.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcStructuralLoad. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcstructuralload.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcBoundaryCondition. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcboundarycondition.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcStructuralConnectionCondition. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcstructuralconnectioncondition.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcSharedBldgElements. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcsharedbldgelements.htm>;

-buildingSMART, (2017). IFC4 – Addendum 2 – IfcStructuralElementsDomain. Consultato il 11 giugno 2017, da <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifcsharedbldgelements.htm>;

-ST-Developer Tools Reference. (2016) “EXPRESS-G Language Overview”. Consultato il 14 marzo 2017, da [http://www.steptools.com/support/stdev\\_docs/devtools/expg-overview.html](http://www.steptools.com/support/stdev_docs/devtools/expg-overview.html);

-Standard ISO. (1994a). ISO 10303-1:1994 - Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles. Consultato il 28 giugno 2017, da: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=20579](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=20579);

-Standard ISO. (1994b). ISO 10303-31:1994 - Industrial automation systems and integration - Product data representation and exchange - Part 31: Conformance testing methodology and framework: General concepts. Consultato il 28 giugno 2017, da: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=20581](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=20581);

-Standard ISO. (1998a). ISO 10303-22:1998 - Industrial automation systems and integration - Product data representation and exchange - Part 22: Implementation methods: Standard data access interface.

Consultato il 28 giugno 2017, da:  
[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=25097;](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=25097;)

-Standard ISO. (1998b). ISO 10303-32:1998 - Industrial automation systems and integration - Product data representation and exchange - Part 32: Conformance testing methodology and framework: Requirements on testing laboratories and clients. Consultato il 28 giugno 2017, da:  
[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=20582;](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=20582;)

-Standard ISO. (2001). ISO 10303-34:2001 - Industrial automation systems and integration - Product data representation and exchange - Part 34: Conformance testing methodology and framework: Abstract test methods for application protocol implementations. Consultato il 28 giugno 2017, da:  
[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=25073;](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=25073;)

-Standard ISO. (2003). ISO/TS 10303-35:2003 - Industrial automation systems and integration - Product data representation and exchange - Part 35: Conformance testing methodology and framework: Abstract test methods for standard data access interface (SDAI) implementations. Consultato il 28 giugno 2017, da:  
[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=29286;](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=29286;)

-Standard ISO. (2004a). ISO 10303-11:2004. - Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 11: Description Methods: The Reference Manual EXPRESS Language. Consultato il 28 giugno 2017, da:  
[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=38047;](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=38047;)

-Standard ISO. (2004b). ISO 1101:2004. - Geometrical Product Specifications (GPS) - Geometrical Tolerancing - Tolerances of Form, Orientation, Location and Run-Out. Consultato il 28 giugno 2017, da: [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=1147;](http://www.iso.org/iso/catalogue_detail.htm?csnumber=1147;)

-Standard ISO. (2005). ISO 10303-14:2005 - Industrial automation systems and integration – Product data representation and exchange - Part 14: Description methods: The EXPRESS-X language reference manual. Consultato il 28 giugno 2017, da:  
[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=32472;](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=32472;)

-Standard ISO. (2007a). ISO 10303-28:2007. - Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 28: Implementation Methods: XML Representations of EXPRESS Schemas and Data, Using XML Schemas. Consultato il 28 giugno 2017, da: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=40646](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=40646);

-Standard ISO. (2010). ISO 29481-1: 2010 (E): - Building Information Modeling - Information Delivery Manual - Part 1: Methodology and Format. Kassem, M., Brogden, T. and Dawood;

-Standard ISO. (2011). ISO/TS 10303-26:2011 - Industrial automation systems - Product data representation and exchange - Part 26: Implementation methods: Binary representation of EXPRESS-driven data. Consultato il 28 giugno 2017, da: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=50029](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50029);

-Standard ISO. (2013). ISO 16739:2013 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. Consultato il 28 giugno 2017, da: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=51622](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=51622);

