

Univerza  
v Ljubljani

Fakulteta za  
*gradbeništvo in  
geodezijo*



*PODIPLOMSKI ŠTUDIJ  
GRADBENIŠTVA  
KONSTRUKCIJSKA SMER*

*DOKTORSKI ŠTUDIJ*

Kandidat:

**TOMAŽ PAZLAR, univ. dipl. inž. grad.**

**PRESLIKAVE MED ARHITEKTURNIMI IN  
RAČUNSKIMI ASPEKTI V INFORMACIJSKIH  
MODELIH ZGRADB**

Doktorska disertacija štev.: 186

**MAPPING BETWEEN ARCHITECTURAL AND  
STRUCTURAL ASPECTS IN THE BUILDING  
INFORMATIONS MODELS**

Doctoral thesis No.: 186

Temo doktorske disertacije je odobrila Komisija za doktorski študij  
na svoji 28. seji, dne 15. septembra 2006, po pooblastilu  
s 7. seje Senata Univerze v Ljubljani z dne 27. junija 2006  
in imenovala mentorja prof.dr. Žiga Turka.

Ljubljana, 03. oktober 2008

Univerza  
v Ljubljani

Fakulteta za  
*gradbeništvo in  
geodezijo*



Komisijo za oceno ustreznosti teme doktorske disertacije v sestavi  
prof.dr. Žiga Turk  
prof.dr. Janez Duhovnik  
prof.dr. Danijel Rebolj (UM, FG)

je imenoval Senat Fakultete za gradbeništvo in geodezijo na 9. redni seji dne  
19. aprila 2006.

Komisijo za oceno doktorske disertacije v sestavi  
prof.dr. Žiga Turk  
prof.dr. Janez Duhovnik  
prof.dr. Danijel Rebolj (UM, FG)

je imenoval Senat Fakultete za gradbeništvo in geodezijo na 19. redni seji dne  
02. julija 2008.

Komisijo za zagovor doktorske disertacije v sestavi  
prof. dr. Bojan Majes, dekan, predsednik  
prof.dr. Žiga Turk  
prof.dr. Janez Duhovnik  
prof.dr. Danijel Rebolj (UM, FG)

je imenoval Senat Fakultete za gradbeništvo in geodezijo na 20. redni seji dne  
24. septembra 2008.

Univerza  
v Ljubljani

Fakulteta za  
*gradbeništvo in  
geodezijo*



## IZJAVA O AVTORSTVU

Podpisani **TOMAŽ PAZLAR, univ. dipl. inž. grad.**, izjavljam, da sem avtor doktorske disertacije z naslovom: »**PRESLIKAVE MED ARHITEKTURNIMI IN RAČUNSKIMI ASPEKTI V INFORMACIJSKIH MODELIH ZGRADB**«.

Ljubljana, 03. oktober 2008

.....  
(podpis)

## **ERRATA**

<b>Stran z napako</b>	<b>Vrstica z napako</b>	<b>Namesto</b>	<b>Naj bo</b>
-----------------------	-------------------------	----------------	---------------

---



## BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

**UDK:** 659.2:004:624:(043.3)

**Avtor:** Tomaž Pazlar

**Mentor:** prof. dr. Žiga Turk

**Naslov:** Preslikave med arhitekturnimi in računskimi aspekti v informacijskih modelih zgradb

**Obseg in oprema:** 218 str., 14 pregl., 5 priv. pregl., 46 sl., 36 priv. sl., 1 en.

**Ključne besede:** gradbena informatika, informacijski modeli zgradb – BIM, temeljni razredi za industrijo – IFC, arhitekturni modeli, računski modeli, preslikave, hevristika.

### Izveček:

Doktorska disertacija obravnava področje modeliranja produktov industrije ki oblikuje grajeno okolje. Kompleksnost panoge v kateri ima običajno vsak udeleženec svoj pogled na zgradbo oz. svoj model zahteva natančno opredelitev terminologije, izdelan kronološko razvojni opis informacijskega modeliranja pa omogoča lažje razumevanje trenutnega stanja na obravnavanem področju.

Naloga temelji na trenutno najbolj aktualnem informacijskem modelu zgradb, to je na temeljnih razredih za industrijo (IFC – Industry Foundation Classes). Posebej podrobno sta obravnavana domensko specifična podmodela arhitekture in računske analize. Ker gre za podmodela enotnega informacijskega modela zgradb je smiselna preučitev relacij med njima.

Predstavljeno delo zagovarja hipotezo da je problem prirejanja med arhitekturnimi in računskimi modeli znotraj informacijskega modela zgradb IFC možno poenostaviti ter tudi delno avtomatizirati. V nalogi se navedeno hipotezo potrjuje na primeru prirejanja geometrije ter zunanjih vplivov na konstrukcijo.

Izdelan je bil tudi demonstracijski prototip ki načeloma potrjuje pravilnost podane hipoteze, hkrati pa opozarja na nezmožnost delovanja predlaganega sistema v praksi, predvsem zaradi nezrelosti IFC specifikacije ter njenih implementacij.



## **BIBLIOGRAPHIC – DOCUMENTALISTIC INFORMATION**

- UDC:** 659.2:004:624:(043.3)
- Author:** Tomaž Pazlar
- Supervisor:** prof. dr. Žiga Turk
- Title:** Mapping between architectural and structural aspects in the building information models
- Notes:** 218 pg., 14 tabl., 5 adop. tabl., 46. fig., 36 adop. fig., 1 eq.
- Keywords:** construction informatics, Building Information Model – BIM, Industry Foundation Classes – IFC, architectural models, structural models, mapping, heuristic.

### **Abstract:**

Presented thesis is focused on the building information modelling in the AEC-FM (Architecture, Engineering, Construction and Facility Management) sector. The complexity of discussed brunch reflects in number of domain specific models and in not clearly defined terminology. Therefore the first part of presented thesis deals with modelling terminology and with chronological modelling development description.

Further work is focused on the Industry Foundation Classes (IFC), the most popular AEC-FM specification. Presented research is limited on the domain specific architectural and structural submodels and on corresponding relations between them. This thesis defends the hypothesis that presented problem can be simplified and partly automated if IFC model and appropriate computer encoded heuristic rules are used. IFC based information server for collaborative building design between architects and structural engineers with semi – automatic geometry mapping between sub models and with imposed load assignments is proposed.

A demonstration prototype partly proves the regularity of proposed hypothesis. However, the prototype also points out inability of its implementation in practice, mainly due to immaturity of IFC specification and its implementations.





## **ZAHVALA**

Najprej bi se rad zahvalil prof. dr. Žigi Turku za vodenje, usmerjenje in pomoč pri izdelavi disertacije ter za veliko mero raziskovalne svobode.

Pričujoča disertacija med drugim predstavlja tudi rezultat spodbud mentorja in somentorja na univerzitetnem študiju gradbeništva, za kar velja zahvala prof. dr. Janezu Duhovniku in viš. pred. mag. Vidu Maroltu.

Za koristne nasvete in pomoč se zahvaljujem tudi vsem sodelavcem Katedre za gradbeno informatiko: asist. dr. Tomu Cerovšku, doc. dr. Matevžu Dolencu, doc. dr. Iztoku Kovačiču, dr. Etielu Petrinji, mag. Vladu Stankovskemu, mag. Jerneju Trnkoczyju, Damjanu Murnu, uni. dipl. ing. gr. ter Mateji Šmid Hribar, univ. dipl. fil.

Posebna zahvala velja sodelavcu Robertu Klincu, uni. dipl. ing. gr. za vso pomoč pri izdelavi prototipa.

Rad bi se rad zahvalil tudi Ministrstvu za visoko šolstvo, znanost in tehnologijo Republike Slovenije za financiranje raziskovalnega dela v okviru programa mladi raziskovalci.

Skrbni jezikovi pregled dela je opravila Romana Fekonja, za kar se ji najlepše zahvaljujem.

Hvaležen sem tudi staršem za vso pomoč med študijem ter prijateljem za vso vzpodbudo.



## KAZALO VSEBINE

<b>1</b>	<b>UVOD.....</b>	<b>1</b>
1.1	OPIS OŽJEGA PODROČJA RAZISKAVE .....	1
1.2	MOTIVACIJA RAZISKOVANJA IN OPIS PROBLEMA.....	4
1.3	TEMELJNA HIPOTEZA.....	6
1.4	STRUKTURA NALOGE .....	7
<b>2</b>	<b>PREGLED KLJUČNIH PODROČIJ POVEZANIH Z RAZISKAVO.....</b>	<b>9</b>
2.1	INFORMACIJSKO MODELIRANJE KONSTRUKCIJ.....	9
2.1.1	<i>Terminologija</i> .....	9
2.1.2	<i>Informacijsko modeliranje</i> .....	10
2.1.2.1	Informacijski modeli .....	10
2.1.2.2	Konceptualno modeliranje .....	12
2.1.2.3	Jeziki za informacijsko modeliranje.....	14
2.1.2.4	Stopnja interoperabilnosti .....	16
2.1.2.5	Produktni podatkovni model in podatkovni model.....	17
2.1.2.6	Informacijski model zgradbe.....	18
2.1.2.7	Informacijski model zgradbe – geometrijski model .....	20
2.2	STEP .....	23
2.2.1	<i>Potrebe industrije</i> .....	23
2.2.1.1	Identifikacija potreb industrije .....	23
2.2.1.2	Standardizacijska iniciativa.....	25
2.2.2	<i>STEP arhitektura</i> .....	26
2.2.2.1	Osnovni gradniki.....	26
2.2.2.2	Vsebinska delitev standarda STEP:.....	27
2.2.2.3	Metode opisa.....	30
2.2.2.4	Metode implementacije .....	38
2.2.2.5	Integrirani generični viri.....	40
2.2.2.6	Aplikacijski generični viri .....	42
2.2.2.7	Aplikacijski protokoli (AP).....	44
2.3	INFORMACIJSKI MODEL ZGRADB IFC .....	46
2.3.1	<i>Splošna predstavitev modela zgradb IFC</i> .....	46
2.3.1.1	Zgodovina modela zgradb IFC.....	46
2.3.1.2	Obseg IFC modela zgradb.....	50
2.3.1.3	Širitev modela zgradb IFC .....	51
2.3.1.4	Izmenjava informacij in implementacije .....	52
2.3.1.5	»Pogledi na model«.....	53
2.3.1.6	»Implementacijski dogovor«.....	56
2.3.1.7	Implementacije modela zgradb IFC ter certificiranje .....	56
2.3.1.8	Alternativni zapis IFC specifikacije v XML obliki .....	58

2.3.2	Zgradba modela IFC.....	60
2.3.2.1	Ključne strukture modela zgradb IFC 2x3.....	63
2.3.3	Razširitev specifikacije IFC na področje statične analize konstrukcij (ST4).....	81
2.3.3.1	Projekt ST4.....	81
<b>3</b>	<b>PRESLIKAVE V INFORMACIJSKIH MODELIH ZGRADB.....</b>	<b>89</b>
3.1	DOMENSKI MODELI.....	89
3.1.1	Domenski modeli in izmenjava.....	92
3.2	PRESLIKAVE MED MODELI ZNOTRAJ INFORMACIJSKEGA MODELA ZGRADB IFC.....	93
3.2.1	Uporabni minimum.....	93
3.3	PRESLIKAVE MED ARHITEKTURNIMI IN RAČUNSKIMI MODELI V IFC SPECIFIKACIJI.....	95
3.3.1	IFC skladna programska oprema za analizo konstrukcij.....	95
3.3.2	Raziskave na področju prirejanja med pod modeli IFC specifikacije.....	97
3.3.2.1	Uporaba IFC specifikacije pri prefabriciranih armiranobetonskih konstrukcijah.....	97
3.3.2.2	Računsko modeliranje konstrukcij na podlagi IFC arhitekturnega modela.....	98
3.3.2.3	Računska analiza IFC skladnih modelov zgradb.....	100
3.3.2.4	IFC spletni strežnik za načrtovanje zgradb.....	103
3.3.3	Zahteve pri preslikavah med arhitekturnimi in računskimi modeli in scenarij uporabe.....	104
3.3.4	Pogledi na model.....	105
3.3.4.1	Pogled na model: Arhitektura – računsko modeliranje (načrtovanje) nosilne konstrukcije.....	105
3.3.4.2	Pogled na model: Računsko modeliranje (načrtovanje) nosilne konstrukcije – Analiza konstrukcije.....	115
3.3.4.3	Splošne zahteve pri specifikaciji računskega podmodela.....	116
3.4	PRIREJANJE GEOMETRIJE.....	117
3.4.1	Hevristika.....	117
3.4.2	Algoritem preslikav med arhitekturnimi modeli in modeli za analizo.....	118
3.5	PREDLAGANA REŠITEV ZAOKROŽENEGA NAČRTOVANJA ZNOTRAJ MODELA IFC.....	121
3.5.1	Temeljni principi.....	121
3.5.2	Opredelevitev problema.....	122
3.5.2.1	Vplivi na konstrukcijo.....	123
3.5.2.2	ENV 1991-1-1.....	125
3.5.2.3	Transformacija ENV 1991-1-1 v XML obliko.....	125
3.5.3	Arhitektura predlaganega sistema.....	126
<b>4</b>	<b>PROTOTIPNA IMPLEMENTACIJA PREDLAGANE REŠITVE.....</b>	<b>131</b>
4.1	PREDPOSTAVKE IN OMEJITVE.....	131
4.1.1	Predpostavke in omejitve dosega in preslikav.....	131
4.1.2	Predpostavke in omejitve programske opreme.....	133
4.1.2.1	Programska oprema za arhitekturno načrtovanje konstrukcij.....	133
4.1.2.2	Programska oprema za računsko modeliranje in analizo konstrukcij.....	134
4.1.2.3	Strežnik modela zgradb IFC.....	134

4.2	IMPLEMENTACIJA REŠITVE .....	135
4.2.1	<i>Modeliranje arhitekture</i> .....	136
4.2.2	<i>Računsko modeliranje/analiza</i> .....	139
4.2.3	<i>Zaokroženo načrtovanje</i> .....	142
4.3	TESTIRANJE IMPLEMENTIRANE REŠITVE .....	142
4.3.1	<i>Testni primeri</i> .....	143
4.3.2	<i>Ugotovitve pri testiranju</i> .....	143
4.4	PREDLAGANE IZBOLJŠAVE PROTOTIPA .....	144
<b>5</b>	<b>SKLEP</b> .....	<b>145</b>
5.1	GLAVNE UGOTOVITVE.....	145
5.2	POVZETEK BISTVENIH PRISPEVKOV NALOGE .....	147
5.3	NADALJNE RAZISKAVE.....	148
<b>6</b>	<b>POVZETEK</b> .....	<b>151</b>
<b>7</b>	<b>SUMMARY</b> .....	<b>153</b>
	<b>LITERATURA</b> .....	<b>155</b>
	<b>PRILOGA A: KRATICE, OKRAJŠAVE</b> .....	<b>165</b>
	<b>PRILOGA B: PREGLED INFORMACIJSKEGA MODELIRANJA</b> .....	<b>171</b>
B.1	Mejniki informacijskega modeliranja.....	171
B.2	Geometrijsko modeliranje .....	172
B.3	CAD sistemi .....	172
B.4	Geometrijsko modeliranje in CAD sistemi.....	173
B.5	Centralizirana izmenjava – podatkovni nivo .....	175
B.6	Informacijsko modeliranje.....	181
B.7	Razvoj informacijskih modelov zgradb .....	186



## KAZALO PREGLEDNIC

Preglednica 1: Objektno orientirani programski jeziki : Informacijsko modeliranje .....	13
Preglednica 2: Uporaba standardov za izmenjavo informacij .....	24
Preglednica 3: Karakteristike opisa informacijskega modela .....	25
Preglednica 4: Področja, pokrita z IFC specifikacijo .....	50
Preglednica 5: Pogledi na model – IFC 2x3 .....	54
Preglednica 6: Koordinacijski pogled – IFC 2x3 .....	55
Preglednica 7: Rezultati certificiranja (oktober 2007) – IFC 2x3.....	57
Preglednica 8: Preslikave v programu ETABS – uvoz v model IFC.....	96
Preglednica 9: Preslikave v programu ETABS – izvoz v model IFC.....	96
Preglednica 10: Možna stikovanja med elementi zgradb .....	118
Preglednica 11: Evrokod 1: Vplivi na konstrukcijo.....	124
Preglednica 12: Kategorije uporabe .....	132
Preglednica 13: Koristna obtežba prostorov, balkonov in stopnišč v zgradbah .....	133
Preglednica 14: Struktura DXF datoteke .....	180



**KAZALO PRIVZETIH PREGLEDNIC**

Privzeta preglednica 1: Deli standarda STEP (ISO 10303, 1992) .....	29
Privzeta preglednica 2: STEP – integrirani viri (ISO 10303-42, 1992) .....	41
Privzeta preglednica 3: Entitete v ISO 10303-del 42 (ISO 10303-42, 1992).....	42
Privzeta preglednica 4: Splošno veljavni integrirani aplikacijski viri – deli 1xx (ISO 10303-42, 1992).....	43
Privzeta preglednica 5: Aplikacijsko interpretirani konstrukti – deli 5xx (ISO 10303-42, 1992) .....	44

## KAZALO SLIK

Slika 1: Struktura STEP aplikacijskih protokolov .....	27
Slika 2: Entitete – atributi.....	36
Slika 3: Odnosi med entitetami .....	36
Slika 4: Nadtipi in podtipi .....	37
Slika 5: Inverzna pravila.....	37
Slika 6: EXPRESS-G shema na več straneh – sklicevanje.....	38
Slika 7: EXPRESS shema – STEP fizična datoteka.....	39
Slika 8: Zgodovina modela zgradb IFC.....	49
Slika 9: Število entitet v posamezni IFC različici.....	49
Slika 10: Širitev modela zgradb IFC .....	52
Slika 11: Nivoji interoperabilnosti IFC modela.....	54
Slika 12: Arhitekturni model – računski model: relacije med modeli .....	90
Slika 13: Načrtovanje zgradb – relacije med pod modeli v modelu zgradb IFC .....	91
Slika 14: Uporabni minimum .....	94
Slika 15: Enotni MKE računski model.....	98
Slika 16: Grafikon poteka.....	99
Slika 17: Pod modeli v računski analizi IFC skladnih modelov zgradb .....	100
Slika 18: Model povezav – objekti povezav, objekti razlike .....	101
Slika 19: Objekta $W_i$ , $W_j$ , stična ploskev $f_{ij}$ in vektor normale $n_{ij}$ .....	102
Slika 20: A-RM pogled: IfcProject.....	109
Slika 21: A-RM pogled: IfcSite.....	110
Slika 22: A-RM pogled: IfcBuilding .....	110
Slika 23: A-RM pogled: IfcBuildingStorey.....	111
Slika 24: A-RM pogled: IfcBeam.....	111
Slika 25: A-RM pogled: IfcColumn .....	112
Slika 26: A-RM pogled: IfcSlab.....	113
Slika 27: A-RM pogled: IfcWall .....	114

Slika 28: Stikovanje elementov .....	119
Slika 29: Algoritem za ugotavljanje stikov.....	120
Slika 30: XML zapis koristne obtežbe.....	126
Slika 31: Diagram (upo)rabe.....	127
Slika 32: Diagram poteka predlaganega zaokroženega načrtovanja.....	129
Slika 33: Arhitektura IFC strežnika modela zgradb.....	130
Slika 34: Implementiran prototip.....	135
Slika 35: Arhitekturno načrtovanje (Graphisoft Archicad 10).....	136
Slika 36: Strežnik modela zgradb IFC – nalaganje arhitekturnega modela .....	137
Slika 37: Arhitekturni model z elementi zgradb (IfcBeam, IfcColumn, IfcSlab) in računski model (IfcStructuralCurveMemeber, IfcStructuralSurfaceMember).....	138
Slika 38: IFC vmesnik programa AxisVM za uvoz informacijskega modela zgradb.....	139
Slika 39: Uvoz arhitekturnega modela v program AxisVM .....	140
Slika 40: Uvoz računskega modela v program AxisVM .....	141
Slika 41: Komunikacijske revolucije in gradbeništvo .....	171
Slika 42: Neposredna in centralizirana izmenjava informacij .....	175
Slika 43: Pregled geometrijskega modeliranja.....	178
Slika 44: Troslojna ANSI-SPARC arhitektura .....	182
Slika 45: Informacijski modeli v AEC-FM sektorju.....	189
Slika 46: Komunikacije v modelu VEGA .....	217

## KAZALO PRIVZETIH SLIK

Privzeta slika 1: Različni opisi ne opišejo modela v celoti, temveč nakazujejo njegovo kompleksnost (Björk, 1995).....	20
Privzeta slika 2: Geometrijski model / Informacijski model, hipotetični atributi ter razmerja med entitetami (Khemlani, 2004).....	22
Privzeta slika 3: Elementi standarda STEP (ISO 10303, 1992).....	29
Privzeta slika 4: Seznam EXPRESS-G simbolov (ISO 10303, 1992).....	35
Privzeta slika 5: Shema specifikacije IFC2x3 (IAI, 2008).....	60
Privzeta slika 6: Entitete v IFC2x3 (druga stopnja podrobnosti) (IAI, 2008).....	67
Privzeta slika 7: Koncept predstavitve oblike (IAI, 2008).....	71
Privzeta slika 8: Entitete geometrijske predstavitve (IAI, 2008).....	72
Privzeta slika 9: Predstavitev oblike s pomočjo površin (IAI, 2008).....	73
Privzeta slika 10: Predstavitev oblike s pomočjo trdnih teles (IAI, 2008).....	75
Privzeta slika 11: Hierarhija elementov zgradb (IAI, 2008).....	77
Privzeta slika 12: Primeri standardnih (zgoraj) in specifičnih sten (spodaj) (IAI, 2008).....	78
Privzeta slika 13: Scenariji uporabe znotraj domene analize konstrukcij (Weise et al., 2003).....	82
Privzeta slika 14: Interoperabilnost s preostalimi domenami (Weise et al., 2003).....	83
Privzeta slika 15: EXPRESS-G shema razširitve ST4 (IAI, 2008).....	85
Privzeta slika 16: Elementi zgradb: Elementi računskega modela (Weise et al., 2003).....	86
Privzeta slika 17: IGES tekstovna datoteka – opis dveh točk, daljic in krožnih lokov (IGES5, 2007).....	177
Privzeta slika 18: Taksonomija BSM (Liebich, Wix, 2002).....	190
Privzeta slika 19: PDU in podtipi modela GARM (Ghang, 2004).....	192
Privzeta slika 20: GARM – hamburger diagram (Ekholm, 1994).....	192
Privzeta slika 21: Aspektni modeli v projektu COMBINE (Liebich, Wix, 2002).....	194
Privzeta slika 22: Sloji v večslojnih integriranih sistemih (Augenbroe, 1995).....	197
Privzeta slika 23: CIS2 in podmodeli (Eastman, 2000).....	198
Privzeta slika 24: CIS2 model za analizo (EXPRESS-G) (Eastman, 2000).....	199
Privzeta slika 25: CIS2 model za načrtovanje (EXPRESS-G) (Eastman, 2000).....	200
Privzeta slika 26: CIS2 model za proizvodnjo (EXPRESS-G) (Eastman, 2000).....	201

Privzeta slika 27: CIS2 – preslikave med pod modeli (EXPRESS-G) (Eastman, 2000) .....	202
Privzeta slika 28: Ogradje modela RATAS (EXPRESS-G) (Björk, 1995) .....	204
Privzeta slika 29: Tipski objekti v informacijskem modelu RATAS (Björk, 1995) .....	205
Privzeta slika 30: Tipski objekti v informacijskem modelu ATLAS (Liebich, Wix, 2002) .....	207
Privzeta slika 31: Del arhitekture ATLAS demonstratorja (Tolman, 1999) .....	208
Privzeta slika 32: Ogradje modela COMBI (Katranuschkov, Scherer, 1996) .....	210
Privzeta slika 33: Globalna arhitektura aplikacijskega protokola 236 (Liebich, Wix, 2002) .....	212
Privzeta slika 34: Arhitektura BCCM – predlog in implementacija (Liebich, Wix, 2002) .....	213
Privzeta slika 35: Ogradje modela BCCM (Froese, 1996) .....	214
Privzeta slika 36: Osnovni gradniki platforme VEGA (Debras et al., 1997) .....	215

## LIST OF TABLES

Table 1: Object oriented languages : Information modelling .....	13
Table 2: The use of information exchange standards .....	24
Table 3: The characteristics of information model description .....	25
Table 4: AEC-FM areas included in IFC specification .....	50
Table 5: Model view – IFC 2x3 .....	54
Table 6: Coordination view – IFC 2x3 .....	55
Table 7: Certification results (October 2007) – IFC 2x3 .....	57
Table 8: Mapping in ETABS – IFC model import .....	96
Table 9: Mapping in ETABS – IFC model export.....	96
Table 10: Possible connections between building elements.....	118
Table 11: Eurocode 1: Actions on structures.....	124
Table 12: Categories of use .....	132
Table 13: Imposed loads on floors, balconies and stairs in buildings .....	133
Table 14: DXF file structure.....	180

**LIST OF ADOPTED TABLES**

Adopted table 1: STEP parts (ISO 10303, 1992).....	29
Adopted table 2: STEP – integrated resources (ISO 10303-42, 1992) .....	41
Adopted table 3: Entities in ISO 10303-part 42 (ISO 10303-42, 1992) .....	42
Adopted table 4: Universally valid application models – 1xx series (ISO 10303-42, 1992).....	43
Adopted table 5: Application Interpreted Constructs – 5xx series (ISO 10303-42, 1992) .....	44

## LIST OF FIGURES

Figure 1: STEP application protocol structure .....	27
Figure 2: Entity – attributes .....	36
Figure 3: Relations between entities.....	36
Figure 4: Supertypes and subtypes .....	37
Figure 5: Inverse rules .....	37
Figure 6: EXPRESS-G schema on several pages – referencing .....	38
Figure 7: EXPRESS-G schema – STEP physical file.....	39
Figure 8: IFC building model history .....	49
Figure 9: Number of entities in different IFC releases .....	49
Figure 10: IFC model extension .....	52
Figure 11: Levels of interoperability inside IFC model .....	54
Figure 12: Architectural model – Structural model: relationships.....	90
Figure 13: Building design – submodels relations in IFC based BIM.....	91
Figure 14: Usefull minimum .....	94
Figure 15: Unified finite element model .....	98
Figure 16: Flow chart .....	99
Figure 17: Submodels in structural analysis of IFC accordant BIMs.....	100
Figure 18: Connection model – coupling objects, difference objects.....	101
Figure 19: Objects $W_i$ , $W_j$ , intersection face $f_{i,j}$ with normal vector $\mathbf{n}_{i,j}$ .....	102
Figure 20: A-RM view: IfcProject.....	109
Figure 21: A-RM view: IfcSite.....	110
Figure 22: A-RM view: IfcBuilding.....	110
Figure 23: A-RM view: IfcBuildingStorey.....	111
Figure 24: A-RM view: IfcBeam.....	111
Figure 25: A-RM view: IfcColumn .....	112
Figure 26: A-RM view: IfcSlab.....	113
Figure 27: A-RM view: IfcWall .....	114



Figure 28: Mechanical connectivity.....	119
Figure 29: Connectivity algorithm.....	120
Figure 30: Imposed loads in XML form .....	126
Figure 31: Use-case diagram .....	127
Figure 32: Flowchart of suggested roundtrip engineering .....	129
Figure 33: IFC model server architecture .....	130
Figure 34: Implemented prototype.....	135
Figure 35: Architectural design (Graphisoft Archicad 10) .....	136
Figure 36: IFC model server – uploading architectural design.....	137
Figure 37: Architectural model with building elements (IfcBeam, IfcColumn, IfcSlab) and structural model (IfcStructuralCurveMemeber, IfcStructuralSurfaceMember).....	138
Figure 38: AxisVM IFC interface for BIM import.....	139
Figure 39: Importing architectural model into AxisVM .....	140
Figure 40: Importing structural model into AxisVM.....	141
Figure 41: Communication revolutions and construction.....	171
Figure 42: Direct and centralized information exchange.....	175
Figure 43: Geometric modelling – historical overview .....	178
Figure 44: ANSI-SPARC three layer architecture.....	182
Figure 45: Information models in AEC-FM industry .....	189
Figure 46: VEGA model – communications.....	217