-Standard ISO. (2016a). ISO 10303-21:2016. - Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure. Consultato il 28 giugno 2017, da: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=63141](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63141);

-Standard ISO. (2016b). ISO/TC 184 - Automation systems and integration. Consultato il 28 giugno 2017, da: [http://www.iso.org/iso/iso\\_technical\\_committee%3Fcommid%3D54110](http://www.iso.org/iso/iso_technical_committee%3Fcommid%3D54110);

-Robert W. Schuler, “The Application of ISO 10303-11 (the EXPRESS Language) in Defining Data Models for Software Desing and Implementation”, april 2001;

-W3C. (2016). XML Technology. Consultato il 28 giugno 2017, da: <https://www.w3.org/standards/xml/>;

-Graphisoft. (2016). ARCHICAD. Consultato il 28 giugno 2017, da: <http://www.graphisoft.com/archicad/>.

## BIBLIOGRAFIA STRUTTURE E FEM

- M. Hassanien Serror, J. Inoue, Y. Adachi, Y. Fujino, “Shared computer-aided structural design model for construction industry (infrastructure)”, pubblicato su “Computer-Aided Design”, Issue 7, Volume 40, Anno 2008, pp. 778-788;
- A. Farello, “Open (?) BIM Structural Interoperability Model”, intervento presso SAIE 2016;
- National Agency for Finite Element Methods & Standards, “A finite Element primer”, 1986, Edizioni Department of Trade and Industry;
- L. Corradi Dell’Acqua, “Le teorie strutturali e il metodo degli elementi finiti”, Ed. MacGraw-Hill - Milano, 2010;
- F. Cesari, “Metodo degli elementi finiti: elemento asta, elemento ad anello C°, elemento trave, elemento ad anello C1, elementi isoparametrici piani e assialsimmetrici, elementi lastra C1, elementi flessionali C1, elementi isoparametrici spaziali”, Ed. Pitagora Editrice Bologna – Bolohna, 2015;
- E. Tonti, “Introduzione Elementare agli Elementi Finiti”, Università degli Studi di Trieste;
- S. Baragetti, “Progettazione strutturale per elementi finiti”, Università degli Studi di Bergamo;
- G. Strang, “Computational Science and Engineering”, Massachusetts Institute of Technology, Wellesley-Cambridge Press;
- A. Gugliotta, “Elementi finite-Parte I”, Politecnico di Torino, maggio 2002;
- A. Gugliotta, “Elementi finite-Parte II”, Politecnico di Torino, maggio 2002;
- A. Gugliotta, “Elementi finite-Parte III”, Politecnico di Torino, maggio 2002;
- A. Gugliotta, “Elementi finite-Parte IV”, Politecnico di Torino, maggio 2002;
- Decreto Ministeriale 14 gennaio 2008, “Norme Tecniche per le Costruzioni”; consultato il 25 giugno 2017  
tratto da:  
[http://www.cslp.it/cslp/index.php?option=com\\_content&task=view&id=66&Itemid=20](http://www.cslp.it/cslp/index.php?option=com_content&task=view&id=66&Itemid=20);

-Circolare esplicativa, n. 617 02 febbraio 2009, “Istruzioni per l’applicazione delle «Nuove norme tecniche per le costruzioni» di cui al decreto ministeriale 14 gennaio 2008”; consultato il 25 giugno 2017, tratto da [http://cslp.mit.gov.it/index.php?option=com\\_content&task=view&id=79&Itemid=20](http://cslp.mit.gov.it/index.php?option=com_content&task=view&id=79&Itemid=20);

-Decreto del Presidente dei Ministri, 09 febbraio 2011, “Linee Guida per la valutazione e riduzione del rischio sismico del patrimonio culturale allineate alle nuove Norme tecniche per le costruzioni (d.m. 2008)”, tratta da: [http://www.beniculturali.it/mibac/export/MiBAC/sito-MiBAC/Contenuti/Avvisi/visualizza\\_asset.html\\_1141304737.html](http://www.beniculturali.it/mibac/export/MiBAC/sito-MiBAC/Contenuti/Avvisi/visualizza_asset.html_1141304737.html);

-Direttiva del presidente del Consiglio dei Ministri, n. 12 ottobre 2007, “Direttiva del presidente del Consiglio dei Ministri per la valutazione e la riduzione del rischio sismico del patrimonio culturale con riferimento alle norme tecniche per le costruzioni”, GU n. 24 del 29 gennaio 2008-Suppl. Ordinario n. 25;

-Maria Rosa Valluzzi et alii, “Dispense del Corso di Restauro e Laboratorio”, Corso di Laurea in Ingegneria Edile-Architettura, Università degli Studi di Padova, A.A. 2015/16;

-Claudio Modena, Maria Rosa Valluzzi et alii, “Dispense del Corso di Problemi Strutturali dei Monumenti e dell’Edilizia Storica”, Corso di Laurea in Ingegneria Edile-Architettura, Università degli Studi di Padova, A.A. 2015/16;

-Francesca Da Porto, “Dispense del Corso di Tecnica delle Costruzioni 2”, Corso di Laurea in Ingegneria Edile-Architettura, Università degli Studi di Padova, A.A. 2015/16;

-L. Teresi, “Corso di Dinamica delle Strutture. Dispense – parte #1” versione 1.1.2, Università degli Studi di Roma3, A.A. 2015-16;

-L. Teresi, “Corso di Dinamica delle Strutture. Dispense – parte #2” versione 1.1.2, Università degli Studi di Roma3, A.A. 2015-16;

-L. Teresi, “Corso di Dinamica delle Strutture. Dispense – parte #3” versione 1.1.2, Università degli Studi di Roma3, A.A. 2015-16;

- L. Teresi, “Corso di Dinamica delle Strutture. Dispense – parte #4” versione 1.1.2, Università degli Studi di Roma3, A.A. 2015-16;
- F.C. Ponzo, R. DiTommaso, “Dispense del corso di Elementi di Dinamica delle strutture”, Università degli Studi della Basilicata, A.A. 2014-2015;
- Polidoro Barbara, “La valutazione della vulnerabilità sismica: il caso di Pettino (AQ)”, Tesi di laurea discussa alla Facoltà di Ingegneria, Università degli Studi di Napoli Federico II, A.A. 2009/10;
- NIKER Project. New Integrated Knowledge based approaches to the protection of cultural heritage from Earthquake-induced Risk. Consultato il 21 giugno 2017, da: <http://www.niker.eu/>;
- NIKER Catalogue. Collegamento tra i meccanismi di collasso indotti dal terremoto, in base alle tipologie e i materiali con le tecniche d'intervento. Consultato il 21 giugno 2017, da: <https://niker.dicea.unipd.it/>;
- NAFEMS, “home page”. Consultata il 20 giugno 2017, da: <https://www.nafems.org/>;
- NAFEMS, “about NAFEMS”. Consultata il 20 giugno 2017, da: <https://www.nafems.org/about/>;
- NAFEMS, “History”. Consultata il 25 giugno 2017, da [https://www.nafems.org/about/about\\_nafems/history/](https://www.nafems.org/about/about_nafems/history/);
- NAFEMS, “Benchmark Magazine”. Consultata il 25 giugno 2017, da <https://www.nafems.org/publications/benchmark/>;
- NAFEMS Italia, “home page”. Consultata il 20 giugno 2017, da <https://www.nafems.org/about/regional/italia/>;
- NAFEMS Italia, “Attività di NAFEMS”. Consultata il 20 giugno 2017, da <https://www.nafems.org/about/regional/italia/attivita/>;
- NAFEMS Italia, “Professional Development – The Changing Role of Analysis”. Consultata il 25 giugno 2017, da [https://www.nafems.org/about/about\\_nafems/professional\\_dev/](https://www.nafems.org/about/about_nafems/professional_dev/);