## LIST OF ADOPTED FIGURES

Adopted figure 1: Different viewpoints point out different aspects of the model, but none of them completely describes the whole model (Björk, 1995).....	20
Adopted figure 2: Geometric model / Information model, hypothetical attributes and relations between entities (Khemlani, 2004).....	22
Adopted figure 3: STEP basic elements (ISO 10303, 1992).....	29
Adopted figure 4: List of EXPRESS-G symbols (ISO 10303, 1992).....	35
Adopted figure 5: Schema overview of IFC2x3 (IAI, 2008).....	60
Adopted figure 6: Entities in IFC2x3 (second level of details) (IAI, 2008).....	67
Adopted figure 7: Concept of shape representation (IAI, 2008).....	71
Adopted figure 8: Geometric representation entities (IAI, 2008).....	72
Adopted figure 9: Surface model shape representations (IAI, 2008).....	73
Adopted figure 10: Solid model representations (IAI, 2008).....	75
Adopted figure 11: Hierarchy of building elements (IAI, 2008).....	77
Adopted figure 12: Examples of standard (top) and specific walls (bottom) (IAI, 2008).....	78
Adopted figure 13: Typical scenarios inside structural engineering domain (Weise et al., 2003).....	82
Adopted figure 14: Interoperability with other domains (Weise et al., 2003).....	83
Adopted figure 15: EXPRESS-G schema of ST4 extension (IAI, 2008).....	85
Adopted figure 16: Building elements: Structural elements (Weise et al., 2003).....	86
Adopted figure 17: IGES text file – description of two points, lines and arches (IGES5, 2007).....	177
Adopted figure 18: BSM taxonomy (Liebich, Wix, 2002).....	190
Adopted figure 19: GARM PDU units and subtypes (Ghang, 2004).....	192
Adopted figure 20: GARM – hamburger diagram (Ekholm, 1994).....	192
Adopted figure 21: Aspect models in COMBINE project (Liebich, Wix, 2002).....	194
Adopted figure 22: Layers in integrated multilayer systems (Augenbroe, 1995).....	197
Adopted figure 23: CIS2 and submodels (Eastman, 2000).....	198
Adopted figure 24: CIS2 analysis model (EXPRESS-G) (Eastman, 2000).....	199
Adopted figure 25: CIS2 design model (EXPRESS-G) (Eastman, 2000).....	200
Adopted figure 26: CIS2 fabrication model (EXPRESS-G) (Eastman, 2000).....	201

Adopted figure 27: CIS2 – mapping between submodels (EXPRESS-G) (Eastman, 2000).....	202
Adopted figure 28: RATAS framework (EXPRESS-G) (Björk, 1995).....	204
Adopted figure 29: Type object in RATAS information model (Björk, 1995).....	205
Adopted figure 30: Type object in ATLAS information model (Liebich, Wix, 2002).....	207
Adopted figure 31: ATLAS model architecture – demonstrator (Tolman, 1999).....	208
Adopted figure 32: COMBI framework (Katranuschkov, Scherer, 1996).....	210
Adopted figure 33: Application protocol 236 global architecture (Liebich, Wix, 2002).....	212
Adopted figure 34: BCCM model architecture – proposal and implementation (Liebich, Wix, 2002).....	213
Adopted figure 35: BCCM framework (Froese, 1996).....	214
Adopted figure 36: Basic elements of VEGA framework (Debras et al, 1997).....	215

# 1 UVOD

## 1.1 Opis ožjega področja raziskave

Načrtovanje, gradnja, vzdrževanje in odstranitev sodobnih gradbenih objektov temeljijo na sodelovanju številnih strokovnjakov različnih strok. Njihovo število in področje dela pogojuje predvsem vrsta obravnavanega objekta oz. zgradbe, to je vsakega izdelka industrije, ki oblikuje grajeno okolje (v tuji literaturi predstavljeno industrijo označuje akronim AEC/FM – Architecture, Engineering, Construction, Facilities Management), sodelovanje pa poteka na osnovi medsebojne komunikacije oz. izmenjave informacij o zgradbi. Nabor relevantnih karakteristik zgradbe označuje termin model zgradbe. Njegova interpretacija ni nujno vedno objektivna, interesi množice akterjev, vpletenih v življenjski cikel zgradbe in načini opisa le-te, pa se toliko razlikujejo, da je obstoj vsesplošnega ter vsenamenskega modela za opis kompleksne gradbene domene malo verjeten. Delno k temu prispeva tudi vezanost posameznih akterjev na specifične faze življenjskega cikla zgradbe oz. na njihove (različne) vloge.

Modeliranje produktov v njihovem življenjskem ciklu je zahteven proces. Pri tem je potrebno razlikovati med produktnim in procesnim modeliranjem. Prvo obravnava predvsem fizikalne lastnosti produktov (v najširšem pomenu besede), drugo pa opisuje zaporedje dogodkov oz. procesov, ki se izvršijo v življenjskem ciklu produkta. Čeprav je za učinkovito obvladovanje produktov potrebno upoštevati spoznanja obeh vej modeliranja, bo zaradi specifičnosti obravnavane teme v tej nalogi podrobneje obravnavano le produktno modeliranje. O slednjem lahko govorimo vse od renesanse dalje, ko so pričeli zamišljene zgradbe pred izdelavo konsistentno in celovito risati ter zbirati v projektno dokumentacijo in ko se je vloga »stavbarja« – to je osebe, ki je načrtovala in vodila gradnjo – vse bolj izgubljala. Z digitalizacijo modela oz. procesa modeliranja in uporabo sodobnih informacijskih in komunikacijskih tehnologij (ICT – Information and Communication Technologies) je proces načrtovanja postal še bolj fragmentiran. Programska oprema iz začetnih faz digitalnega načrtovanja, ki je temeljila na dvodimenzionalni paradigmi, ni bila več kos zahtevam sodobnega modeliranja. Delna interoperabilnost med modeli je sicer mogoča tudi z uporabo

dvodimenzionalnih gradnikov (običajno črt), vendar pa iz takšnega zapisa težko razberemo njihov pravi pomen. Ker elementi niso natančno specificirani, so onemogočene nadaljnje analize in avtomatizacije. Delna interoperabilnost zato v praksi velikokrat povzroča podvajanja, napake, zamude, slabo kvaliteto, nizko produktivnost ter posledično dodatne stroške (Graphisoft, 2004). Skladno z razvojem programske opreme za načrtovanje produktov tudi v gradbenem sektorju dvodimenzionalno načrtovalno paradigmo izpodriva tridimenzionalno načrtovanje. Nedvoumna specifikacija elementov in tvorba enotnega informacijskega modela zgradb je namreč mogoča le z uporabo tridimenzionalnih gradnikov. Slednji predstavljajo osnovo objektno orientiranih modelov, ki jih označuje termin informacijski model zgradb (BIM – Building Information Model). Le-ta združuje aspekte posameznih strok, ki so trenutno v praksi običajno zapisani v obliki dokumentov, preglednic, terminskih planov ter nepovezanih in neskladnih datotečnih zapisov. S tem se zvišuje stopnja integracije, koordinacije, sinhronizacije, konsistence ter natančnosti v obravnavani panogi.

Semantika oz. skupno dojemanje vseh bistvenih konceptov panoge predstavlja eno izmed poglobitnih karakteristik informacijskih modelov zgradb. Akterji v življenjskem ciklu zgradbe namreč uporabljajo enake termine (npr. steber, stena, vodnik ipd.), njihov dejanski pomen pa se največkrat razlikuje: izbrani primitiv lahko včasih predstavlja nepomemben atribut, v določenih primerih pa središče zanimanja. Zato je potrebno vsakemu poimenovanju otipljivih in neotipljivih gradnikov pripisati natančen pomen oz. določiti njihovo semantiko. Drugi bistveni element interoperabilnosti predstavlja sintaksa, ki specificira način zapisa (to je zaporedje besed in znakov) semantičnih primitivov modela.

Informacijski model zgradbe po eni izmed popolnejših definicij označuje večdimenzionalno, semantično bogato predstavitev produkta z namenom podpore medsebojni komunikaciji med udeleženci, sodelovanju, simulaciji ter optimizaciji. Informacijske modele zgradb načeloma lahko razdelimo v dve skupini – v prvo skupino sodijo modeli, razviti iz monolitnih aplikacij, ki imajo avtorsko zaščiten format zapisa, predvsem zaradi zagotavljanja prevladujočega položaja na trgu. Lastni zaprti informacijski modeli praviloma izhajajo iz orodij za načrtovanje arhitekture, interoperabilnost pa je pogojena s kapitalskimi povezavami med avtorji programske opreme. V drugo skupino uvrščamo modele, razvite v raziskovalnih ustanovah oz. inštitutih ter neprofitnih organizacijah. Specifikacija teh modelov je v nasprotju z lastniškimi informacijskimi modeli javno objavljena in dostopna. Ne glede na nevšečnosti,

ki se pojavljajo pri drugi skupini modelov (nepopolnost, nezadostna robustnost, ipd.) (Bazjanac, 2002) pa gre pričakovati, da se bodo po zgledu specifikacij spletnih tehnologij v prihodnosti vse bolj uveljavile odprte specifikacije oz. standardi.

V nadaljevanju bo predstavljenih kar nekaj odprtih specifikacij produktnih modelov, specifičnih za gradbeno stroko. Veliko število iniciativ oz. modelov predstavlja posledico velikega števila vpletenih strok v življenjski cikel zgradbe in hkrati nezmožnost doseganja konsenza z definiranjem in implementacijo enotnega informacijskega modela zgradb. Predstavljena naloga se sicer omejuje na trenutno najbolj popularno in uveljavljeno specifikacijo IFC (Industry Foundation Classes). Osnovo temeljnih razredov za industrijo predstavlja mednarodno uveljavljen standard za splošen opis produktov ISO 10303 – STEP (STandard for the Exchange of Product model data), dodatno pa so definirani AEC-FM specifični gradniki. V javno objavljeni prostodostopni specifikaciji (IAI, 2008) je z definicijo entitet ter atributov natančno opredeljena semantika in sintaksa IFC modela. Specifikacija IFC je v razvojni fazi in (še) ne omogoča popolnega opisa poljubnega produkta industrije, ki oblikuje grajeno okolje. Nekatere najbolj pesimistične ocene (Behrman, 2002) karakterizirajo IFC specifikacijo kot neustrezno, in sicer zaradi neprimerne pristopa (uradna definicija specifikacije pred kakršno koli implementacijo) ter počasnega razvoja.

Čeprav je dinamična izmenjava informacij med aplikacijami in platformami na podlagi specifikacije IFC doživela že relativno veliko implementacij v uveljavljeni programski opremi pa se v gradbeništvu za izmenjavo informacij v veliki večini še vedno uporabljajo lastniški formati zapisa. Implementacije se ne glede na naravo aplikacij v veliki večini nanašajo na sicer domensko prilagojeno geometrijsko modeliranje, vse ostale panoge so relativno skromno podprte. Kljub opisani povezavi s standardom STEP in zrelosti specifikacije na predstavljenem področju pa tudi novejša raziskave kažejo, da nekaterim – tudi certificiranim vmesnikom – ni mogoče slepo zaupati ((Geiger, 2001), (Backas, 2002), (Dayal, 2004), (Amor, Ma, 2006), (Pazlar, Turk, 2006)). Takšno stanje sicer ni najbolj zadovoljivo, vendar pa je glede na opisane značilnosti AEC-FM panoge pričakovano. Ne glede na trenutno stanje predstavljajo informacijski modeli zgradb najprimernejši pristop za opis gradbenih produktov v njihovem življenjskem ciklu.

## 1.2 Motivacija raziskovanja in opis problema

Ne glede na jasno predstavo oz. definicijo produktnega modeliranja v gradbeništvu (standardni način opisa poljubnega produkta skozi njegov življenjski cikel) razvoj modelov in njihova implementacija v praksi ni trivialna. To dokazuje veliko število modelov oz. njihovih različic ter tudi številne raziskave implementacij sodobnih informacijskih in komunikacijskih tehnologij v gradbeni praksi ((Samuelson, 2002), (Pazlar, Turk, 2004), (Goh, 2005), (Samuelson, 2008)). Kot najbolj verjetne vzroke za opisano stanje analizirane raziskave navajajo razdrobljenost panoge, neustreznost ponujenih rešitev ter nepoznavanje sodobnih IT rešitev oz. navezanost na uveljavljene pristope pri načrtovanju. Opisano stanje predstavlja motivacijo za izdelavo temeljitega pregleda na področju gradbeništva trenutno najbolj uveljavljene IFC specifikacije: njenih predhodnikov, zgodovine, različic, strukture vsebine in implementacij.

IFC specifikacija ponuja solidno osnovo za raziskave na širšem področju modeliranja gradbenih produktov. Po zgledu ostalih industrijskih panog (npr. avtomobilska, letalska industrija, ipd.) se skuša posamezne faze v življenjskem ciklu digitalizirati, prehode med pod modeli znotraj posameznih specifikacij pa čim bolj poenostaviti. Obstoječi pod modeli znotraj IFC specifikacije (npr. arhitekturni model, računski model) je tesno povezan z individualno predstavo akterjev o modelu. Tako je npr. v Ronneblad (2003) predstavljen eden izmed prvih praktično implementiranih primerov izmenjave podatkov med arhitektom in načrtovalcem prefabriciranih armiranobetonskih (AB) konstrukcij z uporabo strežnika modela zgradb. Geometrijski model konstrukcije je možno prikazati kot predlogo pri načrtovanju prefabriciranih AB elementov v programskem paketu IMPACT, sicer namenjenem izdelavi projektne dokumentacije (delavniških risb, popisa del, ipd.). Novo ustvarjeni gradniki se poleg obstoječega arhitekturnega modela shranijo na strežnik, vendar pa relacije med obema predstavitevama niso shranjene.

S pod modeli imamo opravka v vseh fazah načrtovanja. Najbolj nazoren primer prikazuje zaokroženo načrtovanje zgradb (»roundtrip engineering«): gradbenik konstruktor mora za predstavljeno arhitekturno zasnovo izbrati ustrezno nosilno konstrukcijo ter določiti dimenzije nosilnih elementov. Pri tem ustvari svoj pod model, ki ga tuja literatura označuje s terminom »structural model«. Trenutno najprimernejši splošni prevod predstavlja termin računski

model, čeprav – kot bo prikazano v nadaljevanju – IFC specifikacija razlikuje med računskim modelom in modelom za analizo.

Geometrija konstrukcije (znotraj IFC specifikacije) predstavlja osnovo tako arhitekturnega kot tudi računskega modela, vendar pa med njima lahko nastopajo razlike. Navedena razlika je vzrok, da so v arhitekturnem modelu za opis istih primitivov zgradbe (npr. stene) namenjene povsem druge entitete kot za opis računskega modela. Dodatno se modela lahko razlikujeta tudi v dimenziji prostora: v regularni konstrukciji lahko gradbenik konstruktor za preliminarne izračune namesto tridimenzionalnega uporabi kar ravninski model. Ne glede na izbrane dimenzije in izbrano stopnjo natančnosti se pri iterativnem procesu načrtovanja transformacije med arhitekturnim modelom ter računskimi modeli (teh je v splošnem lahko poljubno število) še vedno opravljajo ročno.

Poenostavljen primer opisanega prirejanja predstavljajo trenutno implementirani IFC vmesniki aplikacij za računsko analizo konstrukcij. Vmesniki so – podobno kot pri dvodimenzionalni načrtovalni paradigmi (DXF, DWG) – trenutno sposobni operirati le z geometrijo produkta, to je z entitetami arhitekturnega modela. Takšno upravljanje z modelom (uvoz in izvoz v aplikacijo) je sicer dobrodošlo, vendar pa ne izkorišča vseh potencialov informacijskih modelov zgradb.

Semantika gradnikov arhitekturnega modela v splošnem ni zadostna, da bi bilo z njimi mogoče opisati vse informacije, potrebne pri generaciji računskega modela. Zato so z razširitvijo IFC specifikacije uvedene nove entitete za opis primitivov računskega modela. IFC specifikacija natančno določa entitete za opis navedenih podmodelov, ne specificira pa možnih preslikav med entitetami navedenih modelov. Sporočilnost arhitekturnega modela v smislu pridobitve dodatnih informacij, potrebnih pri generaciji računskega modela, je v splošnem večja (npr. iz namembnosti prostora lahko določimo obtežbe, iz lokacije objekta pa tudi ostale vplive na konstrukcijo). Navedeni aspekti informacijskega modeliranja v relevantni literaturi še niso bili obdelani.

Številna poročila o rabi modela zgradb IFC v praksi ((Fischer, Cam , 2002), (Bazjanac, 2002)) izpostavljajo najočitnejše pomanjkljivosti obravnavanega pristopa: (1.) velikost fizične STEP datoteke, v kateri je zapisan celoten informacijski model zgradbe, (2.) nekompatibilnost podatkov – pričakovanja aplikacij se ne ujemajo, saj vmesniki lahko določene elemente



predstavijo na različne načine, (3.) omejitve vgrajenega vmesnika – IFC vmesnik je le delno implementiran, (4.) pomanjkljiv model zgradbe – aplikacija pričakuje več informacij, kot jih je v določeni fazi zapisanih v modelu. Oba avtorja priporočata dve rešitvi, ki lahko bistveno pripomoreta k uspešnejšemu obvladovanju modela zgradb IFC. Prvo rešitev predstavljajo delna izmenjava informacij o produktu, drugo pa strežnik modela zgradb IFC. Slednji hkrati rešuje tudi problem sledenja razvoju modela (ali vsi uporabniki uporabljajo najnovejšo različico informacijskega modela zgradbe) oz. izvornih podatkov (kdo je prispeval/modificiral posamezne entitete) (Haymaker et al., 2003). Navedene raziskave predstavljajo motivacijo za tehten premislek o vključitvi strežnika modela zgradb v načrtovani sistem prirejanja med pod modeli v informacijskem modelu zgradb.

Nekoliko nenavadna motivacija za raziskovanje produktne modeliranja v gradbeništvu pa izhaja iz na interoperabilnost vezanih finančnih podatkov ameriškega inštituta za standardizacijo in tehnologije (NIST – National Institute of Standardization and Technology) (Gallaher et al., 2003). Ne zadostna interoperabilnost v življenjskem ciklu zgradb (zajete so nove in obstoječe komercialne, institucionalne ter industrijske zgradbe) izkazuje dodatne stroške tako pri gradnji (6,18 \$/m<sup>2</sup>) kot pri uporabi oz. vzdrževanju (0,23 \$/m<sup>2</sup>). Letni stroški nezadostne interopreabilnosti v ZDA so bili po predstavljeni metodologiji v letu 2002 ocenjeni na 15,8 milijard (10<sup>9</sup>) ameriških dolarjev, pri čemer kar dve tretjini stroškov neposredno bremeni lastnike oz. upravljavce objektov.

### **1.3 Temeljna hipoteza**

Osrednje raziskovalno področje naloge predstavlja IFC specifikacija oz. preslikave med pod modeli znotraj omenjene specifikacije. V klasičnem delotoku načrtovanja izdelkov gradbene industrije je naloga gradbenika konstruktorja analiza arhitekture zgradbe, določitev nosilnega sistema za prevzem obtežb na konstrukcijo in dimenzioniranje nosilnih elementov ter s tem zagotovitev varnosti konstrukcije. Analiza arhitekture zgradbe predstavlja osnovo za določitev računskega modela konstrukcije z zajetimi mehanskimi lastnostmi. Ker sta oba pod modela del informacijskega modela zgradb je smiselno in utemeljeno preučiti relacije in možna prirejanja, s katerimi bi bilo navedene preslikave možno poenostaviti.

Hipotezo naloge predstavlja naslednja trditev: Predstavljeni problem »ročnih« preslikav arhitekturnega modela v računski model, ki jih opravlja gradbenik konstruktor, je moč rešiti oz. delno avtomatizirati z računalniškim kodiranjem hevrističnih pravil ob naslonitvi na semantično naravo izbranega informacijskega modela zgradb IFC.

Že v predlogu disertacije so bile določene naloge, ki bodo sistematično vodile k rešitvi problema ter k oceni pravilnosti postavljene hipoteze:

- Preučiti (na podlagi specifikacije standarda ter dela v praksi) entitete geometrijskega in računskih modelov znotraj standarda IFC ter določiti relacije med njimi.
- Preučiti možne algoritme preslikav med pod modeli IFC specifikacije ter izbrati najprimernejši zapis in/ali algoritem.
- Vključiti standarde s področja projektiranja konstrukcij pri generaciji računskih modelov.
- Oblikovati arhitekturo programskega sistema za reševanje naloge.
- Glede na predlagano arhitekturo izdelati prototip polavtomatskih preslikav med modeli in na podlagi testiranj prototipa preveriti podano hipotezo.

## **1.4 Struktura naloge**

Naloga je razdeljena na pet poglavij:

1. Prvo poglavje je namenjeno uvodni predstavitvi, opisu raziskave, hipotezam ter strukturi naloge.
2. V drugem poglavju je podrobneje predstavljena terminologija informacijskega modeliranja zgradb, prvi poskusi načrtovanja z uporabo računalnikov ter prehod od dvodimenzionalne načrtovalne paradigme k objektno orientiranem modelom.

Najnovejši dosežki informacijskega modeliranja zgradb temeljijo na splošnih spoznanjih in tehnologijah splošnega informacijskega modeliranja. Predstavitev standarda STEP (ISO 10303), njegove arhitekture, grafičnih in tekstovnih metod opisa, generičnih virov in aplikacijskih protokolov je zato samoumevna.

Trenutno najbolj aktualni informacijski model na širšem področju gradbeništva je predstavljen v zadnjem razdelku drugega poglavja. Splošni predstaviti IFC specifikacije sledi pregled zgradbe modela. Posebej podrobno sta predstavljena zapis geometrije modela ter razširitev specifikacije na področje statične analize konstrukcij.

3. V tretjem poglavju so predstavljena tista področja informacijskega modela zgradb IFC, ki predstavljajo predmet zanimanja v zaokroženem načrtovanju konstrukcij. Na podlagi rezultatov sorodnih raziskav in postavljene hipoteze so sintetizirane raziskovalne zahteve, ki predstavljajo osnovo predlagane rešitve zastavljenega problema. Predlagan sistem prirejanja temelji na predpostavki, da je v informacijskih modelih zgradb možno izmenjevati več kot le informacije o geometriji produkta.
4. Četrto poglavje opisuje implementacijo predlaganega sistema. Dodatno so predstavljene potrebne tehnologije in orodja, ki jih vključuje prototipna implementacija predlagane rešitve.
5. V zaključnem delu so povzeti rezultati raziskave ter predstavljeni izzivi za nadaljnje raziskovalno delo.

## **2 PREGLED KLJUČNIH PODROČIJ POVEZANIH Z RAZISKAVO**

Miselni prehod od dvodimenzionalne načrtovalske paradigme k objektno orientiranemu informacijskemu modeliranju ni trivialen. Interdisciplinarnost, ki se zahteva v procesih načrtovanja, gradnje, vzdrževanja in odstranitve zgradb, zahteva sodelovanje številnih strokovnjakov različnih profilov. Predstavljena interdisciplinarnost oz. preplet številnih tehnologij zato zahteva temeljito analizo stanja, ki predstavlja osnovo za iskanje najustreznejše rešitve raziskovanega problema. Zato je v sledečih podpoglavjih izdelana inventarizacija ključnih področij, povezanih z raziskavo:

- Splošna predstavitev informacijskega modeliranja konstrukcij (ključni termini, konceptualno modeliranje, razvoj oz. kronološki pregled modeliranja).
- Standard ISO 10303 – STEP (iniciativa, arhitektura, gradniki, viri, aplikacijski protokoli ter metode implementacije).
- Informacijski model zgradb IFC (splošna predstavitev, obseg, zgradba, ključne strukture, razširitev na področje statične analize konstrukcij).

### **2.1 Informacijsko modeliranje konstrukcij**

V podpoglavju informacijsko modeliranje konstrukcij so podrobneje predstavljene osnove informacijskega modeliranja konstrukcij s poudarkom na pravilni rabi terminologije. V sklepnem delu je podan tudi natančen zgodovinskorazvojni pregled razvoja modeliranja gradbenih konstrukcij.

#### **2.1.1 Terminologija**

Ustrezna terminologija je bistvenega pomena za nedvoumno izmenjavo informacij.

Terminologiji je potrebno posvetiti posebno pozornost zlasti pri obravnavanju interdisciplinarnih tematik, saj se pomen uporabljenih izrazov znotraj obravnavanih panog

lahko razlikuje. Slednjo trditev lahko podkrepimo z rabo termina objekt: v fiziki ta termin označuje skupek mas, v gradbeništvu »s tlemi povezano stavbo ali gradbeni inženirski objekt, narejen iz gradbenih proizvodov in naravnih materialov, skupaj z vgrajenimi inštalacijami in tehnološkimi napravami« (ZGO, 2002) ter v računalništvu konkretno predstavitev razreda v računalniškem programu. Posledično se je na področju informacijskega modeliranja v gradbeništvu (v najširšem pomenu besede) uveljavil termin »zgradba« namesto »objekt«. Povsod, kjer takšnih primerno obrazloženih nadomestitev ni moč uporabiti, mora biti iz konteksta besedila razviden pomen uporabljenega termina.

Žal pri pregledovanju in prevajanju tuje literature pogosto ni moč najti ustreznih slovenskih terminov, ki bi brez dodatnega pojasnjevanja imeli enako ekspresivnost kot v izvornem (angleškem) jeziku. Slednje je najbolj očitno pri kraticah, ki se praviloma ne prevajajo oziroma je namesto njih uporabljen čim bolj ekvivalenten slovenski termin. Tako npr. namesto že predstavljene kratice AEC-FM pogostokrat uporabljamo kar termin gradbeništvo, pri čemer imamo vedno v mislih gradbeništvo v najširšem pomenu besede, to je celotno industrijo, ki oblikuje grajeno okolje.

Termini in kratice so zbrani v dodatku A.

## **2.1.2 Informacijsko modeliranje**

### **2.1.2.1 Informacijski modeli**

S terminom model označujemo abstrakcijo oz. poenostavitev realnosti oz. izvirnika, ki ga v splošnem opišemo z družino specifičnih lingvističnih ali semantičnih terminov. Osnovo modeliranja predstavlja razlikovanje med fenomenom realnosti – izvirnika ter koncepti, uporabljenimi pri opisu le-tega. V tuji literaturi izvirnik običajno označuje termin »Universe of Discourse – UoD«, ki označuje vsa možna stanja objektov realnega sveta z vključeno časovno (razvojno) komponento (Turk, 1992). Primer UoD ki ga navaja citirani avtor: podatkovna baza študentskega referata, v kateri se hranijo vsi podatki o študentih, njihovih (opravljenih) obveznostih ipd.

Turk (1999) modele klasificira kot fizične, socialne in antropološko-tehnične, hkrati pa poudarja, da njihov obstoj ni nujen za inteligentno delo akterjev v specifični stroki.

Na prvi pogled so najbolj relevantni fizični modeli, s katerimi predstavljamo fenomene fizičnega sveta (npr. atom, kolo ipd.). Fizične modele lahko praviloma potrdimo z eksperimenti, njihova verodostojnost pa je odvisna od izbrane natančnosti. Z modelom se skuša približati izvirniku, kar pa na stanje v fizičnem svetu nima vpliva (npr. upogib izbranega nosilca pri določeni obtežbi je v naravi vedno enak ne glede na izbrani model).

Pri socialnih modelih je stanje ravno obratno, saj je možno preko modela vplivati na izvirnik. Tako npr. prebivalci naselja z neurejeno oskrbo s pitno vodo lahko aktivno pripomorejo k izboljšanju stanja, če se neurejenih razmer zavedajo.

Antropološko-tehnični modeli predstavljajo najbolj ranljivo in nestabilno skupino modelov. Prvi del imena poudarja individualistično razumevanje UoD, drugi pa domeno tehnike, ki ji primitivi modela pripadajo in s pomočjo katere so opisani. Verodostojnost antropološko-tehničnih modelov je bolj kot od objektivne predstavitve UoD odvisna od dogovora, kaj pravzaprav v izbranem kontekstu predstavlja realnost.

Rabo modelov v predstavljenem kontekstu običajno označujemo s terminom informacijsko modeliranje. Pri tem je po terminologiji standarda ISO je potrebno ločiti med modeliranjem aktivnosti oz. opisom procesov ter med informacijskim modeliranjem (Cerovšek, 2003). Podobno delitev je mogoče zaslediti tudi v ostalih relevantnih virih (Ghang, 2004): proces označuje zaporedje aktivnosti, ki tvorijo zaključeno celoto, medtem ko produktni model opisuje definicijo, strukturo ter relacije med informacijami v življenjskem ciklu zgradbe.

Eno izmed popolnejših opredelitev predstavljenega termina podaja Forester (2002).

Informacijski model definira kot formalni opis zamisli, dejstev ter procesov, ki skupaj tvorijo model, ki opisuje realnost na podlagi eksplicitno določenih pravil. Ključne faze v navedeni definiciji predstavljajo:

- Formalni opis zamisli, dejstev in procesov predstavlja verodostojen izvleček struktur oz. konceptov, ki omogoča njihovo natančno ter nedvoumno razumevanje.
- Realnost lahko enačimo z že predstavljenim UoD.
- Eksplicitno določena pravila definirajo nedvoumen in natančen pomen oz. semantiko formalnega opisa zamisli dejstev in procesov.

### 2.1.2.2 Konceptualno modeliranje

Raziskave na področju informacijskega modeliranja stremijo k določitvi splošnih konceptov ter relacij med njimi. Poskusi opisovanja vseh podatkov za integrirani CAD sistem na podatkovnem nivoju so se namreč izkazali za neuresničljive (Reed, 1998).

Trenutno najbolj uveljavljen način opisa poljubnega UoD vsebuje naslednje gradnike (Turk, 1992):

- **Entiteta** (oz. objekt) označuje obstoj fizičnih (stebra) ter abstraktnih gradnikov (cena stebra). Entitete lahko razvrstimo v dve skupini: prva obsega simbole, ki opisujejo individualne primerke realnega sveta (Ljubljanski grad), druga pa vključuje simbole, ki predstavljajo razrede oz. tipe objektov (gradovi). V splošnem uvrščamo v razred (abstraktnih) objektov entitete, ki imajo do izbrane stopnje podrobnosti podobne lastnosti.
- **Atributi** v splošnem opisujejo lastnosti entitet. Najenostavneje si je moč attribute predstavljati kot pridevnike entitet (»opečni« zid), čeprav se v splošnem lahko nanašajo na množico objektov, ki imajo določeno lastnost (»opečno« zidovje). Atributi lahko opisujejo specifično lastnost ali pa entiteto karakterizirajo v obliki sestavljenega atributa (dimenzije stene, material ipd.). Ločimo: 1.) eksplicitne attribute (lahko so obvezni s predpisanimi vrednostmi ali opcijski, katerih vrednosti niso nujno predpisane), 2.) inverzne attribute, 3.) izpeljane attribute.
- **Relacija** je splošen pojem za grupiranje elementov. Relacije so lahko binarne ali n-arne, običajno pa je tudi določena smer, v katero relacijo razumemo (je del, ima dele, se stikajo ipd.). Pomembnejše splošno uporabljene relacije so agregacija, generalizacija, abstrakcija, konstrukt, specializacija, verzija ali alternativa, omejitev ter pogled ali aspekt.

Z navedenim opisom, ki je popolnoma neodvisen od implementacije, lahko opišemo poljubne koncepte: fizične objekte, ideje (abstraktne koncepte), procese ipd.

Opisani gradniki in pripadajoče obrazložitve asociirajo na termine oz. koncepte, poznane iz objektno orientiranih jezikov. Primerjava je podana v preglednici 1.

Entitete in pripadajoče lastnosti posameznega UoD lahko karakteriziramo z logičnimi trditvami, ki so lahko resnične ali neresnične. Logične trditve v obliki omejitev in pravil, ki veljajo za vsa stanja UoD in hkrati predstavljajo nabor objektov, ki lahko oz. ki morajo nastopati v opisu izvirnika, označujemo kot obvezne trditve (npr. vsaka hiša ima streho). V drugo skupino (neobvezne trditve) pa uvrščamo tiste, ki so resnične le za specifične objekte v specifičnem času (npr. hiša ima leseno streho, kritino bo v prihodnosti zamenjala pločevina).

#### Preglednica 1: Objektno orientirani programski jeziki : Informacijsko modeliranje

Table 1: Object oriented languages : Information modelling

<b>objektno orientirani programski jeziki</b>	<b>informacijski modeli</b>
razred (»class«)	entiteta (»entity«)
objekt (»object«)	instanca (»instance«)
atribut (»attribute«)	atribut (»attribute«)
asociacija (»association«)	relacija (»relation«)
kardinalnost (»cardinality«)	kardinalnost (»cardinality«)
omejitev (»constraint«)	pravilo (»rule«)
dedovanje (»inheritance«)	dedovanje (»inheritance«)

Če zberemo vse resnične trditve o specifičnem UoD, potem jih glede na njihovo naravo razvrstimo v konceptualni model, ki vsebuje obvezne trditve, ter v informacijsko (ne podatkovno) bazo z neobveznimi trditvami. Informacijsko bazo si lahko predstavljamo kot hipni posnetek UoD v izbranem času, medtem ko konceptualna shema formalizira posplošene zakonitosti. Konceptualna shema lahko vsebuje še dodatne omejitve in pravila: 1.) statična pravila opisujejo odvisnosti med deli modela neodvisno od časovne komponente, 2.) dinamična pravila regulirajo evolucijo sistema skozi čas ter 3.) izpeljana pravila, ki regulirajo medsebojno izpeljavo atributov (Björk, 1995).

Očitno je lahko več informacijskih baz skladnih z eno konceptualno shemo. Precej abstraktni opis navedenih terminov predstavimo na dveh enostavnih primerih: v teoriji podatkovnih baz konceptualno shemo predstavlja deklaracija podatkovnih tipov, informacijsko bazo pa



podatki, ki so v izbranem trenutku zapisani v bazi. Podobno pri modeliranju AEC-FM produktov konceptualno shemo predstavlja specifikacija splošnega gradbenega produkta, informacijsko bazo pa njena implementacija v obliki konkretnega modela zgradbe.

Konceptualnemu modelu ustreza konceptualna shema, ki mora v splošnem izpolnjevati vsaj naslednje kriterije (Björk, Penttila, 1989):

- obsežnost (vključitev podatkov do izbrane stopnje podrobnosti);
- zbirnost oz. kumulativnost (vključitev podatkov iz celotnega življenjskega cikla produkta);
- neredundantnost (izključitev nepotrebnih podatkov);
- neodvisnost izhodnih podatkov (struktura in vsebina izhodnih dokumentov naj bi bila določena neodvisno od informacijske strukture);
- neodvisnost od programske ter strojne opreme (vsebina in specifikacija zapisa vsebine morata biti natančno določeni).

Načrtovanje konceptualnega modela je kompleksen proces, zato je pri tem delu smiselno uporabiti spoznanja ter že razvita ogrodja ter standardizirane načine opisa (npr. ANSI-SPARC arhitekturo, ki je podrobneje opisana v prilogi B – B.6.1).

### **2.1.2.3 Jeziki za informacijsko modeliranje**

Arhitektura konceptualnih shem je lahko nedvoumno definirana le ob izbranem jeziku za njeno predstavitev. V literaturi se za opis konceptualne sheme pojavljajo različno ustrezni termini: jezik za informacijsko modeliranje, jezik za konceptualno modeliranje, jezik za opis podatkovnega modela, jezik za opis minimalne konceptualne sheme ter še nekateri drugi. Najbolj neposrečen oziroma celo zavajajoč termin predstavlja jezik za opis podatkovnega modela. Medtem ko termin informacija poudarja logično modeliranje UoD, raba izraza podatki asociira predvsem na učinkovito shranjevanje ter manipuliranje znotraj podatkovnih baz in datotek (Björk, 1992).

Najprimernejši in zato v nadaljevanju uporabljen termin predstavlja jezik za informacijsko modeliranje. Obstaja kar nekaj takšnih jezikov z nekaterimi skupnimi lastnostmi:

- poleg objektov so definirani tudi razredi objektov;

- možnost grupiranja atributov in integritetnih pravil za posamezne razrede objektov;
- vsebovani so splošni mehanizmi za definiranje kompleksnejših pravil in integritetnih omejitev;
- vsebovani so mehanizmi dedovanja (generalizacija, specializacija);
- atributi entitet so lahko enostavni ali sestavljeni podatkovni tipi.

Björk (1995) specificira najpomembnejše zahteve, ki jih mora izpolnjevati izbrani jezik za informacijsko modeliranje:

- zmožnost modeliranja semantike UoD brez poenostavitev, ki bi izhajale iz jezika za modeliranje;
- zmožnost modeliranja projektantovih namer in ciljev;
- podpora evolucijskemu načrtovanju oz. razširljivost sheme;
- uporabnost pri izmenjavi podatkov med heterogenimi računalniškimi aplikacijami izbranega področja modeliranja;
- tehnična izvedljivost implementacij v trenutno uporabljeno programsko opremo;
- realistične zmožnosti za standardizacijo predlaganega jezika (v smislu konsenza z različnimi standardizacijskimi telesi).

Isti avtor navaja, da standard ISO v poročilu »Koncepti in terminologija konceptualnih shem in podatkovnih baz« razlikuje tri različne pristope, ki posledično pogojujejo uporabo jezika za informacijsko modeliranje:

- entiteta – atribut – povezave (razmerja),
- binarna ter elementarna razmerja,
- pristop predikatne logike.

Pri modeliranju gradbenih produktov se večinoma uporablja prvi pristop. UoD namreč sestavljajo v večini fizične komponente, pri katerih je predstavljena logika modeliranja še najbolj naravna (npr. entiteti – stena, okno, atributi – dimenzije stene in okna, relacija – okno »vsebovano« v steni).

Danes se v informacijskem modeliranju v gradbeništvu največ uporabljajo jeziki EXPRESS, NIAM, IDEF1X ter UML.

#### **2.1.2.4 Stopnja interoperabilnosti**

Bistveni namen konceptualnih modelov je zagotavljanje interoperabilnosti med sistemi. Stopnjo interoperabilnosti je smiselno kvantitativno opredeliti, saj se le-ta med posameznimi sistemi praviloma razlikuje. V domeni tehnike obstaja več klasifikacij konceptualne interoperabilnosti. Tolk (2003) navaja dve takšni razvrstitvi: stopnja interoperabilnosti informacijskih modelov (LISI – Levels of Information System Interoperability) ter NATO NC3TA referenčni model interoperabilnosti (NMI – NATO Reference Model for Interoperability).

LISI model razvršča modele v pet skupin:

- sistemsko specifični podatki (vsak sistem uporablja svoj način zapisa, popolna nekompatibilnost),
- dokumentirani podatki (podatki ter vmesniki so dokumentirani),
- razvrščeni statični podatki (uporaba skupnih referenčnih modelov in ontologij),
- razvrščeni dinamični podatki (skupni sistemski pristop, odprto kodne rešitve),
- harmonizirani podatki (skupni konceptualni model, semantična konsistenca).

Tudi NMI podaja podobno razvrstitev:

- ni izmenjave podatkov (brez kakršnikoli fizičnih povezav),
- nestrukturirana izmenjava podatkov (izmenjava nestrukturiranih človeku berljivih zapisov),
- strukturirana izmenjava (izmenjava strukturiranih človeku berljivih zapisov),
- brezšivno deljenje podatkov (avtomatska izmenjava podatkov znotraj sistema na osnovi skupnega modela),
- brezšivno deljene informacij (univerzalna interpretacija informacij).

Namen izdelave konceptualnih modelov je očitno doseganje harmonizacije oziroma univerzalnosti interpretacije informacij.

Namesto predstavljene natančne razvrstitve oz. opredelitve interoperabilnosti se v literaturi najpogosteje pojavljajo trije termini ki opredeljujejo stopnjo interoperabilnosti:

- Sintaktična inteoperabilnost: Pravilni vrstni red in izbrane kombinacije besede, simbolov, oz. sintaktičnih elementov in njihovih parametrov zagotavljajo inteoperabilnost na nivoju zapisa oz. opisa modela).
- Semantična inteoperabilnost: Poleg sintaktične pravilnosti mora biti zagotovljena tudi semantična pravilnost – smiselnost opisa modela.
- Konceptualna interoperabilnost je sorodna strukturalni, je pa izraz strukturalna načeloma natančnejši, ker koncept kot tak lahko vsebuje tudi semantiko.

#### **2.1.2.5 Produktni podatkovni model in podatkovni model**

Konceptualno shemo določajo formalizirane posplošene zakonitosti UoD. Če je opis namesto splošnega UoD omejen na primitive, ki jih je ustvaril človek in so le-ti zapisani v enem izmed jezikov za informacijsko modeliranje, potem konceptualno shemo imenujemo produktni podatkovni model (Product Data Model). Slednjega lahko označimo tudi kot najbolj splošen koncept v konceptualnem opisu produkta, ki vsebuje informacije o načrtovanju, proizvodnji ter uporabi le-tega. Konceptualni in produktni podatkovni model torej nista enakovredna termina, čeprav nekateri viri med njima ne razlikujejo. Podobno je potrebno razlikovati med produktnim podatkovnim modelom ter produktnim modelom. Termin produktni model predstavlja hipni posnetek izbranih primitivov izbranim času oz. že predstavljeno informacijsko bazo (Björk, 1992).

V literaturi je moč najti različne interpretacije navedenih terminov. V najbolj ohlapnem primeru se med konceptualno shemo, produktnim podatkovnim modelom in produktnim modelom ne razlikuje, vendar pa mora biti iz konteksta razviden pomen termina.

Razliko med termini nazorno prikazuje tudi splošni algoritem izdelave informacijskih modelov (Turk, 1992):

- izbrati ali izdelati ustrezen model, ki bo dal na razpolago osnovne koncepte in orodja za opis podatkov, relacij med podatki, pomena podatkov in omejitev, ki se v zvezi s podatki pojavljajo;
- izbrati primerno tehniko oz. jezik za modeliranje;

- s pomočjo izbrane tehnike izdelati konceptualni oz. produktni podatkovni model;
- implementacija konceptualnega oz. produktnega podatkovnega modela v programsko opremo;
- z modeliranjem konkretnih produktov izdelamo produktne modele (model zgradbe).

Praviloma prve tri korake izdelajo načrtovalci modela, četrtega izdelovalci programske opreme ter zadnji korak končni uporabniki.

### **2.1.2.6 Informacijski model zgradbe**

Implementacijo produktnega podatkovnega modela na področju gradbeništva (v najširšem pomenu besede) imenujemo informacijski model zgradbe (Building Information Model – BIM). Podobno kot konceptualni tudi informacijski model zgradbe ne predstavlja objektivne predstavitve realnosti, temveč individualno interpretacijo realnosti v določenem kontekstu (Turk, 1992a). Poleg informacijskega modela zgradbe (BIM) so v uporabi še nekateri drugi ne tako ekspresivni termini: model zgradbe, virtualni model, objektni model ter objektno orientiran model.

V literaturi lahko najdemo množico precej sorodnih opisov predstavljenega termina:

- Turk (1992) informacijski model zgradbe poimenuje model gradbeniškega produkta. Z njim označuje unijo vseh podatkov, ki nastajajo in se uporabljajo in izginevajo skozi celotno življenjsko dobo produkta. Avtor dodatno izpostavlja nasprotje, ki ga v obrazložitev termina vnaša beseda »vseh«. Modeliranje samo izbranih lastnosti produkta običajno pogojujeta obsežnost ter kompleksnost modela, ki predstavljata vir težav pri informacijskem modeliranju. Dodatno je potrebno izpostaviti še težave pri interpretaciji informacijskega modela zgradbe. Ker je le-ta funkcija konteksta, domene, razčlenitve, kulture, ipd. je splošna primernost kompleksnih modelov vprašljiva (Turk, 1999).
- Björk (1995) informacijski model zgradbe definira kot vrsto konceptualne sheme, kjer UoD predstavljajo zgradbe, in sicer v procesih načrtovanja, gradnje, uporabe in vzdrževanja. Za razliko od tradicionalnega posrednega načina opisa zgradbe se v informacijskem modelu zgradbe uporablja neposredni način opisa fizičnih komponent zgradb. Avtor eksplicitno poudarja razliko med splošnim informacijskim modelom

zgradb ter informacijskim modelom konkretne zgradbe. Slednji predstavlja s konceptualno shemo skladno informacijsko bazo opisovane konkretne zgradbe. Isti avtor (Björk, Penttila, 1989) informacijski model zgradbe označuje tudi kot »zbirko informacij o zgradbi skozi celoten življenjski cikel«.

- Tolman (1999): produktni podatkovni model predstavlja informacijski model, ki implicitno vsebuje podatke o formi, funkciji in obnašanju produkta v njegovi življenjski dobi. Informacijski model zgradbe predstavlja poseben primer produktnega podatkovnega modela, v katerem je predstavljena forma (geometrija in topologija), funkcija (namen zgradbe) in obnašanje (prenos obtežbe, ipd.). Avtor dodatno primerja splošno tehnično dokumentacijo ter informacijske modele zgradbe: sporočilnost slednjih naj bi bila enaka ali večja sporočilnosti vseh načrtov in spremljajoče dokumentacije.
- Cerovšek, Turk (2002): informacijski modeli zgradbe predstavlja vrsto produktnega modela, katerega računalniško berljiv opis zgradb je skladen z izbrano specifikacijo.
- Cerovšek et al. (2002) naslavljajo informacijski model zgradbe kot produkt modeliranja gradbenega projekta – stavbe ali inženirske zgradbe. Pri tem so v produktnih modelih zastopani tudi načini njihove uporabe: v računalniških aplikacijah, za fizični digitalni zapis, za izmenjavo podatkov itd.
- Ghang (2004) karakterizira informacijski model zgradbe posredno preko produktnih podatkovnih modelov. Slednji predstavljajo formalno strukturirano shemo nabora informacij o (gradbenem) produktu, ki so ustvarjene, spremenjene ter izbrisane v njegovem življenjskem ciklu. Produktni podatkovni model se bistveno razlikuje od ostalih podatkovnih modelov: 1.) vsebuje kompleksne informacije o geometriji; 2.) geometrija produkta se navezuje (oz. je izpeljana) iz funkcije produkta; 3.) ker se modelira celoten življenjski cikel, so v modelu lahko vsebovane tudi informacije, potrebne za izdelavo, sestavo, preskus, upravljanje ter odstranitev produkta.

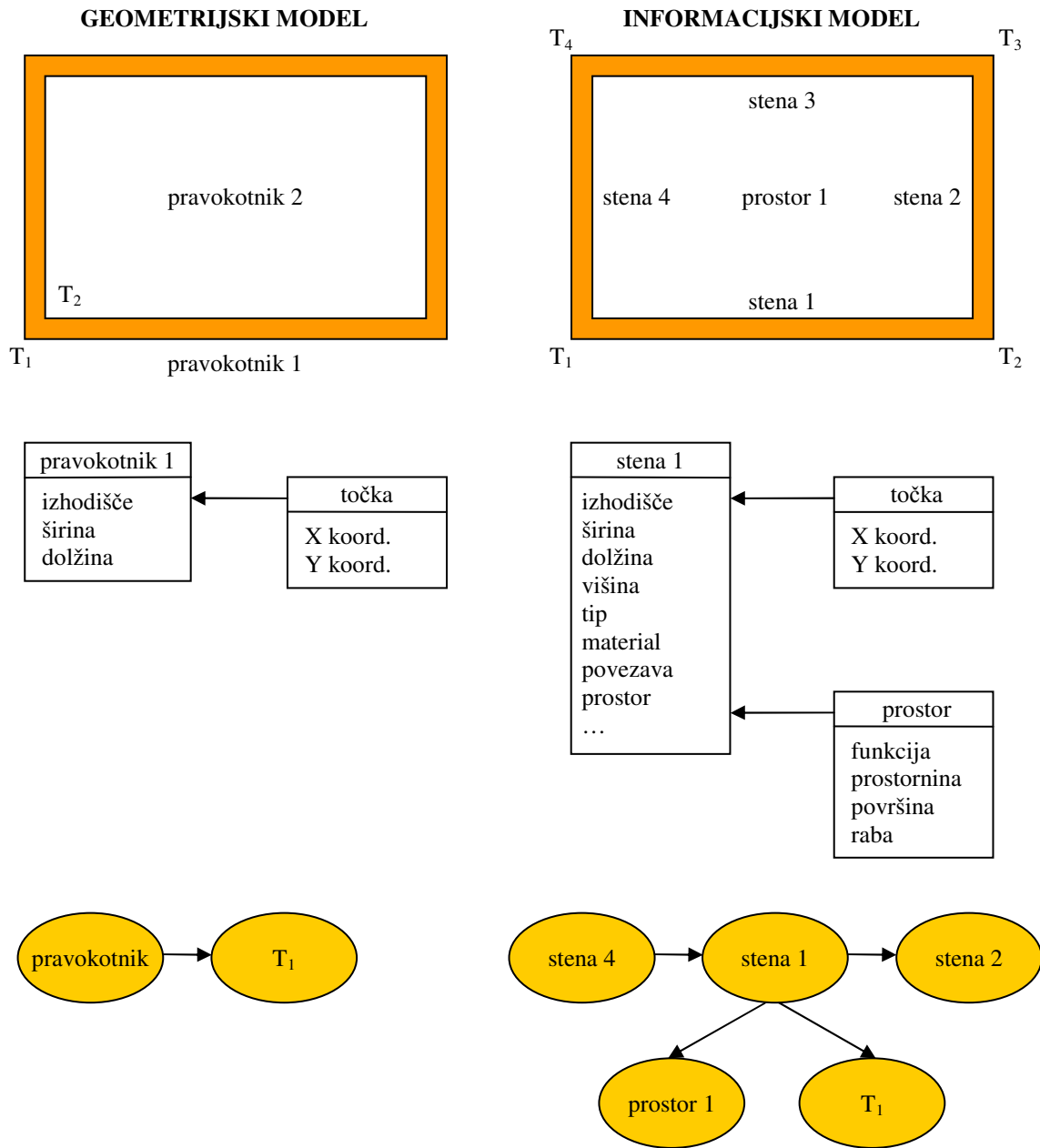
Pojav in razvoj informacijskega modela zgradbe je tesno povezan z rabo računalnikov. V gradbeništvu se namesto predstavitve realnosti favorizira specifično razumevanje obravnavane tematike. Pri tem se tvorijo kompleksni modeli, ki se jih z izpostavljanjem različnih aspektov v nadaljevanju lahko drobi v manjše obvladljive kose.



objektno orientirani modeli. Pri uporabi le-teh je primat geometrije v veliki meri zmanjšan, bistveno pa se poveča obseg informacij o produktu. Slednje je moč (avtomatsko) izluščiti ter ponovno uporabiti (npr. za potrebe dokumentiranja, izdelave predstavitev, vizualizacije itd.). Semantični računalniško berljiv opis zgradbe, ki ga predstavlja informacijski model zgradbe bo v nadaljevanju največkrat označeval zgolj termin model. Izpostaviti je potrebno še nekaj razlik med geometrijskim ter informacijskim modelom zgradbe:

- Informacijski model zgradbe temelji na predstavitvi semantičnih komponent zgradbe. Vendar pa ni nujno, da so primitivi v posameznih pogledih predstavljeni z ekvivalentnimi entitetami (npr. rebričasta plošča je v arhitekturnem modelu geometrijsko predstavljena kot kvader, v računskem modelu pa kot T nosilec). To hkrati pomeni, da se razsežni prostor modelov lahko razlikuje (2D, 3D ali oboje).
- V geometrijsko orientiranih CAD sistemih razlika med geometrijskimi in informacijskimi modeli zgradb ni vedno očitna. Vizualno razliko nakazuje možnost animacij, številnih 3D pogledov (skrivanje elementov, senčenje, fotorealistični pogledi – lahko z upoštevanjem naravne osvetlitve gradnikov) in nenazadnje tudi možnost izdelave poljubnih 2D pogledov – prerezov (tloris, prečni prerez) v poljubnem merilu ter v poljubni z modelom zagotovljeni natančnosti.
- Praviloma je geometrijski opis informacijskega modela zgradb tridimenzionalen. Če temu ni tako, potem določen spekter prednosti informacijskega modela zgradbe ni na voljo (npr. volumetrične informacije za popis del).
- Termin »pogled« pri rabi informacijskega modela zgradbe ni rezerviran zgolj za predstavitev geometrije (»Geometric View«), temveč se uporablja v celotnem življenjskem ciklu zgradbe (npr. pri popisu količin (»Quantity Take-off View«), oceni stroškov, terminskem planu, vzdrževanju ipd.).





Privzeta slika 2: Geometrijski model / Informacijski model, hipotetični atributi ter razmerja med entitetami (Khemlani, 2004)

Adopted figure 2: Geometric model / Information model, hypothetical attributes and relations between entities (Khemlani, 2004)

Razlika med geometrijskim in informacijskim modelom najbolj nazorno prikazuje koncept prostora (privzeta slika 2). Pri uporabi 2D in v večini primerov tudi 3D CAD opreme koncept prostora ne obstaja. Povsem drugače je v informacijskem modelu zgradb, kjer prostor s pripadajočimi relacijami (nadstropje, stene, plošče ipd.) predstavlja enega izmed bistvenih konceptov, integriranih v BIM. Informacije o velikosti in namembnosti prostora so recimo nujno potrebne pri preverjanju zahtev arhitekture (minimalna velikost prostorov, dimenzije prehodov ipd.), pri izračunu energetske učinkovitosti ter še na nekaterih področjih in jih je – vsaj teoretično – iz informacijskega modela zgradb relativno lahko enostavno izluščiti. V primeru rabe geometrijskega modela pa temu ni tako in bi določitev predstavljenih količin zahtevala kompleksne izračune, ki jih ni enostavno (in povsem) avtomatizirati.

V splošnem ima dokument, pridobljen iz modela oz. pripadajoče podatkovne baze, »nižjo stopnjo semantike«, kar onemogoča popolno povratno rekonstrukcijo modela.

## **2.2 STEP**

Standard STEP z uradnim nazivom ISO 10303 predstavlja zbirko standardov za opis in izmenjavo informacij o produktu skozi njegov življenjski cikel. Enega izmed poglavitnih namenov iniciative predstavlja standardizacija izmenjave podatkov med različnimi računalniškimi sistemi npr. med CAD, CAM, CAE, PDM/EDM ter drugimi sistemi CAx, in sicer na vseh področjih inženirskega načrtovanja: v letalstvu, avtomobilski industriji, ladjarstvu, naftni industriji ter tudi v gradbeništvu.

V nadaljevanju so predstavljeni vzroki za nastanek standarda ter njegova arhitektura. Podrobneje bodo predstavljeni deli standarda, ki so privzeti tudi v specifikaciji informacijskega modela zgradb IFC.

### **2.2.1 Potrebe industrije**

#### **2.2.1.1 Identifikacija potreb industrije**

Že v začetku osemdesetih let so na nivoju nacionalnih standardizacijskih teles ugotovili, da z obstoječimi načini izmenjave informacij o geometriji (npr. datotečna izmenjava v formatih DXF, IGES SET VDA-FS – preglednica 2) ne bo mogoče opisati kompleksnih produktov v njihovem celotnem življenjskem ciklu. Izkazana je bila potreba po standardizaciji celotne

informacijske arhitekture produkta, pri kateri bo datotečna izmenjava predstavljala le eno izmed implementacijskih metod izmenjave. Navedeni formati izmenjave namreč predstavljajo specifikacijo za izmenjavo geometrije, ki v sodobnem produktnem modeliranju predstavlja le del informacij o produktu.