-TNO Diana 10.1, “User Manual – Release Note”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-TNO Diana 10.1, “User Manual – Contents”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-TNO Diana 10.1, “User Manual – Element Libray”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-TNO Diana 10.1, “User Manual – Material Library”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-TNO Diana 10.1, “User Manual – Analysis Procedure”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-TNO Diana 10.1, “User Manual – Theory”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-TNO Diana 10.1, “User Manual – Verification Report”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-TNO Diana 10.1, “User Manual – Analysis Examples”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-TNO Diana 10.1, “User Manual – Geotchnical Analysis”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-TNO Diana 10.1, “User Manual – Concrete & Masonry Analysis”. Consultato il 25 giugno 2017, da <https://dianafea.com/manuals/d101/Diana.htm>;

-P. Dasdia, “L’analisi sismica degli edifici in muratura” INGEGNERIA SISMICA, anno:1994 vol: 11  
iss:1 pag:32;

- G. Magenes, D. Bolognini, C. Braggio (A cura di), “Metodi semplificati per l’analisi sismica non lineare di edifici in muratura”, CNR-Gruppo Nazionale per la Difesa dai Terremoti – Roma, 2000, 99 pp.;
- A. Borri, G. Cangi, A. De Maria, “Caratterizzazione meccanica delle murature (anche alla luce del recente sisma in Emilia) e interpretazione delle prove sperimentali a taglio”, Atti del XV congresso ANIDIS 2013;
- The Brick Industry Association, “Technical Notes 3A – Brick Masonry Material Properties”, December 1992;
- S.T.A. Data, “Verifica sismica di strutture in muratura: confronto tra differenti procedure di calcolo”, consultato il 29 giugno 2017, da <http://www.stadata.com/3Muri/documenti/ConfrontoModelliCalcoloMuratura.pdf>;
- S. P. Narayanan, M. Sirajuddin, “Properties of Brick Masonry for FE modeling”, American Journal of Engineering Research (AJER), Volume 1-2013, pp. 06-11;
- B. Calderoni, E. A. Cordasco, A. Sandoli, V. Onotri, G. Tortoriello, “Problematiche di modellazione strutturale di edifici in muratura esistenti soggetti ad azioni sismiche in relazione all’utilizzo di software commerciali”, intervento presso la conferenza ANIDIS 2015, L’Aquila;
- G. Rizzano, R. Sabatino, G. Torello, “Un nuovo modello a telaio equivalente per l’analisi statica non lineare di pareti in muratura”, intervento presso la conferenza “ANIDIS 2011 - Italian National Conference on Earthquake Engineering”, Bari (Italy) settembre 2011;
- Giuffrè A., “Lecture sulla meccanica delle murature storiche” Ed. Kappa, Roma, 1991.



# **12. indice delle figure**

Figura 1 - buildingSMART. Logo.

Figura 2 - NAFEMS. Logo.

Figura 3 - buildingSMART. Logo.

Figura 4 - schema interscambio dati IFC.

Figura 5 - ISO. Logo.

Figura 6 - buildingSMART. Schema esplicativo.

Figura 6 - Schema d'esempio EXPRESS-G.

Figura 7 - EXPRESS-G. Entity.

Figura 8 - EXPRESS-G. Abstract entity e subentity.

Figura 9 - EXPRESS-G. Abstract superentity, superentity e subentity.

Figura 10 - EXPRESS-G. Attributes.

Figura 11 - EXPRESS-G. Simple Data type.

Figura 12 - EXPRESS-G. Enumerated Type.

Figura 14 - EXPRESS-G. Defined Type.

Figura 15 - EXPRESS-G. Select Type.

Figura 16- EXPRESS-G. Attributes con cardinalità.