Flower (1995) navaja štiri posplošene poglavitne potrebe za standardizacijo opisa produkta ter izmenjave informacij o njem:

- potreba po dolgotrajnem shranjevanju informacij o produktu, in sicer v takšni obliki, ki omogoča kontinuirano rabo modela;
- zmanjšanje razvojnih stroškov in stroškov vzdrževanja vmesnikov med različnimi računalniškimi sistemi;
- neodvisnost od programske opreme, ki informacije ustvarja oz. jih uporablja;
- izmenjava informacij med različnimi okolji, disciplinami (oz. v praksi med oddelki ter med podjetji).

Preglednica 2: Uporaba standardov za izmenjavo informacij

Table 2: The use of information exchange standards

	IGES	SET	VDA-FS	DXF
letalska ind.	••	•		•
avtomobilska ind.	••	•	•	
gradbeništvo				••
gradnja-industrija	•	•		•
črpanje surovin (nafta, plin)				•
ladjedelnštvo	•			•
elektrotehnika/elektronika	•	•		
široka potrošnja	•			•

### Preglednica 3: Karakteristike opisa informacijskega modela

Table 3: The characteristics of information model description

informacije	pred ISO 10303	ISO 10303
razpoložljivost	heterogeni specifični formati	prosta izmenjava podatkov med različnimi sistemi
dostopnost	omejena in pogojena s posameznimi specifičnimi preslikavami	standardizirani vmesniki
ponovna raba	znatne izgube podatkov ter ponovno ustvarjanje podatkov	omogoča ustvarjanje ter primerno vzdrževanje v heterogenih okoljih
kvaliteta	zmanjšana zaradi nenatančnosti, nepopolnosti ter lahko tudi zaradi presežka podatkov	povečana zaradi uporabe standardnih modelov ter vmesnikov

Standard, ki bi izpolnjeval navedene zahteve, bi moral dodatno zagotavljati stabilnost, razširljivost ter (prosto) dostopnost.

V primerjavi z uveljavljenimi opisi produkta bi morala nova iniciativa izboljšati opis predvsem glede razpoložljivosti, dostopnosti, ponovne rabe in kvalitete informacij (preglednica 3).

#### 2.2.1.2 Standardizacijska iniciativa

Paralelne iniciative nacionalnih standardizacijskih teles (npr. initiative PDES, ki ga je razvijal ameriški nacionalni institut za standardizacijo) so se sredi osemdesetih let združile, in sicer v okviru mednarodne organizacije za standardizacijo (ISO – International Organization for Standardization). Prve delovne različice STEP specifikacije so se pojavile leta 1988, uradno veljavni standard pa leta 1994. Dolgotrajno sprejemanje standardov znotraj organizacije ISO predstavlja ustaljeno prakso, in sicer zaradi zapletene procedure (predpisanih je šest faz – predlog (»New Working Item – NWI«), delovni osnutek (»Working Draft – WD«), komisijski delovni osnutek (»Committee Draft – CD«), osnutek mednarodnega standarda (»Draft International Standard – DIS«), končni osnutek mednarodnega standarda (»Final Draft International Standard – FDIS«) ter mednarodni standard (»International Standard – IS«)).

Mednarodna standardizacija je dodatno povezana z zahtevno koordinacijo ter usklajevanjem z nacionalnimi standardizacijskimi telesi, ki svoje interese uveljavljajo preko članstva v delovnih skupinah. Za razvoj standarda STEP je v okviru organizacije ISO zadolžena tehnična komisija 184 z nazivom Sistemi za industrijsko avtomatizacijo in integracijo (Technical Committee 184: Industrial Avtomation System and Integration – SC4: Industrial data).

## **2.2.2 STEP arhitektura**

### **2.2.2.1 Osnovni gradniki**

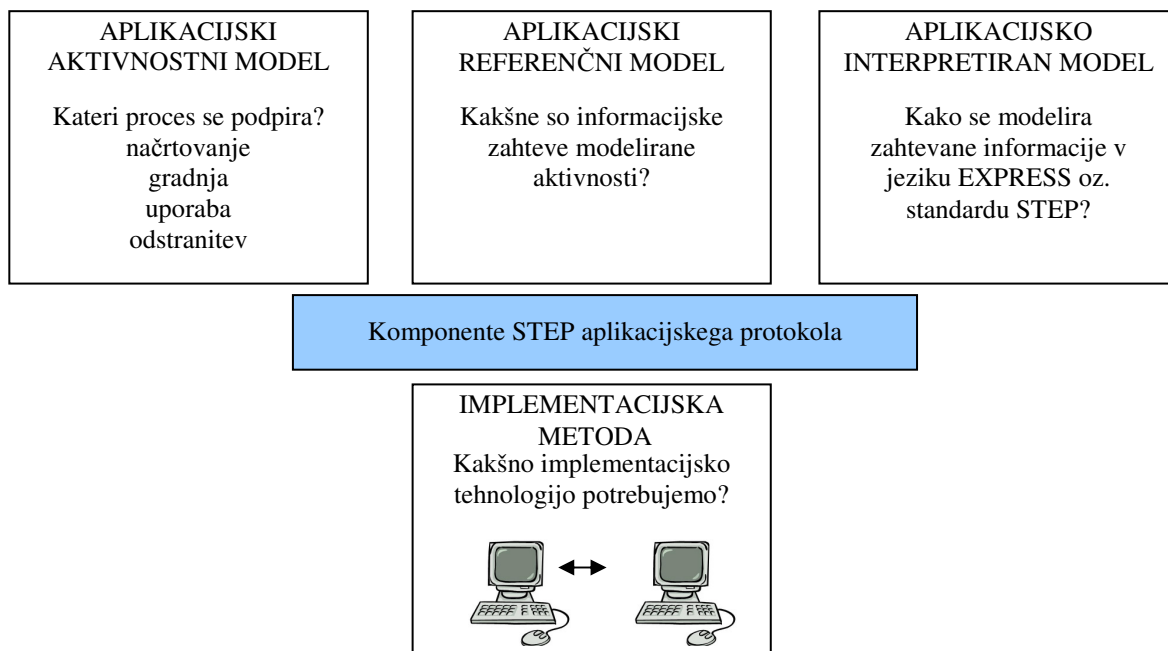
Na podlagi predstavljenih zahtev je bila izoblikovana arhitektura produktnega modela STEP, ki danes predstavlja osnovo sodobnega produktnega modeliranja. Pri oblikovanju arhitekture so bili med drugim upoštevani tudi naslednji principi:

- vključeni so bili koncepti, poznani iz objektno orientiranega programiranja;
- vključene so bile formalne specifikacije definiranih struktur, ki so povsem neodvisne od implementacije (družina jezikov EXPRESS);
- ločena sta bila podatkovni model in fizični zapis (z različnim načinom zapisa v datoteke ali podatkovne baze);
- vključena podpora delnim modelom, s čimer se eliminira implementacija za določeno stroko nepomembnih delov modela.

Ključne elemente STEP arhitekture predstavljajo aplikacijski protokoli (»AP – Application Protocols«), ki opisujejo domensko specifične produkte (slika 1). Osnovne gradnike aplikacijskih protokolov predstavljajo:

- aplikacijski aktivnostni model (»Application Activity Model – AAM«). Z njim se definira obseg posameznega aplikacijskega protokola, in sicer na podlagi funkcijskih zahtev in informacijskih tokov. Identificirajo se vhodno-izhodni podatki, metode ter parametri v modelu uporabljenih funkcij. Ker se v AAM opisuje aktivnosti in ne produktov je potrebno uporabiti primeren standardiziran način opisa (npr. IDEF0);
- aplikacijski referenčni model (»Application Reference Model – ARM«), s katerim se na podlagi AAM modela specificirajo zahteve po informacijah. Model ARM predstavlja referenčni model pri konkretnih implementacijah;

- aplikacijski interpretacijski model (»Application Interpreted Model – AIM«) se uporablja neposredno pri implementaciji produktnih modelov, predstavlja pa opis produktnega modela na podlagi STEP integriranih virov.



Slika 1: Struktura STEP aplikacijskih protokolov

Figure 1: STEP application protocol structure

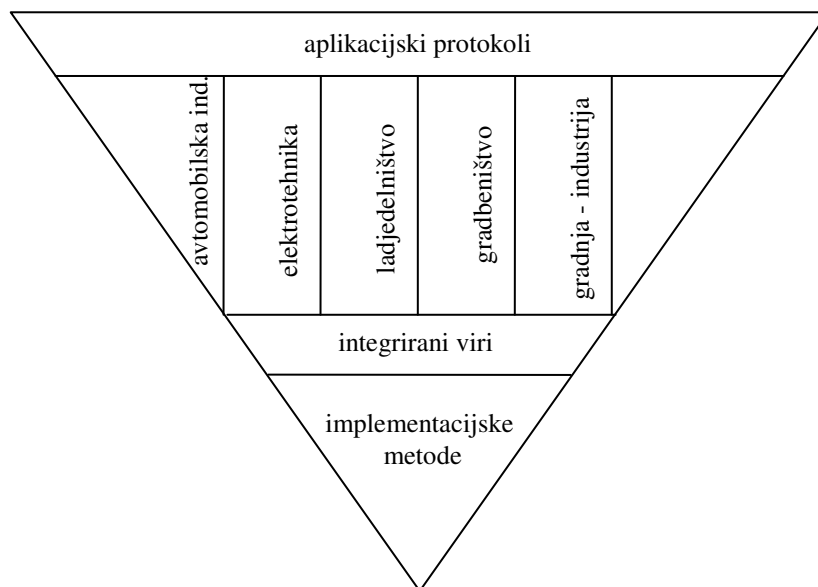
### 2.2.2.2 Vsebinska delitev standarda STEP:

Elementi standarda STEP (privzeta slika 3):

- družina jezikov za informacijsko modeliranje **EXPRESS**;
- v jedru je vključen nabor informacijskih struktur, ki vsebujejo različne poglede na informacijsko modeliranje, ki ga označuje termin **integrirani viri** (»Integrated Resources«). V to skupino sodijo informacije, ki se v opisu produkta pojavljajo večkrat oz. na različnih področjih – npr. opis geometrije produkta ter upravljanje s podatki o produktu (PDM – Product Data Management). Integrirane vire delimo na integrirane generične vire ter na integrirane aplikacijske vire (slednji so vezani na domensko specifične AP);

- domensko specifične modele imenovane **aplikacijske protokole** (Application Protocols«), ki implementirajo integrirane vire, in sicer za predstavitev produktnih modelov v strojništvu, elektrotehniki ter ostalih panogah. Skupek aplikacijskih protokolov označuje termin **razred skladnosti** (»Conformance Class«). Implementacija običajno temelji na razredih skladnosti, saj so aplikacijski protokoli preobsežni za implementacijo v celoti;
- nabor **metod za implementacijo**:
  - datotečna izmenjava podatkov (**STEP fizična datoteka, XML zapis**). Pri datotečni izmenjavi vsaki aplikaciji pripada vmesnik za branje in pisanje določenih STEP aplikacijskih protokolov. Pri tem se lahko število protokolov, ki jih posamezne aplikacije podpirajo, razlikuje, izmenjava pa bo omogočena le za skupne protokole;
  - programski vmesnik (API – Application Program Interface), imenovan **standardni vmesnik za dostop do podatkov** (SDAI – Standard Data Access Interface). Vmesnik je namenjen dostopu do STEP podatkov, zapisanih v repozitoriju. Trenutno so vmesniki definirani za C, C++, Corba/IDL ter Java. SDAI predstavlja bolj dovršen sistem izmenjave informacij, pri katerem STEP strukture predstavljajo jedro pripadajoče podatkovne baze. Aplikacije z vmesnikom SDAI razvrščamo tudi glede na sposobnost operiranja s podatki, shranjenimi v zbirkah podatkov. Če so aplikacije sposobne vsaj delno preverjati pravila, zapisana v EXPRESS shemi, potem takšen nivo implementacije označujemo kot zbirko podatkov. Če pa se upoštevajo vse omejitve in pravila, zapisana v EXPRESS shemi, ter so vključeni tudi mehanizmi sklepanja, potem gre za najvišji nivo implementacije, imenovan zbirke znanja. V praksi se največ implementacij pojavlja na prvih dveh nivojih (in deloma tudi na tretjem) (Cerovšek, 2003).

Predstavljeni vsebinski delitvi modela sledi tudi sistematika označevanja posameznih dokumentov, ki sestavljajo standard STEP. Oznaki ISO 10303 sledi dodatna oznaka, in sicer v območju ISO 10303-1 do 10303-1xxx (privzeta preglednica 1).



Privzeta slika 3: Elementi standarda STEP (ISO 10303, 1992)

Adopted figure 3: STEP basic elements (ISO 10303, 1992)

Privzeta preglednica 1: Deli standarda STEP (ISO 10303, 1992)

Adopted table 1: STEP parts (ISO 10303, 1992)

okolje	1-9	Uvod
	1x	Metode opisa: EXPRESS
	2x	Metode implementacije (STEP fizične datoteke, STEP-XML, SDAI)
	3x	Metodologija in ogrodje za testiranje skladnosti
integrirani podatkovni modeli	4x in 5x	Integrirani generični viri
	1xx	Integrirani aplikacijski viri
	5xx	Integrirani sestavni deli aplikacij (Application Integrated Constructs) oz. aplikacijsko integrirani gradniki
	1xxx	Aplikacijski moduli (AM)
glavni deli	2xx	Aplikacijski protokoli (AP)
	3xx	Abstraktne zbirke (ATS) za testiranje aplikacijskih protokolov
	4xx	Implementacijski moduli za aplikacijske protokole



Standard STEP je zasnovan modularno. Osnovni koncepti standarda se ne spreminjajo, temveč se dodajajo le novi deli, po potrebi pa se opravljajo revizije ter popravki posameznih obstoječih delov.

### **2.2.2.3 Metode opisa**

Temelje informacijskega modeliranja predstavlja standardiziran način zapisa modelov. Ob pričetku razvoja jezika EXPRESS so bile že izoblikovane temeljne zahteve, katerim mora ustrezati sodoben jezik za modeliranje:

- berljiv za človeka,
- omogoča enostavno modeliranje kompleksnih produktov (oz. izdelavo shem),
- definiranje omejitev in operacij nad objekti,
- razširljivost,
- učinkovito računalniško procesiranje,
- enostavne in neodvisne implementacije.

Ker navedenim zahtevam v celoti ni ustrezal nobeden od obstoječih jezikov za modeliranje, je bil v okviru STEP iniciative razvit nov jezik, namenjen izključno zapisu konceptualnih shem. Z jezikom EXPRESS se specificirajo razredi izbrane domene (npr. stebri, vrata), lastnosti razredov (npr. dimenzije, barva) ter relacije in omejitve (npr. vsebovan, edinstven).

Kljub nekaterim skupnim lastnostim jezik za modeliranje ni programski jezik, saj ni prevedljiv. Jezik EXPRESS ne podpira kompleksnih operacij, vključene so osnovne aritmetične, relacijske in logične operacije. Jezik ne podpira podajanja zahtev, sporočil, metod, stanj objektov, obnašanja objektov, komunikacijskih modelov in opozarjanja na dogodke.

Prvi osnutki jezika so se pojavili leta 1987, prva uradna specifikacija pa je bila objavljena leta 1994. Za metode opisa so v standardu STEP rezervirani deli 1 do 19. Izpostaviti velja dva dela: del 1 – Pregled in temeljni principi ter del 11 – Priročnik za jezik EXPRESS.

Predstavitev jezika EXPRESS je v večji meri povzeta ((Eastman, 2000), (IAI-MSG, 1996a), (IAI-MSG, 1996a)).

### 2.2.2.3.1 EXPRESS

Tekstovni zapis predstavlja izvorni zapis jezika EXPRESS. V nadaljevanju bodo predstavljeni osnovni gradniki jezika, in sicer v obsegu, ki bo zadoščal za razumevanje shem, ki definirajo model zgradb IFC.

#### 2.2.2.3.1.1 Sintaksa jezika

Vrstice v jeziku EXPRESS se zaključujejo s podpičjem. Presledki in zamiki se pri interpretaciji sheme ter pri preverjanju skladnosti ignorirajo, vendar pa se zaradi preglednosti vseeno uporabljajo.

Vsa poimenovanja, ki jih v shemo vključijo končni uporabniki (npr. imena entitet), so označena z malimi črkami, vsa rezervirana imena pa z velikimi črkami. Presledki v imenih niso dovoljeni, običajno se uporabljajo podčrtaji.

Podobno kot pri ostalih jezikih ima tudi jezik EXPRESS definiran nabor rezerviranih besed, in sicer za funkcije (+, -, DIV, SIN, EXP, itd.), procedure (REMOVE, INSERT), operatorje (AND, NOT, OR, itd.) ter konstante (PI).

Komentarje označujemo s (\* komentar \*) ali na koncu vrstice s (-- komentar).

#### 2.2.2.3.1.2 Gradniki

Schema (SCHEMA ... END\_SCHEMA) predstavlja največjo enoto znotraj jezika EXPRESS. Shema združuje skupino EXPRESS elementov, ki tvorijo zaključeno celoto. Shema je lahko ena sama, možno pa je definirati tudi več shem z ustreznim sklicevanjem (REFERENCE FROM). Namesto ene same obsežne sheme je običajno bolj smiselno definirati več medsebojno povezanih shem.

Znotraj sheme se posamezno področje oz. domeno opisuje z uporabo entitet (oz. objektov) in tipov. Entitete so ekvivalentne razredom objektov pri objektno usmerjenem programiranju in predstavljajo kompleksne tipe, ki se razlikujejo od preprostih tipov po naslednjih lastnostih:

- entitete imajo lahko precej bolj kompleksno notranjo strukturo kot preprosti tipi;
- entitete omogočajo definicijo nad/pod-entitet ter sorodnosti med njimi;
- entitete imajo enolične identifikatorje, medtem ko le-ti pri preprostih tipih niso definirani;

- entitete lahko vsebujejo attribute in/ali vezi, ki natančneje opisujejo lastnosti entitet.

Entitete označuje ključna beseda `ENTITY`. Znotraj deklaracije se praviloma pojavijo atributi, ki opisujejo lastnosti entitet. Atribut določajo ime atributa, tip atributa ter lastnosti atributa (opcijski, inverzni ali edinstven). Kardinalnost označuje oznaka `OPTIONAL`. Če oznake ni, potem je atribut obvezen.

Enostaven primer:

```
ENTITY Vrata;
...
Material      : les;
Teža:         : OPTIONAL REAL;

END_ENTITY;
```

Relacije med entitetami običajno označujemo z generalizacijo: nadtip (`SUPERTYPE`) ter podtip (`SUBTYPE`).

Pri implementaciji sheme posamezne entitete instanciramo (npr. v zgradbi imamo deset instanc entitete vrata). Vseh entitet ni možno instancirati. Takšne entitete se imenujejo abstraktne entitete (oznaka `ABSTRACT` oz. kar `ABS`), ki se v hierarhiji sheme praviloma pojavljajo v višje ležečih slojih in opisujejo skupne lastnosti nižje ležečih entitet (npr. pohištvo bi lahko predstavljalo abstraktno entiteto vrat). Namesto termina instance se včasih uporablja tudi izraz objekt.

Glede na identifikacijo podatkovnih tipov ločimo:

- Osnovne tipe:
  - `NUMBER` predstavlja nadtip `INTEGER` in `REAL` (slednjemu tipu v oklepaju navedemo tudi natančnost),
  - `STRING` (dolžina niza) predstavlja niz znakov predpisane dolžine,
  - `LOGICAL` (pravilno, napačno, neznano),
  - `BOOLEAN` (pravilno, napačno) in
  - `BINARY` (vektor bitnih vrednosti).

Osnovni tipi se uporabljajo pri definiranju :

- poimenovanih tipov

```
TYPE površina : REAL;
```

- ali pa kot atributi entitet.

```
ENTITY stena;  
  ...  
  teza: REAL;  
  ...  
END_ENTITY;
```

- Poimenovane tipe
- Podatkovne tipe (enostavni (STRING), izbirni (SELECT) in naštevni (ENUMERATION OF)). Znotraj podatkovnega tipa se lahko definirajo le pravila.

```
TYPE barva = ENUMERATION OF (rdeča bela modra);  
  ... (* pravila *)  
END TYPE
```

- Entitetne podatkovne tipe. Z njimi lahko definiramo pravila in attribute.

```
ENTITY ime_entitete;  
  ... (* pravila *)  
  WHERE  
  ... (* pravila *)  
END TYPE;
```

- Agregatne tipe, ki povezujejo spremenljivke enakih tipov. Express pozna štiri različne agregatne tipe:

- ARRAY – polje: predstavlja zbirko (vektor) končnega števila urejenih elementov predstavljenih kot  $A[1:n]$ ;
- BAG – torba: predstavlja zbirko neurejenih elementov z dovoljenim podvajanjem  $B[1:?]$ ;
- LIST – seznam: podobno kot ARRAY, le da je število elementov lahko spremenljivo  $L[1:?]$ ;
- SET – množica: zbirka neurejenih elementov brez ponavljanja  $S[1:?]$ .

Primer (kot kritino strehe lahko uporabimo les, pločevino ali opečno kritino):

```
TYPE kritina  
  ...  
  Material : ARRAY [1:3] OF materiali_za_kritino;  
  Teža:      REAL;  
  ...  
END TYPE;
```

Združevati je mogoče tudi agregatne tipe.

Pri identifikaciji podatkovnih tipov so bila omenjena pravila, s katerimi lahko definiramo omejitve znotraj posameznega podatkovnega tipa, znotraj skupine podatkovnih tipov ali znotraj posamezne sheme. Nekaj osnovnih pravil:

- Inverzno pravilo (`INVERSE`). Nekateri jeziki za modeliranje dovoljujejo definicijo le enosmernih relacij. Enostaven primer: pri poligonu, sestavljenem iz robov je naravno definirati relacije poligon – robovi. Precej bolj neobičajna, a včasih nujno potrebna, je možnost podajanja relacije v obratni smeri – npr. če je posamezen rob del več poligonov, pri čemer so poligoni sestavljeni iz več robov.
- Pravilo enovitosti (`UNIQUE`) se pojavlja pri atributih, ki lahko zavzamejo samo neko specifično vrednost. Najbolj elementaren primer uporabe pravila enovitosti predstavlja opis atributa, katerega vrednost je enaka `GUID` (Globally Unique Identifier).
- Izpeljano pravilo (`DERIVE`). V določenih primerih je najprimerneje vrednost posameznih atributov izračunati iz vrednosti že definiranih atributov.
- Z domenskim pravilom (`WHERE`) definiramo omejitve, ki lahko nastopijo pri posameznih atributih.

V jezik `EXPRESS` so vgrajene funkcije in procedure, ki so poznane iz programskih jezikov – npr. klasične računske operacije (seštevanje, odštevanje, deljenje, množenje, potenciranje, logaritmične, trigonometrične funkcije ipd.). Možno je definirati tudi lastne funkcije in procedure. Podobno kot pri programskih jezikih kličemo funkcije z imenom ter navedenimi parametri: funkcija\_parametri.

Posamezni podatkovni tipi imajo lahko podrejene in nadrejene podatkovne tipe. Podobno kot pri objektno usmerjenem programiranju tudi v navedenem primeru veljajo zakonitosti dedovanja. Podrejeni tip podeduje od nadrejenega vse attribute, omejitve ter tudi morebiten način uporabe.

#### **2.2.2.3.2 EXPRESS-G**

`EXPRESS-G` predstavlja grafično (shematično) notacijo jezika `EXPRESS`. V večini primerov je prikaz produktnega modela v jeziku `EXPRESS-G` mnogo bolj nazoren in berljiv. Žal pa z

notacijo jezika ni mogoče prikazati vseh EXPRESS konstruktov. Navedena pomanjkljivost se nanaša predvsem na funkcije procedure, pravila omejitve ter implicitne attribute (Cerovšek, 2003).

Osnovni simboli jezika EXPRESS-G so prikazani na privzeti sliki 4. Gradnike v grobem razdelimo v dve skupini: v prvo uvrščamo simbole, s katerimi označujemo razrede, osnovne podatkovne tipe, podatkovne tipe ipd., v drugi pa se nahajajo simboli, ki prikazujejo odnose med elementi prve skupine.

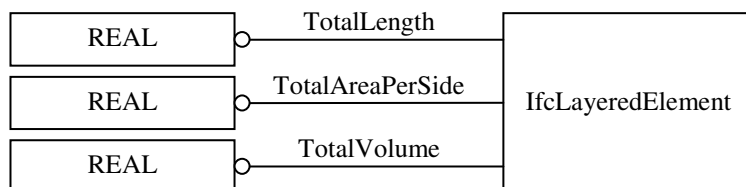
	ime	razred	relacija	obvezno razmerje (1)
	NIZ ZNAKOV	enostavni podatkovni tip	relacija	opcijsko razmerje (0, 1)
	naštevni tip	naštevni podatkovni tip	relacija S[1:?]	relacije (1 ali več)
	izbirni tip	izbirni podatkovni tip	relacija S[1:?]	relacije (0, 1 ali več)
	definiran tip	definiran podatkovni tip	relacija	inverzna relacija
	razred alias	»USE FROM« vmesnik	1	ekskluzivna relacija nadtip/podtip
	razred alias	»REFERENCE FROM« vmesnik		inkluzivna relacija nadtip/podtip
	stran, ref. razred	povezava na drugo stran		izbirna relacija nadtip/podtip
	stran, ref. (iz strani)	povezava iz strani		

Privzeta slika 4: Seznam EXPRESS-G simbolov (ISO 10303, 1992)

Adopted figure 4: List of EXPRESS-G symbols (ISO 10303, 1992)

Nekaj osnovnih primerov rabe predstavljenih simbolov je prikazanih na slikah 2–6.

- entitete, atributi

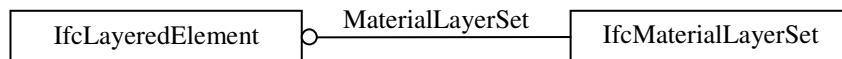


Slika 2: Entitete – atributi

Figure 2: Entity – attributes

Obvezni odnosi med entitetami in relacijami so označeni s polno črto, opcijski pa s prekinjeno. Krožec označuje odnos (atribut, vsebovan v entiteti). Ime atributa je vedno navedeno nad črto povezave.

- odnosi med entitetami



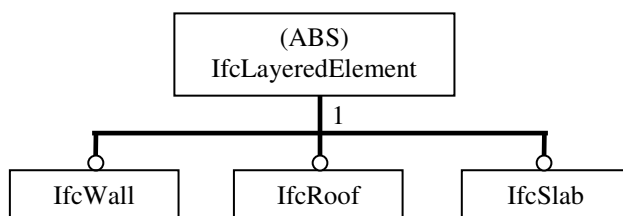
Slika 3: Odnosi med entitetami

Figure 3: Relations between entities

Odnosi ne nastopajo samo med entitetami in atributi, temveč tudi med entitetami. Predstavljen odnos lahko opišemo kot »IfcLayeredElement vsebuje natanko en IfcMaterialSet, ki ga predstavlja entiteta IfcMaterialLayerSet«. V določenih primerih bi bilo na prvi pogled morda enostavneje več odnosov opisati z enim samim imenom, vendar pa takšna poimenovanja v jeziku EXPRESS niso dovoljena.

Če uporabljamo agregatni podatkovni tip, potem je za imenom v zavutih oklepajih potrebno navesti tudi velikost polja: prva vrednost predstavlja minimalno vrednost, druga pa maksimalno oz. nedoločeno vrednost (označeno z ?).

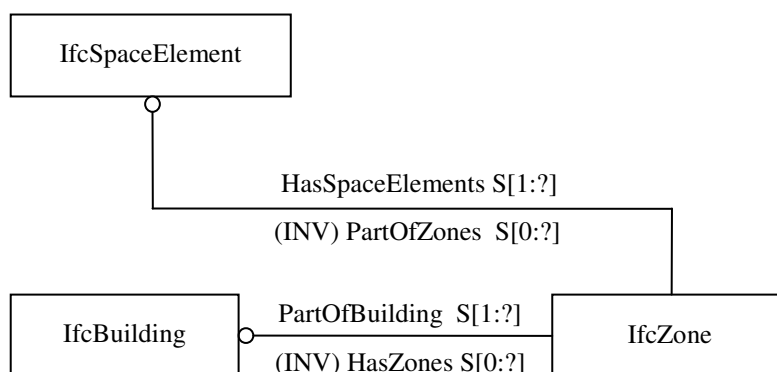
- Nadtipi in podtipi predstavljajo posebno vrsto odnosov zaradi zmožnosti dedovanja. Namesto oznake je (»is a«) se uporablja debela (dvojna) črta. Številka na dvojni črti (1) prikazuje ekskluzivnost podtipov (izberemo lahko le enega izmed navedenih podtipov).



Slika 4: Nadtipi in podtipi

Figure 4: Supertypes and subtypes

- Inverzno domensko pravilo oz. izpeljano domensko pravilo se označi v oklepajih, in sicer pred oznako relacij – (INV), (DER). Pri prikazanem primeru več prostorskih elementov tvori cono. Ker cona ni podrobneje definirana (lahko je varnostna, požarna ipd.), lahko en element nastopa v več conah. Za popoln opis je zato potrebno definirati tudi inverzni atribut in ga označiti pod črto, ki označuje odnos med entitetama. Uporabo domenskega pravila WHERE označuje zvezdica pred imenom relacije.

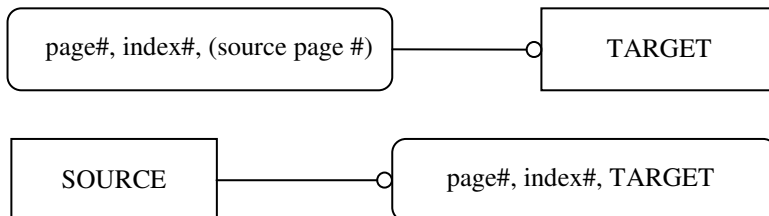


Slika 5: Inverzna pravila

Figure 5: Inverse rules



- EXPRESS-G diagrami že pri enostavnejših primerih presežejo obseg ene strani. Zato so v jezik vključeni konstrukti, ki omogočajo uporabo več strani: (ONTO ANOTHER PAGE, ONTO THIS PAGE).



Slika 6: EXPRESS-G shema na več straneh – sklicevanje

Figure 6: EXPRESS-G schema on several pages – referencing

EXPRESS-G diagrame lahko ločimo glede stopnje podrobnosti opisa (Cerovšek, 2003):

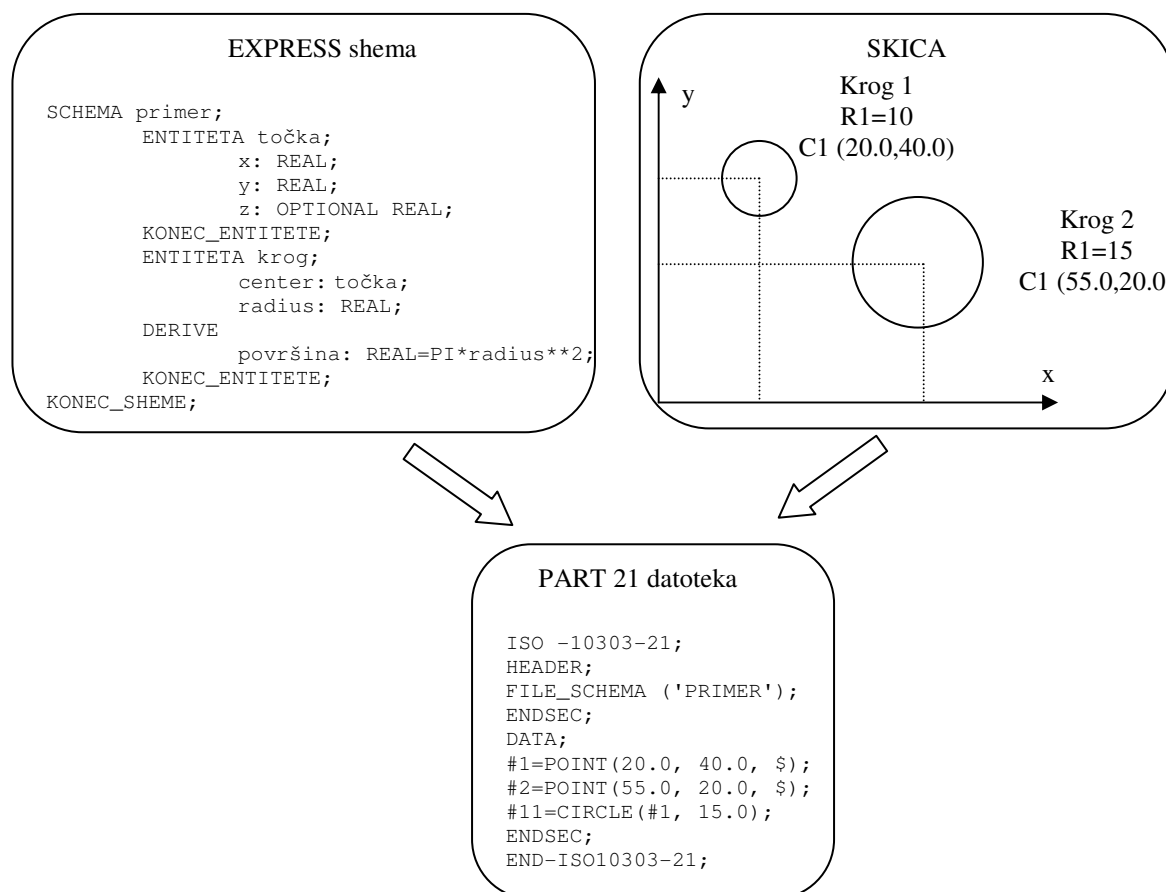
- osnovni entitetni diagram (vsebuje samo entitete),
- entitetni diagram z atributi (poleg entitet so vsebovani tudi atributi),
- entitetni diagram s tipi (poleg entitet so vsebovani tudi atributi ter tipi),
- popolni entitetni diagram (vsebovana je večina elementov jezika EXPRESS-G).

#### 2.2.2.4 Metode implementacije

Uporaba jezika EXPRESS za metodo opisa zagotavlja od implementacije neodvisen opis produkta, hkrati pa močno olajša implementacijo v programsko opremo. Standardizirani so:

- fizična izmenjava datotek – izmenjava podatkov s sekvenčnimi datotekami (del 21);
- XML predstavitev EXPRESS shem (del 28);
- programski vmesnik, imenovan standardni vmesnik za dostop do podatkov – SDAI (del 22);
- vezi med SDAI in C++ (del 23), C (del 24) ter IDL (Interface Definition Language) (del 26).

Fizične datoteke predstavljajo najbolj pogosto uporabljeno metodo implementacije. Enostaven primer STEP fizične datoteke je prikazan na sliki 7. Datoteko sestavljata dva dela: glava ter podatkovni del.



Slika 7: EXPRESS shema – STEP fizična datoteka

Figure 7: EXPRESS-G schema – STEP physical file

V glavi vsake datoteke se morajo nahajati instance naslednjih entitet: opis datoteke (»file\_description«), ime datoteke (»file\_name«) ter shema datoteke (»file\_schema«). V opisu datoteke je naveden neformalni opis vsebine ter stopnja implementacije (»implementation\_level«), ki določa razred skladnosti. Ime datoteke vsebuje berljive splošne informacije o strukturi, ki jo izmenjujemo. Atribut, ki določa datum in uro nastanka dokumenta, je določen v delu 21, medtem ko so ostali atributi (avtor, organizacija, predprocessor, izvorni sistem) lahko predpisani v posameznem aplikacijskem protokolu. Z

entiteto »file\_schema« se določa EXPRESS shema, ki specificira instance entitet, ki se pojavljajo v podatkovnem delu STEP datoteke (npr. AUTOMOTIVE\_DESIGN). V glavi se lahko nahajajo dodatne s strani končnega uporabnika dodane entitete. Le-te se pričenjajo z znakom »!«.

Podatkovni del (omejen z oznakama »DATA« in »ENDSEC«) vsebuje podatke o produktu, ki so namenjeni izmenjavi. Dovoljene so samo instance entitet, ki pripadajo v glavi navedeni shemi. Instance so v podatkovnem delu označene kot: #številka\_vrstice=entiteta (atributi); Številčenje vrstic je bistvenega pomena, in sicer zaradi sklicevanja. Atributi so lahko enostavni ali sestavljeni. Vsi atributi entitet morajo biti podani. Če vrednost posameznega atributa predstavlja že definirana instanca, potem se kot atribut vpiše sklic na navedeno instanco. Ni nujno, da so vrednosti vseh atributov predpisane – opsijske lahko nadomestimo z znakom \$. Podobno kot v glavi se tudi v podatkovnem delu lahko pojavijo instance entitet, ki jih definirajo uporabniki. Način označevanja je enak kot v glavi.

V STEP fizični datoteki se lahko nahajajo le objekti oz. podatkovni tipi, ki so definirani v EXPRESS shemi, ki je imenovana v glavi datoteke. Torej je za interpretacijo STEP fizične datoteke nujna tudi pripadajoča EXPRESS shema, ki pa je pri uporabi standardnih shem praviloma ni potrebno prenašati skupaj s STEP datotekami.

#### **2.2.2.5 Integrirani generični viri**

Integrirani (generični) viri (deli 4x in 5x) (privzeta preglednica 2) niso neposredno namenjeni implementaciji. Standard STEP namreč uporablja aplikacijske protokole kot mehanizme za specifikacijo zahtev po izmenjavi informacij ter za doseganje učinkovite in zanesljive datotečne izmenjave.

Privzeta preglednica 2: STEP – integrirani viri (ISO 10303-42, 1992)

Adopted table 2: STEP – integrated resources (ISO 10303-42, 1992)

ISO 10303	Ime	Status	Opis
41	Osnove opisa produkta in podpore	standard	EXPRESS sheme dela 41 tvorijo integracijsko jedro ki povezuje posamezne integracijske vire.
	Generični viri za opis produkta		Integrirano jedro standarda STEP, preko katerega se povezuje delne modele.
	Upravljanje z viri		Objekti za administracijo v kontekstu aplikacije.
	Podpora virom		Splošni objekti za opis produkta – imena, časovne in podatkovne informacije, merske enote ipd.
42	Geometrijska in topološka predstavitev	standard	Geometrija, topologija (točke, vektorji, krivulje, ploskve).
43	Strukture prikaza	standard	Specifikacija načina predstavitve produkta (npr. predstavitev geometrijskih elementov v koordinatnem sistemu). Relacije med predstavitvami. Ponovna raba struktur prikaza.
44	Konfiguracija strukture prikaza	standard	Prikaz strukture in konfiguracije produkta.
45	Materiali	standard	Materialne lastnosti produkta.
46	Vizualna predstavitev	standard	Predstavitev parametrov in pravil za vizualno (simbolično ali realistično) predstavitev produkta.
47	Tolerance	standard	Specifikacija toleranc.
48	Orisi oblik	umaknjeno	
49	Procesne strukture in lastnosti	standard	Definicija logičnega zaporedja procesnih aktivnosti (in njihovih parametrov).

Ker opis geometrije modela zgradb IFC v veliki meri temelji na delu 42, so v privzeti preglednici 3 podrobneje predstavljeni gradniki posameznih predstavitev.

Privzeta preglednica 3: Entitete v ISO 10303-del 42 (ISO 10303-42, 1992)

Adopted table 3: Entities in ISO 10303-part 42 (ISO 10303-42, 1992)

GEOMETIJA	TOPOLOGIJA	GEOMETRIJSKI MODEL
točka (point)	nabor povezanih robov (connected_edge_set)	trdni model (solid_model)
vektor (vector)	nabor povezanih ploskev (connected_face_set)	pol prostorsko trdni model (half_space_solid)
krivulja (curve)	rob (edge)	blok (block)
ploskev (surface)	ploskev (face)	torus (torus)
	omejena ploskev (face_bound)	desnosučni stožec (right_circular_cone)
	zanka (loop)	desnosučna zagozda (right_circular_wedge)
	pot (path)	lupinski model (shell_based_surface_model)
	predmet predstavitve (representation item)	ploskovni model (face_based_surface_model)
	točke-stičišča oz. vozlišča (vertex)	lupinski žični model (shell_based_wireframe_model)
	stičišče lupin (vertex_shell)	robni žični model (edge _bfased_wireframe_model)
	žična lupina (wire_shell)	nabor geometrije (geometric_set)
		boolean rezultat (boolean_result)
		desnosučni cilinder (right_circular_cylinder)
		sfera (sphere)

### 2.2.2.6 Aplikacijski generični viri

Če so generični viri, ki jih uporablja posamezen aplikacijski protokol, vezani na eno samo domeno, potem namesto o integriranih generičnih virih govorimo o integriranih aplikacijskih virih (privzeta preglednica 4). Standard STEP slednje deli v dve skupini:

- **Splošno veljavni integrirani aplikacijski viri** – del 1xx. Namen predstavljenih virov je najbolj nazorno moč prikazati na primeru geometrije produkta: geometrijski

konstrukti iz dela 42 se v delu 101 specializirajo v kontekstu tehničnih risb (določena skupina črt – daljic se deklarira kot kotirne črte).

Privzeta preglednica 4: Splošno veljavni integrirani aplikacijski viri – deli 1xx (ISO 10303-42, 1992)

Adopted table 4: Universally valid application models – 1xx series (ISO 10303-42, 1992)

ISO 10303	Ime	Status
101	Tehnično risanje	standard
104	Analiza po metodi končnih elementov	standard
105	Kinematika	standard
106	Gradbeništvo – jedro modela	umaknjeno
107	Analiza po metodi končnih elementov – definicija relacij	komisijski delovni osnutek
108	Parametrizacija in omejitve pri eksplicitnem produktnem modeliranju	delovni osnutek
109	STEP sestavni model produktov	delovni osnutek
110	Mreže končnih elementov pri izračunih dinamike plinov in tekočin	delovni osnutek

- **Aplikacijsko interpretirani konstrukti** – del 5xx (Application Interpreted Constructs - AIC) se v večini nanašajo na podroben opis geometrije modelov z uporabo objektov, definiranih v ISO 10303-42 (privzeta preglednica 5). Dodatno definirani pogoji namreč specializirajo generično predstavitev oblike, ki je podana v delu 42. S tem je zagotovljeno, da vsi aplikacijski protokoli uporabljajo isto specifikacijo pri opisu npr. žičnih modelov.

Privzeta preglednica 5: Aplikacijsko interpretirani konstrukti – deli 5xx (ISO 10303-42, 1992)

Adopted table 5: Application Interpreted Constructs – 5xx series (ISO 10303-42, 1992)

<b>ISO 10303</b>	<b>Ime</b>	<b>Status</b>
501	Robni žični model	standard
502	Lupinski žični model	standard
503	2D žični model – geometrijsko omejen	standard
504	Razlaga tehničnega risanja	standard
505	Struktura risb ter administracija	standard
506	Elementi tehničnega risanja	standard
507	Geometrijsko omejene površine	končni osnutek standarda
508	Nemnogovrstne površine	končni osnutek standarda
509	Mnogovrstne površine	končni osnutek standarda
510	Geometrijsko omejeni žični modeli	standard
511	Topološko omejene površine	standard
512	Mejni prikaz (B-rep)	standard
513	Elementarni mejni prikaz	standard
514	Napredni mejni prikaz	standard
515	Konstruktivna geometrija trdnih teles	standard
517	Načrti v strojništvu – geometrijska predstavitev	standard
518	Načrti v strojništvu – senčenja	osnutek standarda
519	Geometrijske tolerance	standard
520	Asociativni elementi tehničnega risanja	standard

### 2.2.2.7 Aplikacijski protokoli (AP)

Domeno izmenjave definiramo z aplikacijskimi protokoli. Obseg posameznega AP med drugim določajo:

- vrsta produkta,
- faza oz. faze življenjskega cikla produkta, ki ga želimo modelirati,

- discipline, ki bodo uporabljale podatke o produktu,
- namen uporabe podatkov o produktu.

Aplikacijske protokole lahko poenostavljeno razumemo kot selektivni filter za specifikacijo konceptov generičnih virov (del 41–49), potrebnih za opis potreb posamezne industrije. Tudi razvrstitev aplikacijskih protokolov temelji na obsegu podpore posameznim fazam v življenjskem ciklu ter nadalje na pripadnosti industrijskim panogam. V prvi fazi aplikacijske protokole razdelimo v tri skupine: načrtovanje, proizvodnja ter podpora produktom v njihovem življenjskem ciklu. Aplikacijske protokole, namenjene načrtovanju, lahko razdelimo še podrobneje, in sicer na protokole, namenjene strojništvu, gradbeništvu, elektrotehniki, elektrotehniki elektroniki, instalacijam ter ladjedelništvu.

Del 225 z nazivom »elementi zgradb z eksplicitnim prikazom oblik« (»Building elements using explicit shape representation«) predstavlja edini gradbeniško specifičen aplikacijski protokol. Ker opis temelji na izmenjavi geometrije, definirane v integriranih generičnih virih (ISO 10303-42), navedeni aplikacijski protokol ne ustreza potrebam sodobnega informacijskega modeliranja.

Po nekaterih razvrstitvah naj bi v skupino gradbeniških aplikacijskih protokolov sodil tudi del 202: Asociativno tehnično risanje (»Associative draughting. 2D/3D drawing with association«). V njem je definirana hierarhična struktura risb, ki opisujejo produkte, ter administrativni podatki, potrebni za interpretacijo določenih delov risb (npr. kotiranja ter opomb).



## **2.3 Informacijski model zgradb IFC**

Informacijski model zgradb IFC predstavlja trenutno najbolj aktualni informacijski model zgradb, s katerim bi bilo možno opisati celotno področje gradbeništva. Specifikacijo IFC razvija mednarodno združenje za interoperabilnost (IAI – International Alliance for Interoperability). Tehnična specifikacija temelji na standardu ISO 10303, vsebinsko pa uporablja koncepte, razvite v prilogi B predstavljenih iniciativah (BCCM, GARM, CIMsteel in drugih).

### **2.3.1 Splošna predstavitev modela zgradb IFC**

#### **2.3.1.1 Zgodovina modela zgradb IFC**

Model zgradb IFC predstavlja odziv na pobudo podjetij iz ZDA za zagotovitev neodvisne izmenjave podatkov med različnimi AEC-FM programskimi orodji. Iniciativa iz leta 1994 je poleg ugodnega odziva v celotnem AEC-FM sektorju pritegnila tudi številne razvijalce programske opreme in je botrovala ustanovitvi Industrijskega združenja za interoperabilnost (IAI – Industry Alliance for Interoperability). Uspešnost združenja se kaže v hitrem širjenju organizacije ter v petih izdanih različicah specifikacije (prva je bila izdana novembra 1996) ter v številnih implementacijah v AEC-FM programskih orodjih. Vizijo združenja, ki danes povezuje preko 600 članov iz 18 držav, predstavlja izboljšanje komunikacije, produktivnosti, skrajšanje procesov, znižanje stroškov ter zagotavljanje oz. povečanje kakovosti skozi celoten življenjski cikel zgradbe, in sicer z generacijo modela zgradb IFC, ki naj bi predstavljal »de facto« standard za formulacijo in elektronsko izmenjavo informacij v AEC-FM sektorju. Člane združenja (arhitekta, inženirje, izvajalce gradbenih del, lastnike, upravljavce, raziskovalce, državne uradnike, razvijalce programske opreme) povezuje vizija interoperabilnosti. IAI je organizirana v geografska oz. jezikovna združenja, imenovana poglavja. Njihovo število neprestano raste, trenutno jih je enajst: severnoameriško, nemško govoreče, nordijsko, francosko, singapursko, korejsko, avstralsko-azijsko, japonsko, angleško, ibernsko, italijansko in kitajsko (IAI, 2008).

Organizacija IAI tesno sodeluje s številnimi organizacijami s podobnim oz. sorodnim ciljem razvoja nevtralnega formata za komunikacijo med programsko opremo:

- ISO TC184/SC4 – razvoj standardov za avtomatizacijo in integracijo v industriji,
- ISO TC59/SC13/WG6 – razvoj standardov za objektno orientirano izmenjavo informacij,
- ICIS – International Classification Information Society – priprava standardov na področju specifikacije del in stroškov v gradbeništvu.

Zaradi obsežnosti industrije, ki oblikuje grajeno okolje, je le-ta znotraj združenja IAI razdeljena na domene, ki predstavljajo ozko specializirano disciplino (ogrevanje, prezračevanje, osvetlitev, akustiko ipd.) oziroma proces (načrtovanje, organiziranje gradnje ipd.). Vsaka domena ima poslovnega menedžerja ter tehničnega koordinatorja. Domenska komisija je praviloma sestavljena iz članov posameznega poglavja. Če določeno poglavje ne more zagotoviti interdisciplinarnosti komisije, se le-ta razširi s člani ostalih poglavij. Pripravljeni predlogi se pred vključitvijo v standard IFC dodatno usklajujejo s krovno organizacijo.

Tehnične aktivnosti koordinira komite za mednarodni tehnični menedžment (ITM – International Technical Management Committee). Člani komiteja (tehnični koordinatorji posameznih poglavij, produktni menedžerji posameznih različic specifikacije IFC) usklajujejo projekte, ki tečejo znotraj domen, ocenjujejo izvedljivost in prioriteto projektov ter dodeljujejo naloge skupinam za podporo. Le-te so organizirane v štiri skupine:

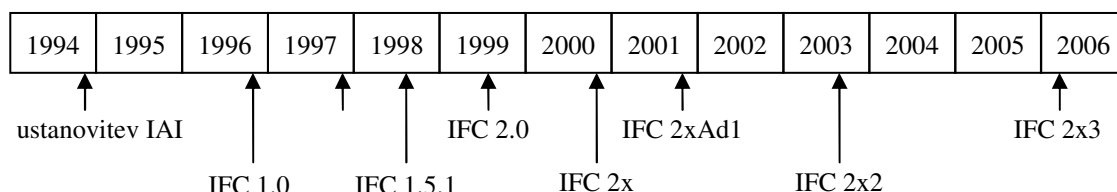
- skupina za podporo modelom (MSG – Model Support Group) skrbi za razvoj in integracijo modela zgradb IFC in pripadajoče dokumentacije;
- skupina za podporo implementaciji (ISG – Implementation Support Group) zagotavlja kakovost ter implementacijo modela IFC. S skupino za podporo modelom preverja koncepte razvoja modelov in skrbi za nedvoumnost navodil za praktično implementacijo. Podeljevanje certifikatov o skladnosti in demonstracija sta prav tako v domeni te skupine;
- skupina za podporo XML (XSG – XML Support Group) se ukvarja predvsem z razvojem in podporo modelov IFC v XML zapisu;
- skupina za tehnično svetovanje (TAG – Technical Advisory Group) podaja smernice za razvoj, predlaga nove tehnologije, ipd.

Standardizacija IFC informacijskega modela zgradb je obsežen in kompleksen proces, ki ga ni mogoče izvršiti v enem koraku. Takšno strategijo je moč zaslediti pri razvoju modela zgradb IFC, ki smiselno sledi življenjskemu ciklu gradbenih objektov.

Predstavljena industrijska pobuda iz leta 1994 je doživela sedem uradnih izdaj (sliki 8, 9):

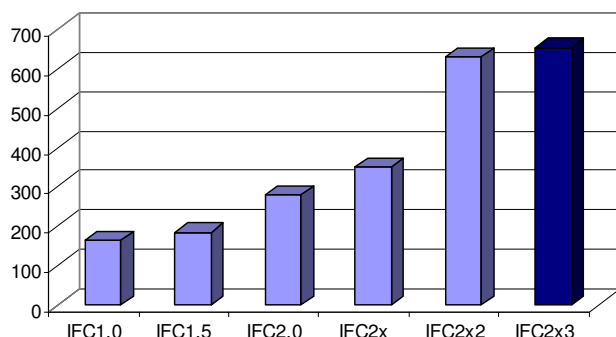
- IFC 1.0 (november 1996): v prvi različico IFC specifikacije je bilo vključenih le manjše število procesov s področja arhitekture, HVAC, vzdrževanja zgradb ter ocene stroškov;
- IFC 1.5 (november 1997): obseg druge različice IFC specifikacije je sicer enak kot v prvi, vendar sta bila izboljšana tehnična arhitektura ter jedro;
- IFC 1.5.1 (junij 1998): z nadgradnjo predhodne različice so bili rešeni problemi z implementacijo, izboljšano pa je bilo tudi jedro. Predstavljena različica predstavlja prvo implementirano različico IFC specifikacije (vmesniki za Archicad, Allplan, Architectural Desktop), podelijo pa se prvi certifikati skladnosti;
- IFC 2.0 (maj 1999): različica ne posega v jedro modela zgradb IFC, temveč razširja obseg specifikacije. Bistvena je širitev znotraj obstoječih področjih ter na ostala, še ne obdelana področja AEC/FM sektorja. Istočasno z objavo IFC 2.0 specifikacije se prične delo na projektu BLIS, s katerim naj bi se pospešila implementacija;
- IFC 2x (november 2000): predstavljena razvojna stopnja predstavlja prelomnico v razvoju IFC specifikacije. Izdelana je bila namreč platforma IFC2x z vsebovanimi razredi, ki predstavljajo ogrodje za integracijo različic modelov zgradb, temelječih na 2x platformi. Lastnosti objektov so ločene od objektov in se podajajo kot dodatne lastnosti (in ne kot atributi). Ta različica pridobi oznako ISO, in sicer ISO PAS 16739. Prvič je specifikacija na voljo tudi v obliki XML;
- IFC 2x Dodatek 1 (november 2001): odpravljene so bile manjše nekonsistence v modelu in dokumentaciji, in sicer brez vpliva na strukturo datotek;
- IFC 2x2 (maj 2003): predzadnja uradna različica predstavlja razširitev modela zgradb IFC na platformi 2x, in sicer tudi na področje računske analize konstrukcij. Kot alternativni pristop k zapisu informacijskega modela zgradb verzija 2x2 definira celoten IFC model v XML shemi (XSD – XML Schema Definition Language);

- IFC 2x2 Dodatek 1 (2004): glede na predhodno različico so bile odpravljene manjše napake. Poleg specifikacije je bila dopolnjena tudi dokumentacija z izboljšanimi napotki za implementacijo ter praktičnimi primeri;
- IFC 2x3 (februar 2006): predstavljena različica odpravlja manjše nekonsistentosti modela in razširja njegov obseg. Zadnji manjši popravki specifikacije so bili objavljeni julija 2007.



Slika 8: Zgodovina modela zgradb IFC

Figure 8: IFC building model history



Slika 9: Število entitet v posamezni IFC različici

Figure 9: Number of entities in different IFC releases

Model zgradb IFC temelji na strukturiranem standardizacijskem pristopu, kjer je specifikacija objavljena pred implementacijami v programsko opremo. Takšen pristop je zaradi nezmožnosti sočasne implementacije, testiranja ter iteracijskih izboljšav specifikacije precej neobičajen. Poglavitni očitak takšnega pristopa predstavlja nezmožnost ocene ustreznosti standarda glede postavljenih zahtev (Berhman, 2002). Citirani avtor favorizira minimalističen

pristop z uporabo standarda XML ter neposredno vključitvijo razvijalcev programske opreme. Navedeno izbiro utemeljuje z uspehi predstavljenega principa pri hitro razvijajočih se internetnih standardih. Navedeni argumenti sicer v večji meri veljajo, vendar pa primernost minimalističnega pristopa pri standardizaciji kompleksnih modelov ni bila v praksi nikoli potrjena (Kivinemi, 2005).

### 2.3.1.2 Obseg IFC modela zgradb

V preglednici 4 so navedena področja pokrita v različici IFC 2x3.