Figura 17 - EXPRESS-G. Attribute e inverse attribute.

Figura 18 - EXPRESS-G. External reference, ONTO THIS PAGE.

Figura 19 - EXPRESS-G. External reference, ONTO ANOTHER PAGE.

Figura 20 - EXPRESS-G. Schema reference.

Figura 21 - IFC. Definizione dei livelli gerarchici e dei domini.

Figura 22 - IFC SAD. Elementi e connessioni strutturali e loro relazioni.

Figura 23 - IFC SAD. Elementi strutturali mono e bidimensionali.

Figura 24 - IFC SAD. Relationship schema.

Figura 25 – Spiegazione lunghezza della connessione. Cfr. buildingSMART, “IFC4 - Add.2; 7.7.3.2 IfcRelConnectsStructuralMember”.

Figura 26 - IFC SAD. Definizione della direzione dei carichi.

Figura 27 - IFC SAD. Modello analitico, carichi e combinazioni di carico, risultati.

Figura 28 - IFC SLR. Carichi statici e loro combinazione. Forze puntuali, lineari e superficiali.

Figura 29 - IFC SLR. Carichi statici. Spostamenti impressi e carichi di temperatura.

Figura 30 - IFC SLR. Carico di temperatura. Specificazione variazioni di temperatura generanti azioni.

Figura 31 - IFC SLR. Specificazioni dei vincoli.

Figura 32 - IFC SLR. Condizione di slittamento e di rottura dei vincoli.

Figura 33 - IFC SED. Barre e reti d'armatura.

Figura 34 - IFC SED. Tipi di barre e reti d'armature.

Figura 35 - NAFEMS. Logo.

Figura 36 - NAFEMS Italia. Logo.

Figura 37 - NAFEMS. Locandina NWC 2017.

Figura 38 - Elementi finiti monodimensionali. Nodi e punti di integrazione.

Figura 39 - Elementi finiti monodimensionali. Gradi di libertà.

Figura 40 - Elementi finiti bidimensionali. Nodi.

Figura 41 - Elementi finiti tridimensionali. Nodi.

Figura 42 - Elementi finiti tridimensionali. Gradi di Libertà.

Figura 43 - Elementi finiti. Congruenza degli elementi.

Figura 44 - Elementi finiti tridimensionali. Deformazioni.

Figura 45 - Dinamica delle strutture. Elemento a un grado di libertà.

Figura 46 - Dinamica delle strutture. Tipologia di forzanti esterne.

Figura 47 - Proposta ST-7. Carichi dinamici.

Figura 48 - Proposta ST-7. Modello agli elementi finiti.

Figura 49 - Proposta ST-7. Azioni sul modello agli elementi finiti.

Figura 50 - Proposta ST-7. Risultati di analisi strutturali.

Figura 51 - Proposta ST-7. Risultati di analisi sul modello agli elementi finiti.

Figura 52 - Relationship Classes per interfacciare il modello FEM. Di nuovo in arancione le nuove *classes*, in giallo quelle già esistenti.

Figura 53 - Stato di fatto e proposta di modifica Softing srl dello SAD

Figura 54 - Proposta Softing srl per l'implementazione azioni dinamiche

Figura 55 - Proposta Softing srl per l'implementazione dei risultati di dinamica

Figura 56 - Sviluppo nel tempo di IFC e differenti versioni.

Figura 57 - Proposta modello elementi finiti. Nodi, punti di integrazione e struttura.

Figura 58 -Proposta modello agli elementi finiti. Elemento finito.

Figura 59 - Proposta modello agli elementi finiti. Relazionalità di questo.

Figura 60 - Proposta modello elementi finiti. Risultati modello.

Figura 61 - Proposta azioni dinamiche.

Figura 62 - Proposta risultati analisi dinamica. Fattori di smorzamento, fattori di partecipazione, autovalori.