Preglednica 4: Področja, pokrita z IFC specifikacijo

Table 4: AEC-FM areas included in IFC specification

geometrija	elementi stavb (vrata, okna, stopnice, ipd.)
geometrija – napeljave oz. vodi	elektrosistemi (stikala, vtičnice, ipd.)
HVAC (ventilatorji, grelci, toplotne črpalke, ipd.)	požarna zaščita (hidranti, škropilci, ipd.)
sanitarni elementi	odnosi med elementi (odprtine, cone, ipd.)
pohištvo	cone (požarne, jaški, ipd.)
prostori, prostorska struktura	osvetlitev
razni sistemi (cevovodi, kabli, ipd.)	strukturni elementi (profili, stiki, ipd.)
računska (statična) analiza konstrukcije	mreže (vezava elementov na mrežo – 2D ali 3D)
nadzorni instrumenti	upravljanje zgradb
označevanje (tip črt, šrafure, ipd.)	splošno (garancija, navodila za uporabo, ipd.)
vplivi na okolje	omejitve (zahteve, specifikacije, ipd.)
časovni potek dogodkov	podpiranje elementov (izolacija vibracij, ipd.)
konzole in vešalke	stroški
akterji procesov (posamezniki, organizacije, ipd.)	naročila (dela, materiala, ipd.)
načrti dela ter terminski plani	klasifikacije
zunanji podatki	zunanji dokumenti, vezani na produkte

Specifikacija informacijskega modela zgradb je obsežen in kompleksen proces, ki ga ni moč izvesti v enem koraku. Takšno strategijo je moč zaslediti tudi pri razvoju modela zgradb IFC. Uporabljena je večslojna arhitektura modela, ki omogoča širitev IFC specifikacije na preostala, še ne pokrita področja.

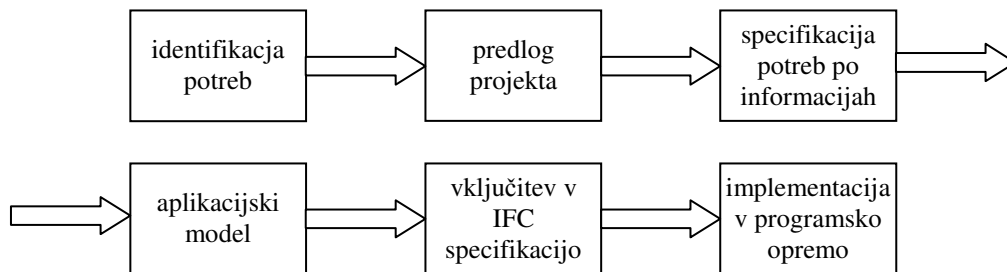
### **2.3.1.3 Širitev modela zgradb IFC**

Obseg IFC modela se širi z razširitvenimi projekti. Po identifikaciji potrebe ter po odobritvi projekta razširitve delo poteka po natančno predpisani določeni metodologiji (IAI, 2008):

- scenariji uporabe predstavljajo opise procesov, ki jih uporabniki izvajajo (npr. kako upravljavec vzdržuje posamezne elemente zgradbe – npr. dvigala). Scenariji uporabe zajamejo vse odločitve in informacije, ki jih potrebujemo v vsaki stopnji procesa;
- procesni diagrami omogočajo vizualno predstavo procesa, ki ga definiramo. Gre torej za grafično predstavitev uporabniškega scenarija;
- določijo se razredi, ki predstavljajo objektno orientirane programske komponente pri definiranju objektov. V razredu so lahko predstavljeni fizični objekti (npr. vrata) ali pa bolj abstraktni objekti oz. procesi (npr. cena vrat ali postopek za montažo vrat);
- definirajo se atributi, ki podrobneje opisujejo AEC-FM objekte. Atribut omenjenih vrat je lahko smer odpiranja, material ipd.;
- opiše se razmerja (relacije) med razredi, ki določajo vzajemno delovanje objektov. Z relacijami tako na primer določimo, da v določeno odprtino v določeni steni sodijo točno določena vrata;
- ustvari se IFC model. Le-ta predstavlja razrede, njihove attribute in razmerja. Za grafično ponazoritev je praviloma uporabljen jezik EXPRESS-G;
- v zadnji fazi se izdelata testna primera. Razvijalcem programske opreme omogočajo testiranje njihovih aplikacij.

Celotno razširitev modela zgradb IFC označuje termin »dvostopenjski proces standardizacije modela IFC« (slika 10). Poleg že opisane prve faze (razvoj aplikacijskega modela in implementacija z uporabo trenutno veljavne platforme) termin vključuje tudi integracijo v prihodnjo platformo (v roku 2–3 let). Pogoj za izvedbo druge faze je raba aplikacijskega

modela v praksi ter zahteva po vključitvi razširitve s strani vsaj treh razvijalcev programske opreme.



Slika 10: Širitev modela zgradb IFC

Figure 10: IFC model extension

Trenutno je v teku več razširitvenih projektov, ki predstavljajo nove predloge ali reciklažo starejših razširitev, ki so se izkazale za nepopolne ali za neprimerne. Več informacij o razširitvenih projektih je moč najti na spletni strani IAI (IAI, 2008).

#### 2.3.1.4 Izmenjava informacij in implementacije

Posamezni poskusi implementacije celotne (oz. večjega dela) IFC specifikacije so se namreč izkazali za nezanesljive oz. nezadostno robustne, da bi jih bilo moč zanesljivo uporabiti (Hietanen, 2006). Že od različice 2.0 dalje je prevladalo mnenje, da je model zgradb IFC preobsežen za implementacijo v celoti. Ugotovitve, ki jih je podal Bazjanac (2002) in ki so navedene v poglavju 1.2, v veliki meri veljajo še danes. Kljub temu je izmenjava fizičnih datotek še vedno najbolj pogosto implementirana rešitev izmenjave informacij o produktu v gradbeništvu.

Kot možno rešitev problema številni raziskovalci pogostokrat favorizirajo spletni strežnik modela zgradb IFC. V ozadju takšnega sistema je centralizirana spletna podatkovna baza. Raziskave navedenega področja kažejo, da je takšen način shranjevanja modelov sicer najboljši, vendar pa je način, kako tak način doseči, še ne povsem izoblikovan oz. nejasen (Amor, Faraj, 2001). Med drugim so izpostavljene naslednje ugotovitve:

- objektno usmerjeni pristop ne rešuje vseh težav pri modeliranju;

- centralizirano spletno skladišče še ne predstavlja »enotnega modela vsega«;
- kljub uporabi centraliziranega spletnega skladišča se rabi oz. izmenjavi na nivoju dokumentov ni mogoče povsem izogniti;
- opisan način shranjevanja produktnih modelov ne rešuje vseh težav s pravicami in lastništvom posameznih delov modela.

Predstavljeni način se kljub prednostim (še) ni uveljavil v praksi – trenutno obstajata le dva komercialna ponudnika spletnega strežnika zgradb.

#### **2.3.1.5 »Pogledi na model«**

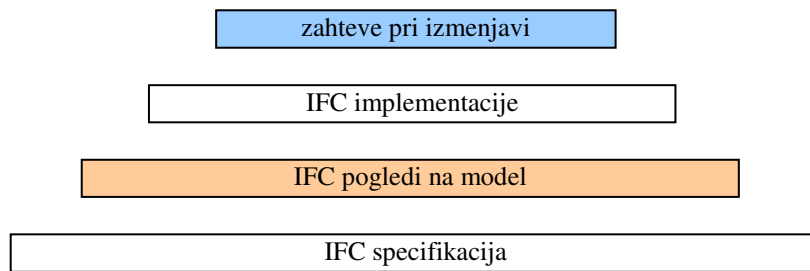
Zaradi preobsežnosti modela IFC bi bilo potrebno med razvijalci programske opreme in stroko doseči dogovor o vrsti informacij, ki jih bo moč izmenjevati in posledično določiti podmodel, ki bo implementiran v individualno programsko opremo. Takšni modeli so bolj obvladljivi, hkrati pa je z manj vloženega truda moč doseči bolj konsistentno implementacijo. Temelji opisanega dogovora so bili opredeljeni v projektu BLIS (BLIS, 2004), v katerem opisani dogovor označuje termin »pogled na model« (»view of the model«).

Tehnično je opis izveden s pomočjo konceptnih blokov. Le-ti opisujejo funkcionalnost »manjših« kosov IFC specifikacije. Posamezne implementacije IFC specifikacije lahko vsebujejo enak (zaželeno), večji ali manjši nabor entitet, kot je vsebovan v posameznem »pogledu na model«. Teoretično je možno za posamezen primer izmenjave informacij definirati še ožji nabor entitet, vendar pa se ob tem omejuje tudi deskriptivnost (slika 11).

Pogledi na model so bili najprej definirani za različico IFC 2.0: 1.) arhitektura – popis del, 2.) HVAC instalacije – popis del, 3.) arhitektura – toplotni odziv zgradb (+ HVAC instalacije), 4.) zahteve naročnika in umestitev v prostor – arhitektura, 5.) CAD – vizualizacija.

Izmed navedenih pogledov so za okvire te naloge zanimive relacije med arhitekturnim in računskim modeliranjem (termin računsko modeliranje se nanaša na izdelavo računskega modela). Predstavljeni pogled skuša definirati minimalni nabor informacij, vsebovanih v arhitekturnem modelu, ki jih je moč koristno uporabiti pri računskem modeliranju.





Slika 11: Nivoji interoperabilnosti IFC modela

Figure 11: Levels of interoperability inside IFC model

Za IFC specifikacije od različice 2x naprej je sicer zamišljenih precej več pogledov, žal pa je natančneje opredeljen le razširjen koordinacijski pogled. Vsi podeljeni in trenutno veljavni certifikati IFC vmesnikov temeljijo na koordinacijskem pogledu, ki obsega v preglednici 5 predstavljene gradnike oz. načine opisa.

Preglednica 5: Pogledi na model – IFC 2x3

Table 5: Model view – IFC 2x3

»pogled na model«	referenca	status
arhitektura – umestitev v prostor	CRC_CI-003	zamisel
arhitektura – popis del (prva stopnja)	VBL-004	zamisel
arhitektura – popis del (druga stopnja)	VBL-005	zamisel
arhitektura – popis del (tretja stopnja)	VBL-006	zamisel
arhitektura – računsko modeliranje (načrtovanje) nosilne konstrukcije	VBL-002	zamisel
arhitektura – toplotni odziv zgradbe	VBL-007	osnutek
razširjen koordinacijski pogled	ISG-001	specificiran
razširljivost	VBL-003	zamisel
upravljanje zgradb – inventure	GSC-001	zamisel
stopnja podrobnosti oddaje IFC modelov – organizacija GSA	GSA-001	osnutek
simulacija klimatskih pogojev v notranjosti zgradb za potrebe HVAC načrtovanja	HTU-HVAC-001	osnutek
umestitev v prostor – načrtovanje cest	CRC-CI-002	zamisel
načrtovanje cest – umestitev v prostor	CRC-CI-001	zamisel
računsko modeliranje – računska analiza	VBL-001	predlog

Preglednica 6: Koordinacijski pogled – IFC 2x3

Table 6: Coordination view – IFC 2x3

<b>elementi zgradb</b>			
prečke	stebri	kritje	proksi
vrata	elementi	plošče	strehe
ograje	rampe	rama rampe	okna
stopnišče	stopniščne rampe	stene	zunanje nosilne stene
<b>prostorski vsebovalniki</b>			
projekt	gradbišče	zgradba	etaža
prostor			
<b>ostali elementi</b>			
pomožni elementi	pohištvo	odprtine	transportni elementi
virtualni elementi	cone		
<b>materiali</b>			
osnovni material	seznam materialov	sloji materialov	nabor slojev materialov
<b>s predstavitvijo povezani gradniki</b>			
gradniki predstavitve	gradniki geometrijske predstavitve	gradniki topološke predstavitve	gradniki slogov predstavitve
preddefinirani gradniki	profili	kontekst predstavitve	preslikave pri predstavitvah
aspekti predstavitve	predstavitev	predstavitev oblike	predstavitev topologije
predstavitev produkta	specifikacija barv	krivulje (barva, fonti slogi)	zunanje reference
osi mreže	osvetlitev	mere in enote	postavitev objektov
virtualne mreže	dodatne lastnosti		
<b>generični nadrazredi</b>			
korenski razred	tipski objekt		

Osrednji del predstavljenega pogleda predstavlja geometrija produkta. V najbolj splošnem primeru geometrija zgradbe kot produkt dela arhitekta predstavlja pglavitno referenco pri načrtovanju računskega modela. Minimalni nabor entitet za izmenjavo predstavljajo projekt, gradbišče, zgradba, etaže, stene, stebri, prečke, plošče, stopnišča, rampe, mreže ter generični s strani končnega uporabnika definirani nadomestni elementi (proxy). Navedenim entitetam pripadajo atributi, ki so glede na IFC specifikacijo lahko obvezni ali neobvezni. Izmenjavo bi

bilo potrebno definirati vse do nivoja atributov, saj teoretično lahko posamezen pogled zahteva definirane vrednosti sicer neobveznih atributov.

Predstavljeni pogled opisuje le zamisel in zato o njem v literaturi ni moč najti podrobnejšega opisa. V nadaljevanju bo zato izdelana lastna specifikacija informacij (oz. entitet IFC specifikacije), potrebnih za učinkovito načrtovanje pri arhitekturnem in računskem modeliranju konstrukcije.

Nezadostnosti opisa pogledov na model se zaveda tudi organizacija IAI, ki naj bi v prihodnjih letih poleg razvoja domenskih modelov pozornost namenila tudi specifikaciji delnih modelov.

#### **2.3.1.6 »Implementacijski dogovor«**

Pri implementaciji BLIS »pogledov na model« se je izkazalo, da kljub predpisanim omejitvam posamezna področja niso opredeljena dovolj dobro, da bi bil opis lahko povsem nedvoumen. Takšno stanje je bilo zaradi izbranega standardizacijskega pristopa pričakovano, edino možno rešitev predstavlja dodatni dogovor o izkazanih nejasnostih. Takšen dogovor označuje termin »implementacijski dogovor«. Implementacijski dogovori se praviloma definirajo za vsak pogled na model posebej. Največkrat se konkretni dogovori pojavijo ob prvih implementacijah posameznih pogledov. Dogovori niso obvezen del specifikacije, temveč priporočljiv in lahko bistveno prispevajo k nedvoumnosti posameznih pogledov.

#### **2.3.1.7 Implementacije modela zgradb IFC ter certificiranje**

V skladu z idejo modela zgradb IFC je specifikacija modela prosto dostopna na spletu (IAI, 2006). Vsi razvijalci programske opreme imajo tako možnost izdelati ustrezne vmesnike, skladne z modelom zgradb IFC.

Organizacija IAI podeljuje certifikate vmesnikom IFC, in sicer ločeno za zmožnost branja in pisanja. Certificiranje je vezano na posamezen »pogled na model«. Programska oprema se trenutno preskuša za edini definirani razširjeni koordinacijski pogled na model (koordinacijskemu pogledu so lahko dodani še slogi upodabljanja (Rendering), 2D elementi, HVAC in elektroinštalacije). Trenutno certificirana programska oprema (IFC 2x3) je zbrana v preglednici 7. Dejansko število IFC 2x3 vmesnikov je nekajkrat višje. Proizvajalci programske opreme namreč predvsem zaradi nezanemarljivih stroškov in preskromne dodane vrednosti certificiranje opuščajo.

Preglednica 7: Rezultati certificiranja (oktober 2007) – IFC 2x3

Table 7: Certification results (October 2007) – IFC 2x3

aplikacija	prva stopnja	druga stopnja	opomba
Nemetschek Allplan 2006.2, 2008	•	•	
Graphisoft Archicad 11	•	•	
Bentley Architecture 8.9.3	•	•	
Revit Architecture 6.4	•	•	
TEKLA Structures 13	•	•	brez 2D
Archimen Active 3D v4.0	•	•	samo uvoz
Solibri Model Checker	•	•	samo uvoz, brez 2D
Autodesk Autocad Architecture 2008	•	•	
DDS HousePartner 6.4	•	•	
Progman MagiCAD	•	•	
Vizelia FacilityOnline	•	•	
SCIA ESA PT	•		
Nemetschek NA VectorWorks	•		
NorConsult IFC3DX	•		samo uvoz

Postopek certificiranja je natančno opredeljen (Lieblich, Wix, 2000): najprej se na dvodnevni delavnici pod okriljem organizacije IAI ugotavlja sposobnost vmesnikov za zapis osnovnih gradnikov IFC modela oz. konkretnega pogleda. Vmesnikom, ki so uspešno prestali prvo fazo testiranja, se podeli certifikat skladnosti za prvo stopnjo. Druga stopnja testiranja sledi po šestih mesecih uporabe vmesnikov v praksi. Na podlagi iz prakse pridobljenih izkušenj ter dodatnih testiranj (tokrat se uporablja kompleksne testne modele iz prakse) se podelijo certifikati še za drugo stopnjo. Tako certificirani vmesniki naj bi zagotavljali zadovoljivo izmenjavo informacij, vendar pa raziskave kažejo, da temu ni tako. ((Dayal, 2004), (Amor, Ma, 2006), (Pazlar, Turk, 2006)).

Predstavljeno testiranje predstavlja kompromis glede cene in stroškov testiranja. Podrobnejše testiranje bi bilo glede na pomembnost pravilnega delovanja IFC vmesnikov vsekakor primernejše, a je žal izven dosega organizacije IAI. Posledično so podeljeni certifikati skladnosti (ne glede na njihovo opredelitev) za končne uporabnike lahko zelo zavajajoči.

### **2.3.1.8 Alternativni zapis IFC specifikacije v XML obliki**

#### **2.3.1.8.1 aecXML**

Avgusta 1999 je podjetje Bentley predlagalo specifikacijo XML za AEC/FM sektor imenovano aecXML. Ta način zapisa lahko smatramo kot dopolnilo in ne kot nadomestilo za obstoječe načine zapisa. Literatura (IAI-NA, 2007) predstavlja IFC zapis kot način izmenjave CAD načrtov, aecXML zapis pa kot način izražanja vsebine CAD načrtov. Tako lahko npr. prvi aecXML zapis vsebuje zahtevo po iskanju specifičnega elementa ogrevalnega sistema, drugi pa zahtevo po iskanju monterja tega izdelka.

IFC in aecXML očitno izvirata iz sorodnih potreb, a vsaka pripada svojemu koncu informacijskega kontinuuma. Motivacijo IFC zapisa predstavlja globalna podpora predstavitvi informacijskih modelov zgradb oz. njihovih delov, medtem ko pa motivacijo aecXML zapisa predstavlja potreba po predstavitvenem mehanizmu za enostaven in učinkovit mehanizem prenosa podatkov. Tehnologiji sta zato očitno bolj komplementarni kot pa konkurenčni.

aecXML si lahko najbolj nazorno predstavljamo kot organizacijski kovček, v katerem hranimo informacije, ki so potrebne za npr. transakcijo preko spleta. Informacije, ki jih prenašamo, lahko izvirajo iz CAD datotek, vendar pa informacije, ki jih prenašamo, praviloma predstavljajo le manjši del vsebine CAD datotek. Takšen pristop je soroden manipulaciji s papirjem, kjer se namesto celotne projektne dokumentacije prenaša le dele, ki so trenutno predmet zanimanja.

#### **2.3.1.8.2 ifcXML**

ifcXML predstavlja alternativo klasičnemu IFC zapisu. ifcXML predstavlja metodologijo za generacijo strukture XML dokumentov (XSD – XML Schema definition). Namesto STEP fizične datoteke imamo pri ifcXML zapisu opravka z .xml ali .ifx datotekami. Praktično gre za prevedbo EXPRESS zapisa v ifcXML XSD model po standardu ISO 10303-28 ed2 "XML representation of EXPRESS schemas and data". Preslikava med shemama je avtomatizirana, prilagajanje preslikav poteka s pomočjo konfiguracijske datoteke (preslikava med EXPRESS jezikom in XSD ni vedno deterministična, obstaja več načinov preslikav primitivov EXPRESS sheme v primitive XSD, pri čemer se njihova semantika ohranja). ifcXML shema (od različice IFC2x dalje) je sestavljena iz dveh delov (IAI, 2008):

- skupne XSD sheme za vse EXPRESS modele (prevedba glave in splošnih podatkovnih tipov);
- IFC specifične XSD sheme, ki vsebuje definicije IFC specifičnih razredov, razmerij, atributov in podatkovnih tipov.

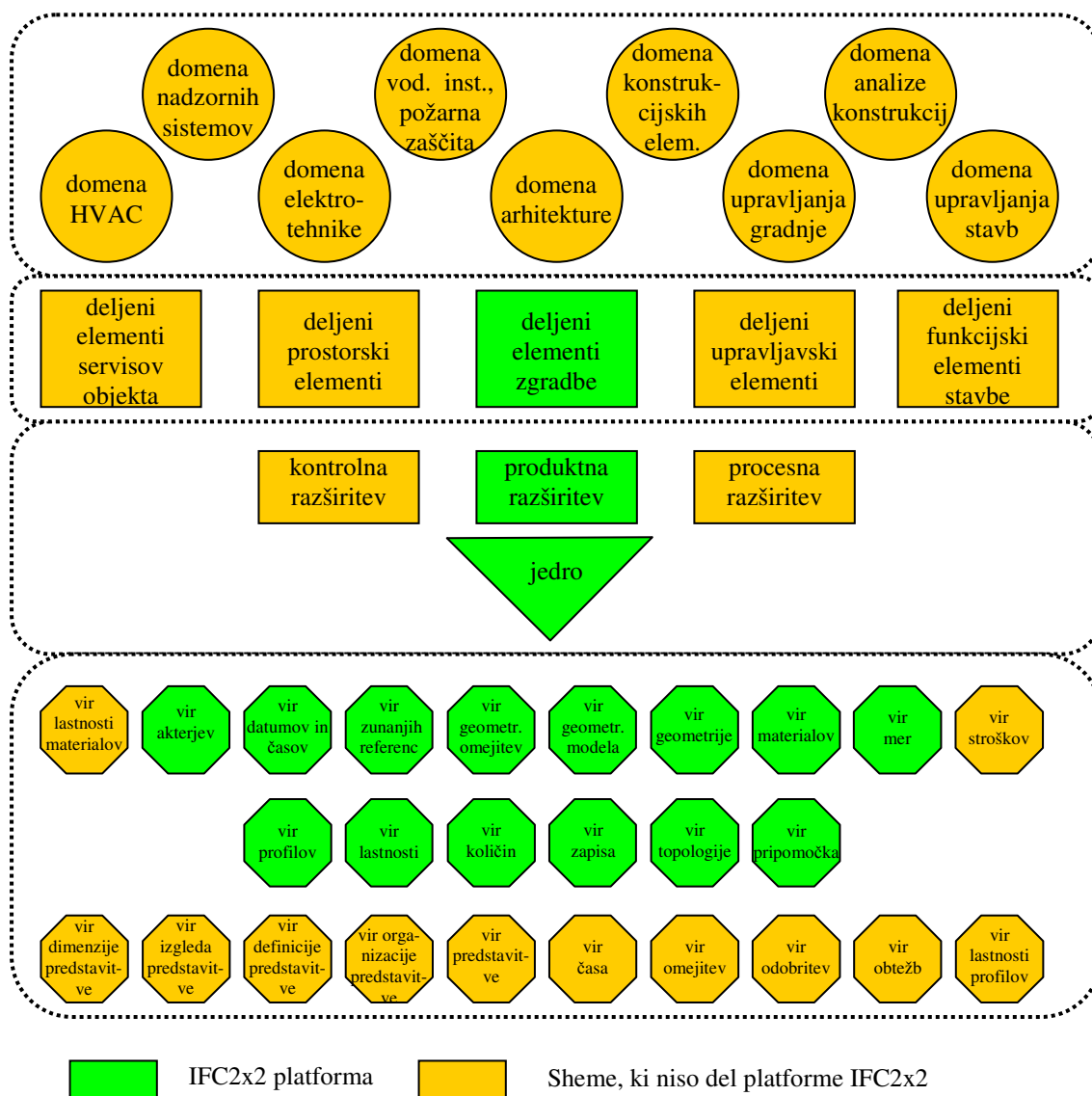
Nastanek predstavljenega zapisa je povezan s potrebo po pridobivanju informacij za grajeni informacijski model zgradbe iz ostalih sorodnih področij, kjer že obstajajo XML zapisi (npr. podatkovne zbirke, GIS, geodezija ipd.) ter za nudenje določenih informacij iz modela (npr. za potrebe e-poslovanja, vizualizacije in navidezne resničnosti ipd.). Avtorji zapisa predvidevajo, da ifcXML ne bo nadomestil IFC zapisa: pri izpeljavi ifcXML sheme iz EXPRESS sheme se namreč izgubi določen del semantike predvsem na področju pravil (WHERE pravilo), inverznih razmerij ter izpeljanih atributov (Lieblich, 2004). Poleg tega je zapis modela v XML obliki precej obsežnejši (2–10 krat) kot pa zapis v STEP fizični datoteki. Pri tem velja omeniti, da je pri večjih modelih tudi velikost STEP fizične datoteke lahko problematična – nekaj 10 do več 100 MB. Ima pa ifcXML zapis tudi nekatere prednosti (predvsem v smislu delnih izmenjav, ipd.).

Očitno sta aecXML in ifcXML podobni oziroma kar enakovredni tehnologiji. V AEC-FM sektorju danes načeloma velja dogovor, da se uporablja ifcXML, na področjih, ki jih trenutno IFC-ji še ne pokrivajo, pa se uporablja kar XML (Lieblich, 2004).

### **2.3.1.8.3 IFC in XML**

IFC predstavlja standardni način zapisa modela zgradbe, XML pa jezik za opisovanje strukturiranih podatkov ali arhitekturo za prenos podatkov in njihovo izmenjavo med različnimi programskimi orodji. IFC in aecXML specifikaciji sicer izvirata iz nasprotnih koncev informacijskega kontinuuma, vendar pa ju lahko označimo kot komplementarni tehnologiji, katerih obstoj in uporaba je nujna pri sodobnem produktnem modeliranju. Pričakovati je, da bo zapis celotnega informacijskega modela zgradb še vedno temeljil na STEP zapisu, predvsem zaradi kompaktnosti zapisa v primerjavi z XML zapisom (Nisbet & Leibich, 2007).

## 2.3.2 Zgradba modela IFC



Privzeta slika 5: Shema specifikacije IFC2x3 (IAI, 2008)

Adopted figure 5: Schema overview of IFC2x3 (IAI, 2008)

IFC specifikacija temelji na nekaj enostavnih principih, ki predstavljajo osnovne zahteve pri modeliranju (Liebich, Wix, 2000):

- zagotoviti modularno strukturo modela;

- zagotoviti ogrodje za izmenjavo informacij med strokami AEC/FM sektorja;
- olajšati vzdrževanje obstoječega in razvoj novih različic modela;
- omogočiti ponovno uporabo komponent modela;
- avtorjem programske opreme omogočiti ponovno uporabo že izdelanih komponent;
- skrbeti za združljivost vnaprej (v prihodnosti) med različicami IFC modela.

Model zgradb IFC je v osnovi zgrajen iz shem, ki opisujejo sorodne informacije. Sheme, združene v sloje, lahko povežemo le v smeri gravitacije (privzeta slika 5).

IFC specifikacijo sestavljajo štiri sloji:

- sloj virov: vire, vključene v najnižji sloj arhitekture modela IFC, lahko označimo kot generične koncepte oziroma objekte, neodvisne od aplikacije oz. domene. Geometrija je recimo široko uporabljen vir, njena specifikacija pa je neodvisna od modela. Vendar pa mora biti objekt znotraj domene definiran, preden geometrija lahko obstaja. IFC specifikacija sicer uporablja svojo konvencijo o poimenovanju (Lieblich, Wix, 2000), vendar pa so informacijski modeli v sloju virov pogosto povzeti po definicijah, ki so bile izdelane v okviru standarda STEP. Posamezni pomensko sorodni gradniki so združeni (npr. geometrija v vir geometrije ipd.), navzven pa so prepoznavni po oznaki vir – »Resource« (npr. vir geometrije – »Geometry Resource«);
- osrednji sloj predstavlja osnovno strukturo modela IFC in definira splošne koncepte, ki so skupni vsem delom modela. S tem je zagotovljena ustrezna mera združljivosti, koncepte pa je v višjih slojih moč natančneje definirati. Osrednji sloj tvori:
  - jedro: v jedru so zbrani osnovni koncepti modela IFC, njegova struktura in dekompozicija. Jedro si lahko predstavljamo kot šablono, znotraj katere se razvija model. Konstrukti v jedru predstavljajo najbolj generične koncepte, ki se specializirajo v preostalih slojih;
  - razširitve jedra: vsaka razširitev predstavlja specializacijo splošnih konstruktov, definiranih v jedru. Konstrukti v jedru so zelo splošni in bi jih bilo moč uporabiti tudi v drugih panogah (ne le v AEC/FM), z uporabo razširitev jedra pa se omejimo na AEC/FM sektor. Sheme, ki pripadajo



razširitvam jedra, prepoznamo po oznaki razširitev – »Extension« (npr. produktna razširitev – »Product Extension«);

- v sloju skladnosti (interoperabilnosti) so definirani koncepti (ali razredi), ki so skupni vsaj dvema domenskima modeloma, z vsebovano shemo deljenih elementov pa je omogočena interoperabilnost med različnimi domenskimi modeli. Predstavljeni sloj označuje beseda elementi – »Elements« (npr. deljeni elementi zgradb – »Shared Building Elements«);
- domenski sloj zagotavlja natančen opis modela, kot je zahtevan z AEC-FM domenskimi procesom ter tipom aplikacije. Primeri domenskih modelov so domena arhitekture, HVAC, analize konstrukcij, domena upravljanja stavb ipd. Sloj označuje termin domena – »Domain« (npr. domena arhitekture – »Architecture Domain«).

Med posameznimi sloji velja princip gravitacije:

- v sloju virov se lahko pojavljajo reference le med posameznimi gradniki sloja virov. Vendar pa le-te niso najbolj zaželeno, saj se tako izgublja neodvisnost posameznih virov;
- v jedrnem sloju se lahko pojavljajo reference znotraj jedra, vendar pa je tudi v jedru potrebno upoštevati princip gravitacije: razredi jedra ne morejo referencirati razredov, ki se nahajajo v razširitvah jedra. Pri referenciranju razredov, ki se nahajajo v sloju virov pa ni ovir;
- razredi iz sloja interoperabilnosti lahko referencirajo razrede iz jedra (in njegovih razširitev) ter iz sloja virov, ne pa iz domenskega sloja;
- razredi domenskega sloja lahko referencirajo katerikoli razred v vseh treh nižje ležečih slojih.

Pri predstavitvi arhitekture modela zgradb IFC je nujno potrebno razlikovati med izdajo standarda (IFC2x Edition 2 Final) ter platformo (IFC2x Edition 2 Platform). Izdaja standarda namreč vsebuje vse sheme trenutne različice, platforma pa zaradi zagotavljanja kompatibilnosti v prihodnje vsebuje le sheme, ki bodo kljub morebitnim novim različicam IFC2x ostale nespremenjene. To pomeni, da platforma IFC2x Edition 2 Platform popolnoma temelji na IFC2x Platform, kar pomeni:

- da vsebuje vse entitete, attribute in relacije ter tipe, ki so bili definirani že v IFC2x platformi;
- nekatere entitete iz platforme 2x niso priporočene za uporabo;
- vsebuje dodatne entitete, attribute in razmerja (le-ta pripadajo novim entitetam);
- obstoječe entitete so nespremenjene – enaki atributi v enakem zaporedju. Imena obstoječih entitet atributov in tipov ostanejo nespremenjena.

Posledično lahko vse IFC datoteke, ustvarjene s platformo IFC2x beremo z IFC2x Edition 2 kompatibilno programsko opremo.

### **2.3.2.1 Ključne strukture modela zgradb IFC 2x3**

V nadaljevanju bodo podrobneje predstavljene ključne strukture modela zgradb, ki so nujno potrebne za razumevanje zgradbe modela IFC ter preslikav, na katere se osredotoča predstavljena naloga. Podrobnejše informacije o zgradbi modela je moč najti v tehničnem (Liebich, Wix, 2000) ter v implementacijskem (Liebich, 2000) vodniku ter v IFC specifikaciji (IAI, 2008).

#### **2.3.2.1.1 Jedro modela**

Izhodišče vseh končnih entitet, ki jih lahko instanciramo v IFC modelu, predstavlja korenska entiteta IfcRoot. Njeni atributi predstavljajo osnovne koncepte za avtorstvo (avtor, čas nastanka in avtorstvo zadnje spremembe), identifikacijo (GUID oznaka) ter opis entitet (opcijnska atributa ime in opis).

##### **2.3.2.1.1.1 Prvi nivo specializacije**

Iz IfcRoot entitete izhajajo trije osnovni tipi, ki tvorijo prvi nivo specializacije modela zgradb IFC (privzeta slika 6):

- definirani objekti (IfcObjectDefinition): generalizacija kateregakoli semantičnega primitiva ali procesa. Kot definirani objekti lahko nastopajo objekti ali tipski objekti;
- definirane lastnosti (IfcPropertyDefinition): generalizacija karakteristik, ki jih je možno pripisati objektom;

- razmerja (IfcRelationship): generalizacija vseh razmerij med objekti v IFC modelu. Razmerja v IFC modelu se namreč objektificirajo.

#### 2.3.2.1.1.2 Drugi nivo specializacije

Druga stopnja podrobnosti predstavlja podrobnejšo delitev entitet, definiranih v prvi stopnji. Abstraktne definirane objekte sestavljajo:

- objekti (IfcObject), ki predstavljajo generalizacijo kateregakoli semantičnega objekta (fizični objekti (stene, vrata, okna), procesi (delavne naloge), viri (delavna sila) ipd.). Objekti se nadalje delijo na:
  - produkte (IfcProduct): le-ti opisujejo fizične objekte (proizvedene, dobavljene, ustvarjene). Kot edini podskupini objektov jim je moč določiti način predstavitve ter lego v izbranem koordinatnem prostoru. Poseben podtip produkta predstavlja proksi (IfcProxy), ki opisuje objekte, ki sicer semantično niso vključeni v IFC model;
  - procese (IfcProcess): opisujejo aktivnosti, ki se odvijajo na projektu z nekim namenom. Procesni so razvrščeni v časovno zaporedje;
  - kontrole (IfcControl): opisujejo koncepte, ki nadzirajo oz. omejujejo druge objekte. Lahko se jih razume kot navodila, specifikacije, regulacije, omejitve oz. zahteve, ki jih lahko pripiše objektom;
  - vire (IfcResource): viri predstavljajo koncepte, ki opisujejo rabo objektov v procesu;
  - akterje (IfcActor): predstavljajo človeške vire, ki so vključeni v življenjski cikel projekta;
  - projekt (IfcProject): projekt opisuje aktivnosti, ki vodijo k opisu produkta v njegovem življenjskem ciklu. V modelu lahko nastopa samo ena instanca entitete IfcProject;
  - skupine (IfcGroup): poljubna zbirka objektov, ki se lahko obravnava kot funkcijska enota znotraj modela zgradb IFC;

- tipski objekti (`IfcTypeObject`), ki opisujejo specifične informacije o tipu. Tipski objekti predstavljajo zbirnike, ki združujejo običajno skupno nastopajoči sorodni nabor lastnosti (v vseh predhodnih različicah IFC specifikacije so bili zato tipski objekti v prvem nivoju delitve uvrščeni med lastnosti). Teoretično je sicer možno objektom pripisati vsako lastnost posebej, vendar pa se priporoča uporaba nabora lastnosti. Instance tipskih objektov se objektom pripenja z `IfcRelDefinesByType`. Če gre za opis nabora lastnosti na nivoju produkta (oblika produkta), potem se uporablja entiteta `IfcTypeProduct`.

V razredu razmerij (`IfcRelationship`) na drugi stopnji podrobnosti nastopa pet vrst izpeljanih razmerij:

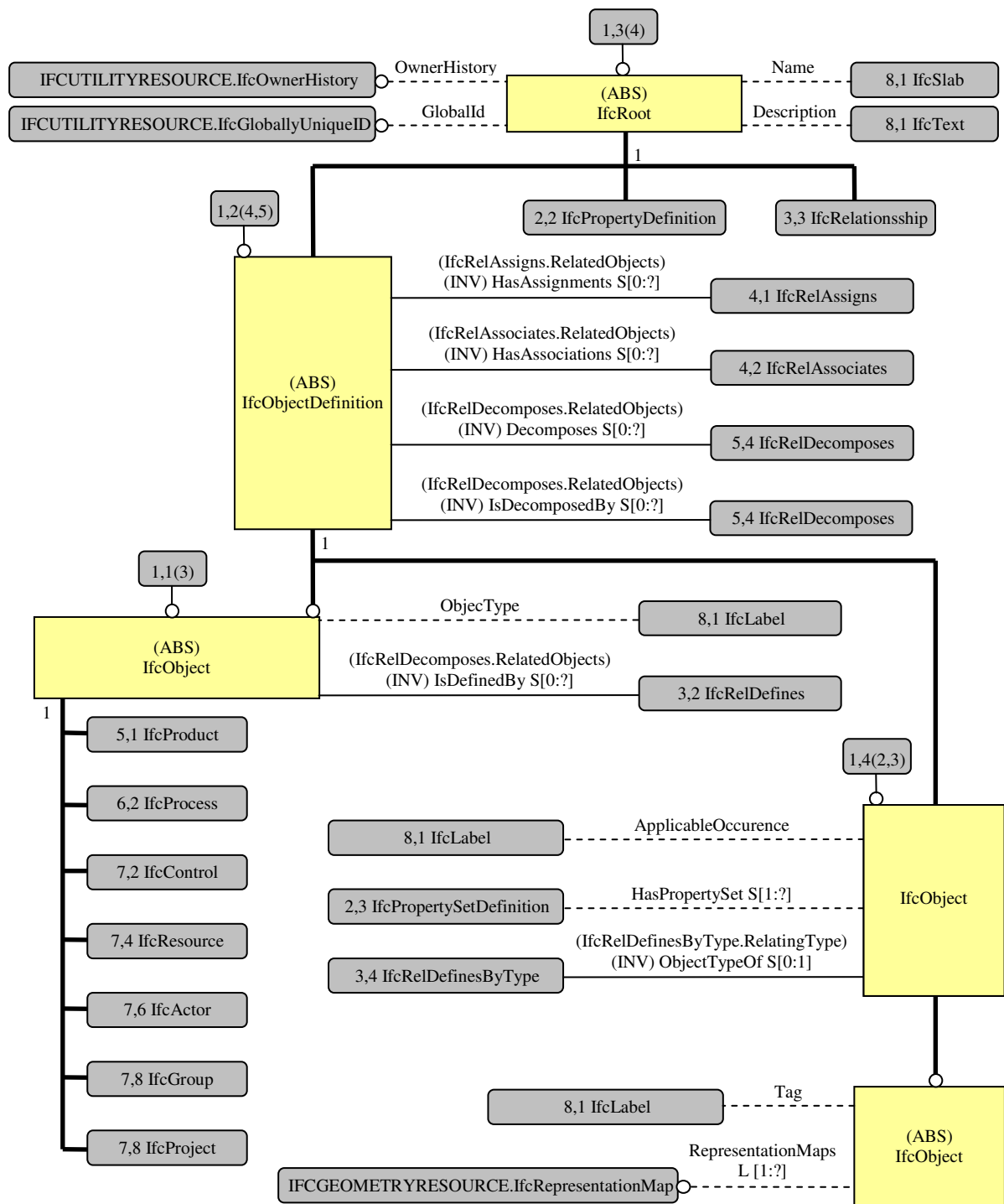
- Koncept dodeljevanja (`IfcRelAssigns`) predstavlja posplošitev povezave med instancami objektov in njihovih podtipov. Definirane so povezave na produkte (`IfcRelAssignsToProducts`), procese (`IfcRelAssignsToProcesses`), kontrole (`IfcRelAssignsToControls`), vire (`IfcRelAssignsToResources`), akterje (`IfcRelAssignsToActors`) ter skupine (`IfcRelAssignsToGroups`).
- Koncept asociacije (`IfcRelAssociates`) predstavlja povezavo na objekte (oz. lastnosti) iz sloja virov. Preko objektov iz sloja virov so možne tudi povezave na zunanje vire informacij – na klasifikacije (`IfcRelAssociatesClassification`), knjižnice (`IfcRelAssociatesLibrary`) in dokumente (`IfcRelAssociatesDocument`).
- Koncept dekompozicije (`IfcRelDecomposes`) določa povezave/razmerja med objekti, kjer nastopa nadrejeni (parent) objekt, ki na določenem nivoju dekompozicije definira celoto ter eden ali več podrejenih objektov (child), ki določajo posamezne dele. Obstaja dvoje razmerij dekompozicije: agregacija – `IfcRelAggregates` (nadrejeni in podrejeni objekt sta/so instance različnih razredov) in gnezdenje – `IfcRelNests` (nadrejeni in podrejeni tip sta/so instance istega razreda).
- Koncept definicije (`IfcRelDefines`) ureja razmerja med individualnimi objekti, tipskimi objekti ter naborom lastnosti. Definirani sta:
  - definicija po lastnostih – `IfcRelDefinesByProperties` (nabor lastnosti se lahko poveže s poljubnim objektom, vsebovanim v razredu `IfcObject`);

- definicija po tipu – `IfcRelDefinesByType` (poljubni objektni tip se lahko poveže s poljubnim objektom, vsebovanim v razredu `IfcObject`).
- S konceptom povezave (`IfcRelConnects`) medsebojno povežemo poljubne procese na podlagi poljubno izbrane (izbranih) skupne (skupnih) lastnosti. Primer takšne povezave predstavlja koncept zaporedja (`IfcRelSequence`), ki opisuje zaporedje procesov.

V drugi stopnji podrobnosti opisa lastnosti nastopa ena sama abstraktna entiteta (`IfcPropertySetDefinition`), ki omogoča širitev modela. Z njo lahko definiramo: 1.) dodatne lastnosti, ki se pripnejo poljubnim instancam, 2.) dodatne lastnosti, ki se dodajo samo izbranim instancam, 3.) dodatne lastnosti, ki jih z namenom (nestandardne) širitve modela določijo končni uporabniki. Dodatno definirane deljene in razširljive nabore lastnosti se s pomočjo že predstavljenih razmerij pripne k objektom – entiteta `IfcRelDefinesByProperties` povezuje entiteto `IfcObject` in `IfcPropertySetDefinition`. Takšen pristop eliminira reference v objektih (oz. njegovih v atributih) ter omogoča neodvisen pripis lastnosti le izbranim objektom.

Pri konkretni implementaciji se lastnosti definirajo z entitetami `IfcPropertySet` (podobjekt `IfcPropertySetDefinition`). Nabore se ločuje med seboj z imenom (`name`), dodan pa je lahko tudi berljiv opis. Dodatno definirane lastnosti se delijo v statično definirane (definirane znotraj IFC modela) ter dinamično razširljive lastnosti, ki predstavljajo eksterni opis karakteristik (lahko so predstavljene z enostavno vrednostjo, naštevno vrednostjo, seznamom vrednosti, območjem vrednosti, z referenco ali kot kompleksna vrednost). Tako se lahko npr. sklicujemo na zunanjo bazo informacij (npr. lastnost posameznih elementov informacijskih modelov predstavljajo cene teh elementov, ki so objavljene na spletu).

Vsak nabor lastnosti mora vsebovati vsaj eno lastnost, to je vsaj eno instanco abstraktnega razreda `IfcProperty`: enostavno vrednost (`IfcPropertySingleValue`), števno vrednost (`IfcPropertyEnumeratedValue`), omejeno vrednost (`IfcPropertyBoundedValue`), območje – rang vrednosti (`IfcPropertyTableValue`), objektno referenco (`IfcPropertyReferenceValue`) ali kompleksno lastnost (`IfcPropertyComplexProperty`).



Privzeta slika 6: Entitete v IFC2x3 (druga stopnja podrobnosti) (IAI, 2008)

Adopted figure 6: Entities in IFC2x3 (second level of details) (IAI, 2008)

Podobno kot ostale entitete se tudi nabore lastnosti v fizične datoteke vključuje skladno z ISO 10303-del 21 (ISO 10303-21). Vendar pa se je pri razvoju specifikacije izkazalo, da je za prenos omejenega nabora lastnosti primernejši zapis XML (Extensible Markup Language). Pri takšnem načinu je potrebno definirati:

- pravila za strukturiranje dokumenta (DTD – Document Type Definition), in sicer za definicijo nabora lastnosti (Property Set Definition Language);
- pravila za strukturiranje dokumenta za dostavo vsebin (Property Set Markup Language);
- slogovno datoteko (XSL – Style Sheet), v kateri so določeni slogi in oblikovanje nabora lastnosti.

Nabori lastnosti, ki se jih vključuje v informacijski model zgradb, so običajno shranjeni v podatkovni bazi. Razred je v bazi predstavljen s preglednico, atributi pa predstavljajo polja v preglednici. Razmerja med razredi so predstavljena s povezavami med preglednicami. Predstavljeni način izmenjave je še posebej primeren za izmenjavo lastnosti, ki so definirane zunaj modela (npr. v zunanjih bazah, knjižnicah ipd.). V primeru knjižnic tehnično obstajata dva načina vključitve lastnosti v model, in sicer neposredna vključitev v model ter referenciranje zunanje knjižnice.

Poleg navajanja dodatnih lastnosti se lahko pri implementaciji IFC modela vključi tudi nove entitete, ki v IFC specifikaciji sploh niso definirane. Predstavljena možnost je namenjena širitvam modela, ki v določeni različici specifikacije niso definirane. V ta namen je predvidena entiteta *IfcProxy* (pripada razredu *IfcProduct*), ki je locirana v jedru modela in je ena redkih neabstraktnih entitet jedra. Pri praktični uporabi entitete *IfcProxy* se je potrebno zavedati težav, ki lahko nastopijo pri opisanih razširitvah (skladnost posameznih vmesnikov z osnovno ali tudi razširjeno shemo).

#### **2.3.2.1.2 Geometrijska predstavitev produktov**

Vsaka izmed instanciranih entitet, izpeljanih iz nadtipa *IfcProduct* ima lahko eno ali več geometrijskih predstavitev. Geometrijske predstavitve so določene v shemah, ki so uvrščene v sloj virov, in sicer:

- *IfcGeometryResource* – osnovni geometrijski elementi,

- IfcTopologyResource – osnovni topološki elementi,
- IfcGeometricModelResource – geometrijska predstavitev objektov z rabo osnovnih geometrijskih in topoloških elementov,
- IfcRepresentationResource – koncepti večpredstavnosti objektov,
- IfcGeometryConstraintResource – koncepti postavitve objektov,
- IfcProfileResource – definicija standardnih profilov (oz. prečnih prerezov), ki se uporabljajo pri geometrijski predstavitvi objektov.

Navedeni načini predstavitve so večinoma povzeti po standardu ISO 10303: prvi trije viri po delu 42 (Integrirani generični viri: geometrijske in topološke predstavitve), četrti vir pa po delu 43 (Integrirani generični viri – strukture prikaza). Povzeti deli ISO 10303 standarda so nekoliko prilagojeni zahtevam pri modeliranju zgradb, in sicer so izpuščeni nekateri kompleksnejši prikazi geometrije, povsem pa je prilagojeno poimenovanje. Zadnja dva navedena vira predstavljata edinstveni del IFC specifikacije, za katere ni moč najti alternative v standardu ISO 10303.

Vsak objekt z geometrijsko predstavitvijo mora imeti določena dva atributa:

- postavitev (IfcObjectPlacement):
  - absolutna – relativno na svetovni koordinatni sistem,
  - relativna – relativno na postavitev nekega drugega objekta,
  - omejena – relativno na mrežne osi.

Privzeto postavitev predstavlja relativna postavitev, ki jo je moč prepoznati po atributu PlacementRelTo. Če atribut ni definiran, potem gre za absolutno postavitev;

- predstavitev (IfcProductRepresentation oz. podtip IfcProductDefinitionShape) kot vsebovalnik potencialnih večkratnih geometrijskih in topoloških predstavitev določa, kako bo izbrani objekt (npr. stena) predstavljen (npr. kot črta, ravnina, telo).

Vse predstavitve morajo biti definirane v istem koordinatnem sistemu. Predstavitve načeloma delimo v tri skupine:

- neposredne geometrijske predstavitve,



- neposredne topološke predstavitve (se ne uporabljajo neposredno, temveč posredno, preko geometrijskih predstavitev),
- preslikane predstavitve, ki izhajajo iz kartezijskih transformacij.

Obstaja več načinov neposredne geometrijske predstavitve teles. V grobem jih lahko razdelimo v dve skupini (Rooney, 1987):

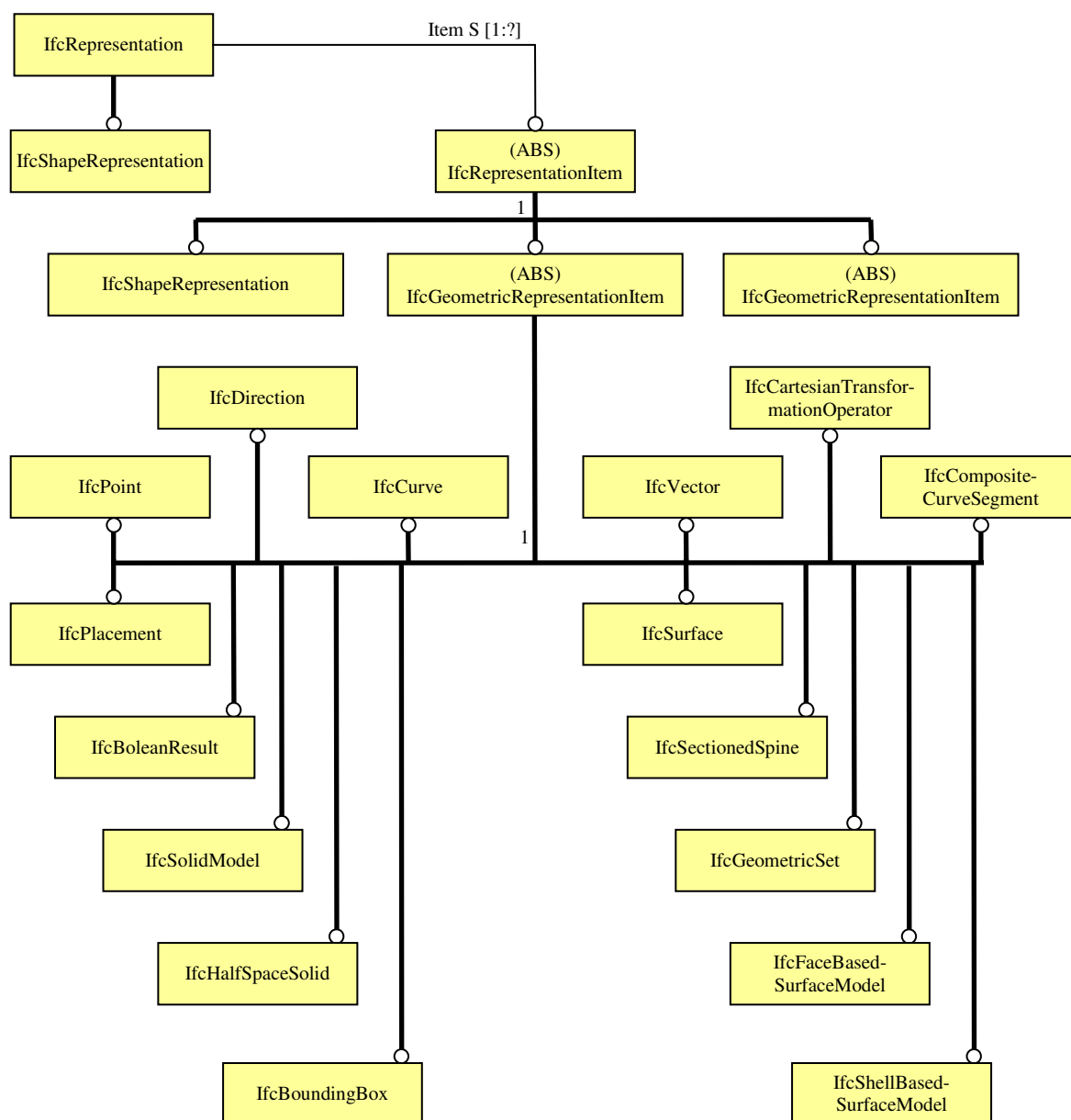
- **deklarativna predstavitve** – pri tej predstavitvi deklariramo trenutno stanje telesa s pomočjo koordinat točk, ki pripadajo povezanim vozliščem, robovom ter ploskvam. Najbolj pogosto uporabljena deklarativna predstavitve je mejna (Brep – Boundary Representation);
- **proceduralna predstavitve** – poleg osnovnih gradnikov se shrani tudi procedura nastanka vsakega telesa. Takšna predstavitve obsega volumska telesa (CSG – Constructive Solid Geometry), izvlečena ter rotirana telesa. V prvem primeru lahko s pomočjo Boolovih operacij iz primitivov sestavljamo poljubna telesa, medtem ko pa pri drugih dveh lahko ustvarimo le osnosimetrične gradnike (vrtenine) ter gradnike, katerih prerez se v izvlečeni smeri ne spreminja (izvlečena telesa).

Podobna, a nekoliko podrobnejša delitev geometrijske predstavitve teles, je prisotna tudi v IFC specifikaciji (privzeti sliki 7, 8):

- **2D krivulje** (Curve2D) so namenjene črtni predstavitvi objektov. Najbolj splošna predstavitve obsega parametrični opis krivulje, v praksi pa se uporabljajo za predstavitve osi elementov. 2D krivulje se opisuje z entitetami `IfcBoundedCurve` (krivulja končne dolžine z znano začetno in končno točko), `IfcCompositeCurve` (krivulja sestavljena iz več segmentov), `IfcLine` (neukrivljena krivulja s konstantno smerjo tangente) ter `IfcPolyline` (ukrivljena krivulja iz  $n-1$  segmentov, ki jo definira  $n$  točk).
- **Geometrijski nabor** (GeometricSet) se uporablja za geometrijsko predstavitve, ki temelji na naboru točk, krivulj, črt in površin, pri čemer topološka struktura ni na voljo. Predstavljeni način predstavitve se lahko uporablja za 2D in 3D predstavitve, pri čemer morajo biti posamezne enote geometrijske predstavitve 2D ali 3D primitivi in

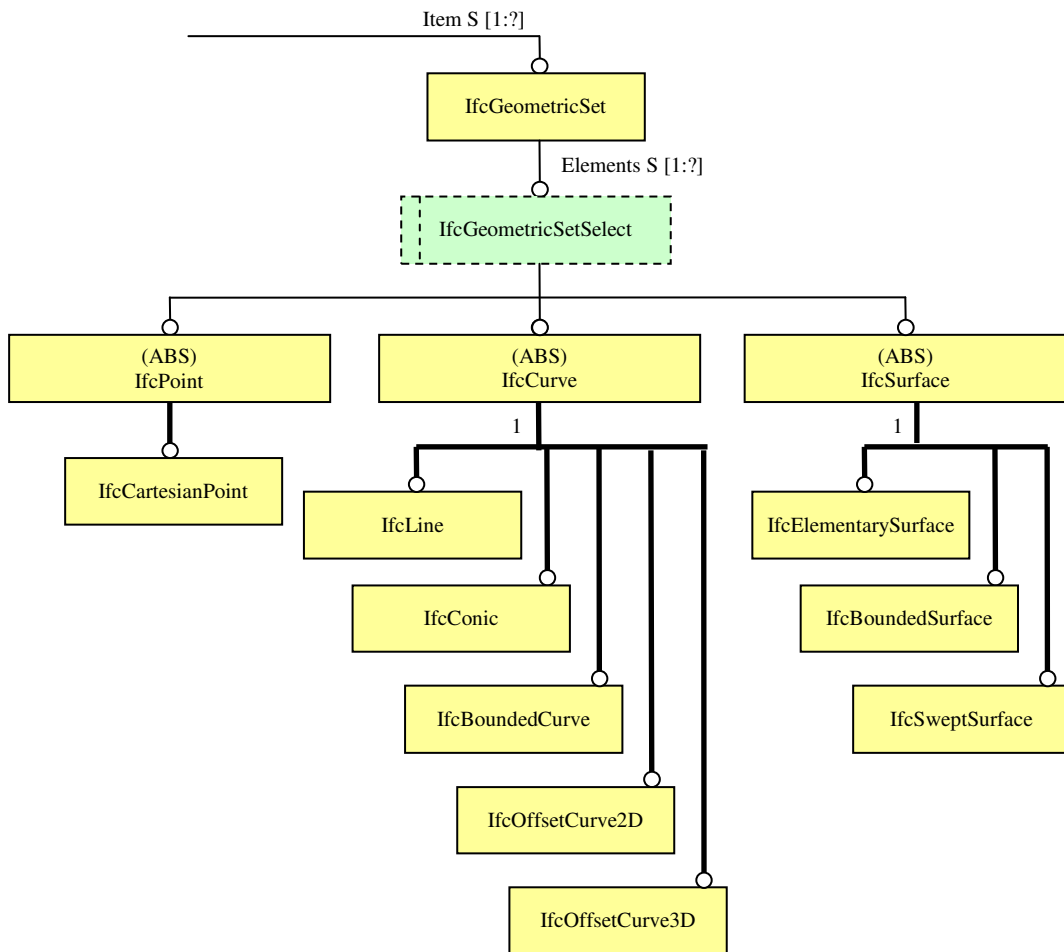
ne kombinacija obeh. V praksi se geometrijski nabor uporablja za predstavitev pohištva, opreme, tlorise prostorov ipd.