Figura 63 - Proposta risultati analisi dinamica. Autovettori e relativa normalizzazione.

Figura 64 - Proposta modifica. Inserimento in IFC delle azioni e dei risultati proposti.

Figura 65 - Proposta modifica. Condizioni dinamiche dei vincoli.

Figura 66 - IfcAnalysisTheoryTypeEnum.

Figura 67 - Proposta modifica. Elemento strutturale analitico tridimensionale. Solita convenzione, in arancione la nuova *entity*, il resto è già presente in IFC.

Figura 68 - IfcAnalysisTheoryTypeEnum.

Figura 69 - Proposta modifica. Oggetti del consolidamento.

Figura 70 - IFC SED. Barre e reti d'armatura.

Figura 71- Proposta modifica. Descrizione materiali fibrorinforzati. Al solito, in arancione le nuove *entities*, in giallo quelle già esistenti; in verde scuro i nuovi *types*.

Figura 72 - Proposta modifica. Descrizione tipi di materiali fibrorinforzati. Al solito, in arancione le nuove *entities*, in giallo quelle già esistenti; in verde scuro i nuovi *types*.

Figura 73 - IFC. Simbologia Property Set.

Figura 74 - Proposta modifica. Modello a telaio equivalente.

Figura 75 - Proposta modifica. Relazionalità modello a telaio equivalente.

Figura 76 - Screenshot ARCHICAD. Vista 1 edificio Ex-Fisica Tecnica.

Figura 77- Screenshot ARCHICAD. Vista 2 edificio Ex-Fisica Tecnica.

Figura 78 - Screenshot Solibri Model Checker. Proprietà personalizzate in IFC.

Figura 79 - Screenshot Solibri Model Checker. Vista 1 edificio Ex-Fisica Tecnica.



Figura 80 - Screenshot ARCHICAD. Vista muro isolato facciata est.

Figura 81 - Screenshot Solibri Model Checker. Vista muro isolato facciata est.

Figura 82 - Screenshot pannello importativo TNO Diana.

Figura 83 - Screenshot TNO Diana. Muro solido importato.

Figura 84- Screenshot pannello importativo TNO Diana.

Figura 85 - Screenshot TNO Diana. Muro bidimensionali importato.

Figura 86 - Screenshot TNO Diana. Report importazione.

Figura 87 – Screenshot TNO Diana. Proprietà elementi analitici.

Figura 88 - Screenshot TNO Diana. Caratterizzazione materiali.

Figura 89 - Screenshot TNO Diana. Caratterizzazione muratura.

Figura 90 - Screenshot TNO Diana. Caratterizzazione calcestruzzo.

Figura 91 - Screenshot TNO Diana. Definizione dei carichi.

Figura 92 - Screenshot TNO Diana. Definizione combinazioni di carico.

Figura 93 - Screenshot TNO Diana. Definizione condizioni di vincolo.

Figura 94 - Screenshot TNO Diana. Generazione della mesh.

Figura 95 - Screenshot TNO Diana. Verifica carichi e vincoli nella mesh.

Figura 96 - Screenshot TNO Diana. Configurazione deformata da combinazione di carico n°1.

Figura 97 - Screenshot da TNO Diana. Spostamento totale da combinazione di carico n°1.

Figura 98 - Screenshot TNO Diana. Deformazioni verticali combinazione di carico n°1.

Figura 99 - Screenshot TNO Diana. Deformazioni taglianti combinazione di carico n°1.

Figura 100 - Screenshot TNO Diana. Configurazione deformata da combinazione di carico n°2.

Figura 101 - Screenshot TNO Diana. Spostamento orizzontale da combinazione di carico n°2.

Figura 102 - Screenshot da TNO Diana. Spostamento verticale da combinazione di carico n° 2.

Figura 103 - Screenshot da TNO Diana. Deformazioni orizzontali da combinazione di carico n°2.

Figura 104 - Screenshot da TNO Diana. Deformazioni orizzontali da combinazione di carico n°2.

Figura 105 - Screenshot TNO Diana. Configurazione deformata da combinazioni di carico n°3.

Figura 106 - Screenshot TNO Diana. Spostamento orizzontale da combinazione di carico n°3.

Figura 107 - Screenshot TNO Diana. Spostamento verticale da combinazione di carico n°3.

Figura 108 - Screenshot TNO Diana. Deformazioni verticali da combinazione di carico n°3.

Figura 109 - Screenshot TNO Diana. Tensioni verticali da combinazioni di carico n°3.

Figura 110 - Screenshot TNO Diana. Determinazione autovalore e deformazione del primo modo di vibrare.

Figura 111 - Screenshot TNO Diana. Determinazione autovalore e deformazione del secondo modo di vibrare.

Figura 112 - Screenshot TNO Diana. Determinazione autovalore e deformazione del terzo modo di vibrare.

Figura 113 - Screenshot TNO Diana. Risultati in deformazione dell'analisi modale del primo modo di vibrare.

Figura 114 - Screenshot TNO Diana. Risultati dell'analisi dinamica in spostamento dalla sovrapposizione di tipo SRSS.

Figure 115 - Screenshot Softing Nòlian. Unità di misura.

Figure 116 - Screenshot Softing Nòlian. Archivio materiali.

Figure 117 - Screenshot Softing Nòlian. Proprietà Calcestruzzo.

Figure 118 - Screenshot Softing Nòlian. Proprietà Muratura.

Figure 119 - Screenshot Softing Nòlian. Generazione mesh di calcolo.

Figure 120 - Screenshot Softing Nòlian. Assegnazione vincoli esterni.

Figure 121 - Screenshot Softing Nòlian. Assegnazione pesi propri portati.

Figure 122 - Screenshot Softing Nòlian. Assegnazione carichi sismici. Rispettivamente 340, 300 e 155 kN.

Figure 123 - Screenshot Softing Nòlian. Assegnazione carichi accidentali portati.

Figure 124 - Screenshot Softing Nòlian. Creazione combinazioni di carico.

Figure 125 - Screenshot Softing Nòlian. Configurazione deformata della combinazione n° 1.

Figure 126 - Screenshot Softing Nòlian. Deformazioni della configurazione n° 1.

Figure 127 - Screenshot Softing Nòlian. Sforzi principali di compressione della combinazione di carico n° 1.

Figure 128 - Screenshot Softing Nòlian. Configurazione deformata della configurazione di carico n° 2.

Figure 129 - Screenshot Softing Nòlian. Sforzi principali di compressione della configurazione n° 1.

Figure 130 - Screenshot Softing Nòlian. Modo fondamentale di vibrare e corrispondenti spostamenti.

# **13. ringraziamenti**

Al termine di questo lungo percorso durato sei anni sono molti i ringraziamenti che vorrei fare.

Prima di tutto ai miei genitori, che mi hanno dato la vita e, metaforicamente, anche più di una volta. Grazie. E grazie perché mi avete supportato davvero tanto. Anche economicamente, perché l'università, e soprattutto architettura, non è stata molto frugale. Grazie perché mi avete supportato anche quando ero teso per gli esami e tutto quanto. Grazie che ci siete stati quando ne avevo bisogno. E grazie perché so che posso contare su di voi. Magari a volte vi ho delusi ma spero che siate fieri di me almeno un po'.

Grazie ai miei fratelli, quelli veri. A cui ho rotto tanto le scatole in questi anni, anche prima dell'università. Grazie per tante cose, ma soprattutto perché non ho bisogno che ce lo diciamo per capire e sapere che ci vogliamo bene.

Grazie ai nonni, ma anche ai bisnonni - almeno quelli che ho conosciuto - e tutti i miei parenti. Perché mi hanno insegnato davvero tanto, anche loro hanno contribuito a fare di me ciò che sono adesso. Nonostante, forse, il risultato non sia dei migliori. Grazie, sinceramente, per ciò che mi avete dato, anche se non sono mai riuscito ad esprimerlo bene come vorrei.

Grazie ai miei fratelli, quegli altri. Perché loro sono pesanti e io sono pesante per loro. Grazie perché se ne ho bisogno avete la capacità di darmi consigli - anche se magari non li chiedo - ma anche quella di stare in silenzio quando serve, cioè non quando penso di averne bisogno. Grazie perché anche su di voi so di poter contare, alla fine.

Grazie a chi in questi anni mi ha fatto compagnia e vorrebbe farmene ancora, a chi non vorrebbe affatto, a chi, insomma non c'è più. Nel bene e nel male mi hanno insegnato davvero qualcosa - a livello umano almeno - e mi hanno voluto bene come io ne ho voluto a loro.

Grazie ai miei amici, con cui ho affrontato questo percorso universitario, anche se magari con qualcuno abbiamo perso un po' i contatti. Con qualcuno le nostre strade si sono separate magari, ma è stato bello studiare e stare con voi. Tutti voi. Avete reso più piacevole il mio percorso e il vostro aiuto è stato tantissime volte prezioso. Anche voi mi avete lasciato qualcosa e vi ricorderò.

Grazie a tutti i professori che ho incontrato in questi anni. Sinceramente a tutti quanti. Perché mi hanno fatto appassionare a tante cose, anche cose che non pensavo mi sarebbero mai potute piacere, né che mi sarebbero mai potute interessare. Grazie a tutti, perché mi hanno fatto appassionare e amare cose che non pensavo mi potessero piacere così tanto. Grazie a tutti, anche solo perché umanamente mi avete insegnato che dietro la cattedra c'è una persona. Grazie anche perché mi avete mostrato cos'è un ingegnere, un architetto, un progettista, uno strutturista e molto altro ancora. Sarà scontato, ma grazie a tutti perché più di tante altre persone, per forza di cose, mi avete insegnato davvero tanto di tante cose.

Grazie in particolar modo alla professoressa Valluzzi, che ha accettato di condurmi attraverso l'enorme avventura che è stata questa tesi e che ha creduto in questo progetto, per quanto sgangherato potesse apparire, magari. Grazie per la disponibilità e la pazienza avute, ma anche per la passione trasmessami.

Grazie all'instancabile e immarcescibile ingegner Carlo Zanchetta, senza cui non avrei potuto nemmeno di pensare di mettere in piedi questa tesi e laurearmi. Grazie perché la sua guida è stata più che fondamentale, grazie perché anche lui mi ha sostenuto tanto e perché ha un'energia fuori dal comune. Insieme ringrazio anche Giulio Pasqualotto, alla cui tesi la mia deve tanto, sebbene lui magari non lo sappia. Un grazie perché quando non ne capivo niente di niente nel suo lavoro trovavo la luce. Grazie anche ad Alberto Calabria, caduto per la patria con questa tesi. Grazie della disponibilità, nonostante le rocambolesche richieste, perché si è ritrovato catapultato in questo mondo e mi ha aiutato, con impegno e sincero interesse nonostante tutto.

Grazie allo staff di Softing srl, a Roberto Spagnuolo, ad Amedeo Farello e Giuseppe Pascucci. Perché hanno dato tanto anche loro a questa tesi, con la loro conoscenza di FEM e meccanica computazionale, con la loro disponibilità a utilizzare i loro software. In così poco tempo mi hanno insegnato così tanto.

Grazie a tante altre persone, che non ricordo adesso. Ma per farla breve, grazie a chi mi è stato vicino. Vi porterò tutti nel cuore.

Grazie, infine, a chi leggerà questa tesi. Oppure, come anche io ho fatto spesso, grazie anche a chi leggerà solo i ringraziamenti.