Geometrijski nabor krivulj (GeometricCurveSet) predstavlja podskupino geometrijskega nabora, pri katerem se za opis uporabljajo le točke ter krivulje (2D ali 3D). Uporaba geometrijskega nabora krivulj je v trenutni različici predvidena za vsak 2D prikaz elementov, katerih predstavitev temelji na točkah in črtah.



Privzeta slika 7: Koncept predstavitve oblike (IAI, 2008)

Adopted figure 7: Concept of shape representation (IAI, 2008)



Privzeta slika 8: Entitete geometrijske predstavitve (IAI, 2008)

Adopted figure 8: Geometric representation entities (IAI, 2008)

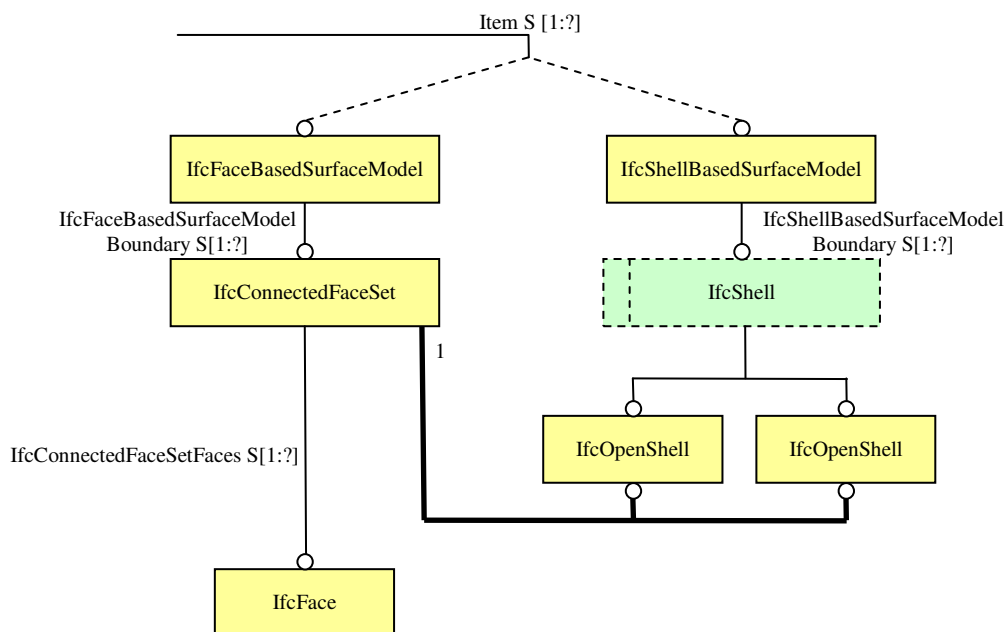
- **Površinski modeli** (SurfaceModel) predstavljajo 3D predstavitve, temelječe na ploskvah, katerih osnovo predstavlja sklenjeno črtovje (privzeta slika 9). Pri površinskih modelih je topološki opis poznan. Predstavljeni modeli se največkrat uporabljajo za predstavitev eksplicitne 3D forme objekta, kjer informacije o volumnu niso potrebne oz. zahtevane, npr: kovinske plošče, predstavitev terena, predstavitev distribucije (tok tekočin), ipd.

Enega izmed pomembnejših gradnikov površinskih modelov predstavlja (orientirana) ploskev »face«. V splošnem ploskev predstavlja orientiran in povezan gradnik v prostoru  $R_n$ , v tridimenzionalnem prostoru pa si ga predstavljamo kot košček površine,

ki jo obkrožajo zanke. IFC dokumentacija ploskev označuje kot topološko entiteto dimenzije 2. Posamezno ploskev lahko opisuje več zank (npr. če so prisotne tudi odprtine), ki pa se ne smejo sekati. Posamezne zanke ne smejo vsebovati notranjih zank.

Zunanja zanka predstavlja meje površine, ki ima določeno normalo (**n**) ter tangento na zanko (**t**). Pri tej definiciji normale in tangente je vektorski produkt  $\mathbf{n} \times \mathbf{t}$  vedno usmerjen v notranjost orientirane ploskve. Očitno sta notranjost in zunanost orientirane ploskve odvisna od smeri zanke oz. od smeri tangente. Posledično je pri vsaki orientirani ploskvi potrebno podati orientacijo. Slednja se podaja z logičnima oznakama »pravilno« (desnosučna zanka) in »napačno«.

Površinske modele lahko opišemo tudi z lupinami (IfcShell). Le-te so lahko odprte ali zaprte, v vsakem primeru pa so sestavljene iz nabora ploskev.



Privzeta slika 9: Predstavitev oblike s pomočjo površin (IAI, 2008)

Adopted figure 9: Surface model shape representations (IAI, 2008)

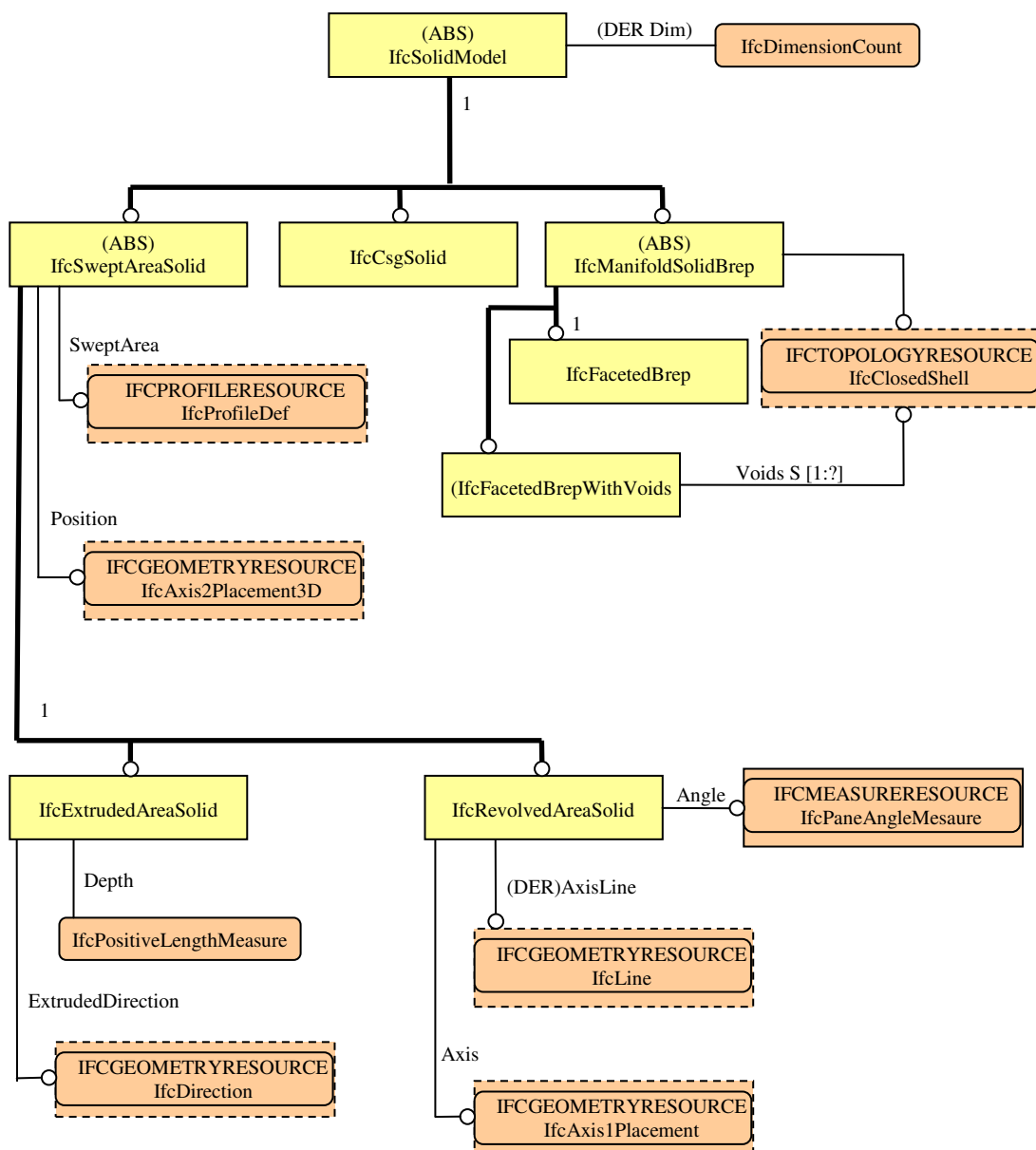
- **Trdni modeli** predstavljajo najbolj popolno tridimenzionalno predstavitev geometrije objektov (privzeta slika 10). Poglavitno prednost predstavlja nedvoumna identifikacija

poljubne točke v prostoru. Le-te lahko ležijo znotraj ali zunaj telesa. Trdna telesa lahko glede na način predstavitve podrobneje razdelimo na:

- **robno predstavitev** (Brep) – pri tej predstavitvi je geometrija telesa predstavljena z orientiranimi ploskvami z ali brez odprtin. S tem je podana tudi topologija, definirana pa je tudi notranjost (oz. volumen) trdnega modela. Glede na definicijo normale orientirane ploskve, podane v prejšnjem poglavju, velja, da so normale usmerjenih ploskev, ki sestavljajo trdno telo, orientirane stran od trdnega telesa. Brep predstavitev zahteva popolno definicijo robov modelirane zaprte oblike: vsi robovi predstavljene oblike tako pripadajo dvema orientiranima ploskvama.

Robna predstavitev je primerna za predstavitev večine oblik v gradbeništvu, in sicer tudi v primerih, kjer se potrebuje podatek o volumnu;

- **širjena telesa** (SweptSolid) omejujejo predstavitev trdnih teles le na telesa, dobljena z linearnim izvlečenjem ter na vrtnine;
- **CSG** (Constructive Solid Geometry) trdna telesa predstavljajo rezultat Boolovih operacij (unija, presek, razlika) med primitivnimi trdnimi telesi in polravninami;
- **mejna škatla** (BoundingBox) predstavlja poenostavljeno predstavitev 3D elementov s kvadrom;
- **preslikana predstavitev** (MappedRepresentation) predstavlja mehanizem za večkratno uporabo posameznih gradnikov. Koncept preslikanih predstavitev je identičen rabi blokov v CAD programski opremi: blok vsebuje geometrijske elemente, njihova lega pa je določena v lokalnem (blokovnem) koordinatnem sistemu. S pomočjo sklicevanja, ki vključuje tudi kartezijski transformacijski operator (sprememba velikosti) pa posamezen blok lahko poljubnokrat vstavimo v načrtovani model. V IFC specifikaciji je koncept definicije bloka predstavljen z entiteto IfcRepresentationMap, instanco posameznega bloka pa določa IfcMappedItem.



Privzeta slika 10: Predstavitev oblike s pomočjo trdnih teles (IAI, 2008)

Adopted figure 10: Solid model representations (IAI, 2008)

### 2.3.2.1.3 Enote in materiali

Enote so definirane v »viru mer«, ki je povzet po ISO 10303-del 42. Enote so definirane kot:

- osnovne SI enote s predponami (npr. m, mm);
- enote, dobljene s pretvorbami iz osnovnih SI enot (npr. cola);

- izpeljane enote, pridobljene s kombinacijo osnovnih enot (npr. kg/m<sup>2</sup>);
- kontekstno pogojene enote (enote, ki jih ni mogoče neposredno izpeljati iz osnovnih enot).

Obstajata dva načina rabe enot:

- globalne enote določa instanca entitete `IfcUnitAssignment`;
- posamezni instanci entitete se lahko pripiše specifične enote (`IfcMeasureWithUnit`).

Instancam vseh podtipov entitete `IfcElement` (z izjemo `IfcOpeningElement`) lahko pripišemo inverzno razmerje, ki kaže na `IfcRelAssociatesMaterial`. Le-ta povezuje element in material, ki je lahko:

- `IfcMaterial` (en material),
- `IfcMaterialList` (več materialov, natančna struktura ni znana),
- `IfcMaterialLayerSetUsage` (več materialov, natančna struktura je poznana).

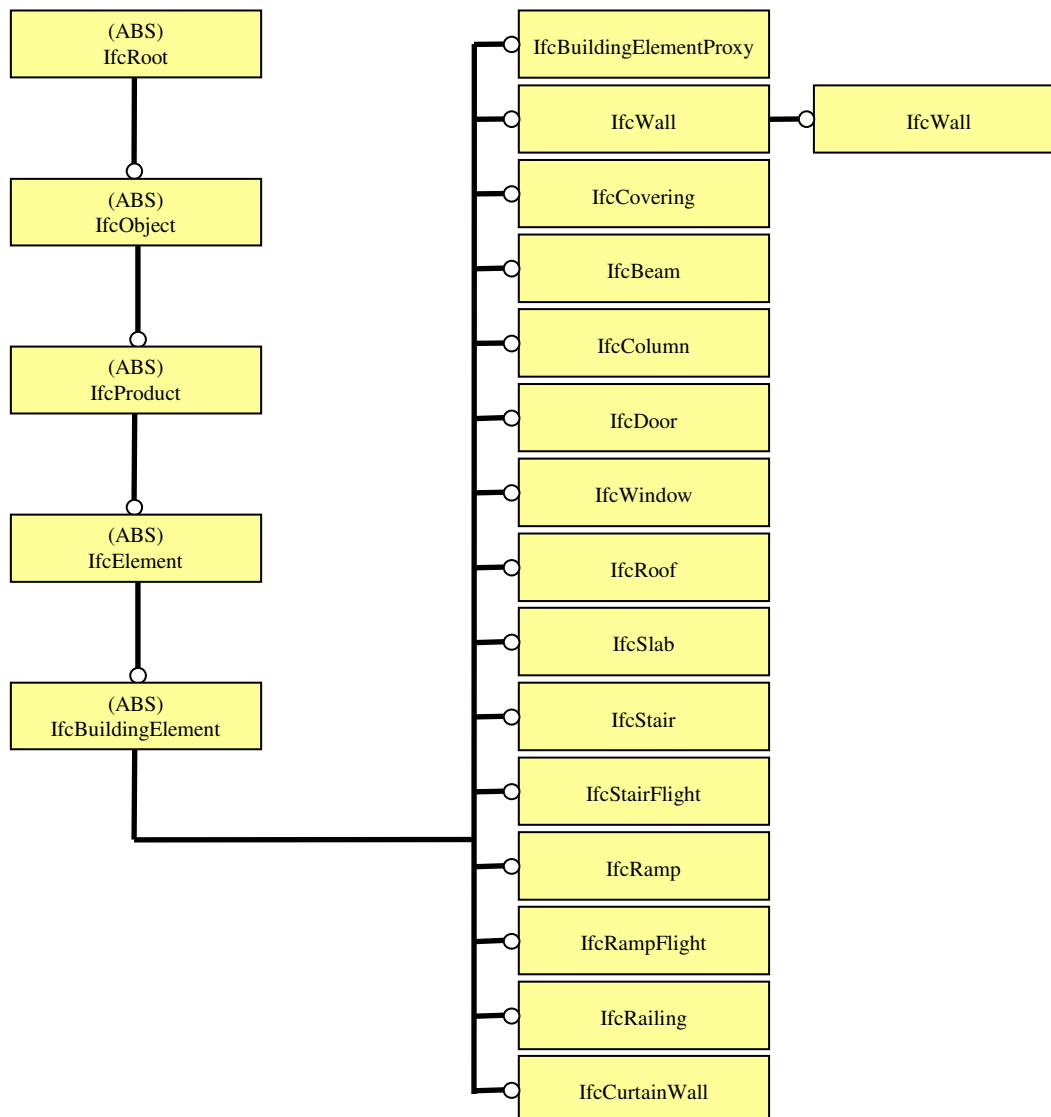
#### **2.3.2.1.4 Struktura prostorskih elementov**

Struktura prostorskih elementov (`IfcSpatialStructureElement`) predstavlja hierarhično delitev modela v manjše obvladljivejše dele:

- Projekt (`IfcProject`). Obvezna entiteta – lahko je le ena sama.
- Gradbišče (`IfcSite`). Neobvezna entiteta.
- Zgradba (`IfcBuilding`). Obvezna entiteta.
- Etaža (`IfcBuildingStorey`). Obvezna entiteta.
- Prostor (`IfcSpace`). Neobvezna entiteta.

#### **2.3.2.1.5 Elementi zgradb**

Razred elementov zgradb (`IfcBuildingElement`) predstavlja abstraktni opis primitivov arhitekturnega načrtovanja (privzeta slika 11).



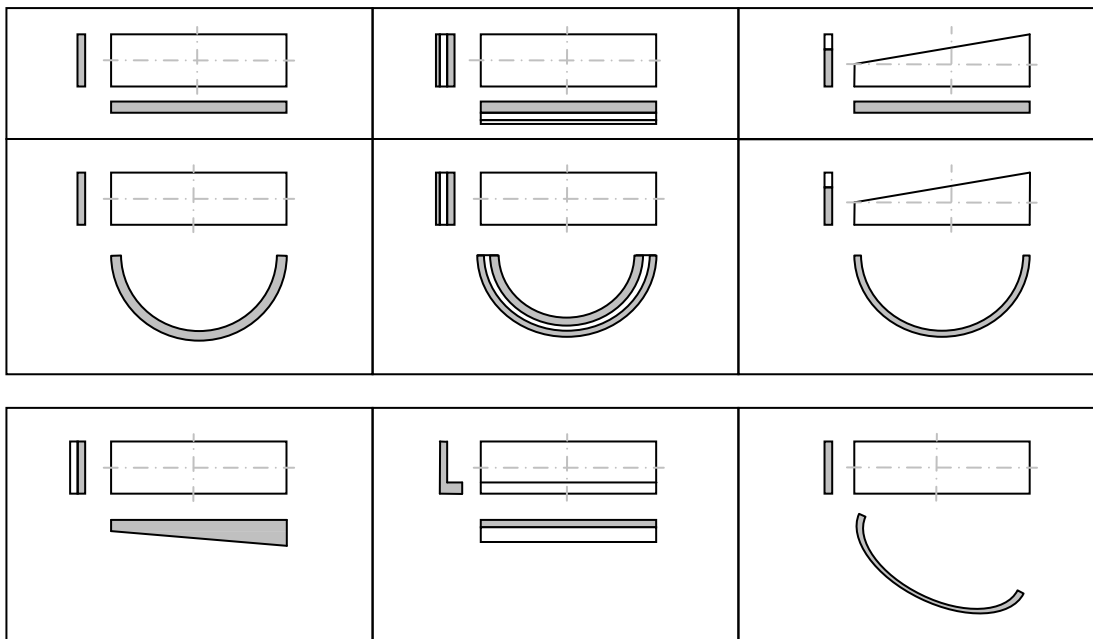
Privzeta slika 11: Hierarhija elementov zgradb (IAI, 2008)

Adopted figure 11: Hierarchy of building elements (IAI, 2008)

#### 2.3.2.1.5.1 Stene (IfcWall)

Stenam pripadajo naslednje enote funkcionalnosti (Units of functionality – UoF): postavitev, umestitev, meje v prostoru, povezave in odprtine. Dodatne enote funkcionalnosti, določene na nivoju stene, predstavljata še večpredstavnost oblike (privzeta slika 12) ter material (slednji velja le za standardne stene).





Privzeta slika 12: Primeri standardnih (zgoraj) in specifičnih sten (spodaj) (IAI, 2008)

Adopted figure 12: Examples of standard (top) and specific walls (bottom) (IAI, 2008)

Večpredstavnost oblike standardnih sten predstavljata vsaj dve predstavitvi oblike instance `IfcWallStandardCase`:

- prva predstavlja os stene (»Axis« kot identifikator stene, »Curve2D« kot način predstavitve ter »IfcTrimmedCurve« ali »IfcPolyline« kot predstavitev),
- druga predstavlja telo stene (»Body« kot identifikator stene, »SweptSolid« ali »Clipping« kot način predstavitve ter »IfcExtrudedAreaSolid« kot predstavitev). Smer izvlečenja je vedno pravokotna na os stene, izvlečena telesa pa je z ravninami možno tudi obrezati (le zgoraj in spodaj). Geometrija sten s spremenljivo višino je določena z Boolovimi operacijami, kjer prvi operand predstavlja izvlečeno telo (ali rezultat predhodnih Boolovih operacij), drugi operand pa polprostor (`IfcHalfSpaceSolid`).

Pri ostalih stenah (`IfcWall`) predstavitev geometrije lahko temelji na izvlečenih telesih (`IfcSweptAreaSolid`), obrezovanju (`IfcBooleanClippingResult`) ter predstavitvi z mejnimi ploskvami (`IfcFacetedBrep`). Pri uporabi dveh različnih geometrijskih predstavitev je potrebno steno razdeliti v dve instanci `IfcWall`. Podobno kot pri enostavnih stenah lahko z

uporabo vsaj dveh načinov prikaza dosežemo večpredstavnost. Implementacija slednje je nujna pri izvlečenih telesih, medtem ko pri predstavitvi s ploskvami ni nujna – je pa zaželeno. Osnovo (odtis) pri izvlečenih telesih predstavlja zaprt lik poljubne oblike brez odprtín. Pri predstavitev s ploskvami ne nastopajo Boolove operacije, saj lahko odstopanja od pravilnih oblik opišemo že z osnovnimi elementi predstavitve. Ne glede na uporabljeno geometrijsko predstavitev je vedno potrebno kompleksne predstavitve sten, s katerimi operirajo nekateri programi za načrtovanje, razdeliti na enostavne primitive.

#### **2.3.2.1.5.2 Odprtine**

Odprtine se opisuje z entiteto `IfcOpeningElement`. Potekajo lahko skozi celotno debelino primitivov (ni nujno – npr. niše) in niso specifično vezane na elemente zgradb (stene, plošče ali strehe). Odprtine so vedno postavljene relativno glede na lokalni koordinatni sistem elementa, geometrijsko pa so lahko predstavljene s površinami (Brep) ali pa kot izvlečena telesa. Vsaj z os koordinatnih sistemov elementa in odprtine naj bi se ujemala. Odprtine v elementih se podaja z razmerjem `IfcRelVoidsElement`, prisotnost polnil (vrata, okna) v odprtini pa z razmerjem `IfcRelFillsElement`.

#### **2.3.2.1.5.3 Polnila**

Polnila v definiciji elementov zgradb predstavljajo vrata (`IfcDoor`) in okna (`IfcWindow`), ki jih lahko vključimo le v predhodno definirane odprtine. Okna in vrata praviloma predstavljajo prefabrikate, njihove tipske karakteristike pa opišemo z entitetama `IfcDoorStyle` (`IfcDoorLiningProperties`, `IfcDoorPanelProperties`) in `IfcWindowsStyle` (`IfcDoorLiningProperties`, `IfcDoorPanelProperties`).

#### **2.3.2.1.5.4 Plošče**

Plošče (`IfcSlab`) predstavljajo nevertikalne ploskovne konstrukcijske elemente: medetažne plošče, strešne plošče in klančine. Tudi plošče delimo na enostavne (enotna debelina po prerezu, ekstrudiranje pravokotno na osnovni profil, neobrezano) ter kompleksne. Podobno kot pri stenah je tudi pri ploščah geometrijo možno predstaviti na tri načine (izvlečena in obrezana telesa, mejne ploskve).

### 2.3.2.1.5.5 Ostali primitivi

Tudi za opis geometrije preostalih primitivov se uporabljajo že predstavljene metode: izvlečena in obrezana telesa ter mejne ploskve. V nadaljevanju so zato navedene le entitete ter njihov osnovni opis:

- prečke (IfcBeam) označujejo horizontalne oz. skoraj horizontalne ne nujno nosilne elemente zgradb, pri katerih so dimenzije prečnega prereza relativno male v primerjavi z dolžino elementa. Opisu skupnih lastnosti prečk je namenjena entiteta IfcBeamType;
- stebri (IfcColumn) predstavljajo vertikalne oz. skoraj vertikalne ne nujno nosilne elemente zgradb. Podobno kot pri prečkah tudi pri stebrih skupne lastnosti le-teh opisujemo s posebno entiteto IfcColumnType;
- kritje (IfcCovering) – entiteta IfcCovering označuje elemente, ki pokrivajo preostale primitive (npr. stene, plošč). Kritje se lahko predpiše robu prostora (z IfcRelSpaceBoundary) ali pa preostalim elementom zgradb (z IfcRelCoversBldgElements);
- stopnice (IfcStair) predstavljajo celotno konstrukcijo, ki omogoča prehod med etažama (npr. stopniščna rama, podest, stopniščna rama);
- stopniščna rama (IfcStairFlight) označuje neprekinjeni del stopnišča (npr. med kletjo in podestom);
- rampa (IfcRamp) predstavlja povezavo (koridor) med dvema različnima nivojema. Posamezne stopnice ne smejo biti vključene v rampe, podesti pa lahko;
- rama rampe (IfcRampFlight) predstavlja ravni segment rampe, ki povezuje različne nivoje;
- ograje (IfcRailing) predstavljajo elemente, ki ljudem nudijo oporo ter preprečujejo poškodbe. Geometrijsko so ograje lahko predstavljene kot mejne škatle, površinski modeli, robni modeli ter preslikane predstavitve;
- proksi element (IfcProxy) predstavlja zbirnik entitet, ki imajo lahko tudi geometrijsko predstavitev in jih uvrstimo v prostor;

- streha (IfcRoof) predstavlja zbirnik entitet (plošče, prečke ipd.), s katerimi opišemo predstavljeni element zgradbe. Entiteta IfcRoof tudi z geometrijskega stališča predstavlja zbirnik, saj so za geometrijski opis gradnikov lahko uporabljeni različni pristopi.

### **2.3.3 Razširitev specifikacije IFC na področje statične analize konstrukcij (ST4)**

#### **2.3.3.1 Projekt ST4**

Razširitveni projekt razvoja aplikacijskega modela za analizo konstrukcij s formalno oznako ST4 so v letih 2001/2002 izdelali na Tehnični univerzi v Dresdnu. S projektom je bil razširjen obseg domenske platforme z domeno gradbene analize konstrukcij ter z domeno jeklenih konstrukcij. Slednja predstavlja priredbo obsežnega nemškega standarda »Produktschnittstelle Stahlbau« (informacijski model za jeklene konstrukcije) v okviru standarda IFC, vendar v tej nalogi ne bo podrobneje predstavljena.

Termin analiza konstrukcije bo v nadaljevanju označeval statično analizo konstrukcije, ki jo v procesu načrtovanja na podlagi izbranega računskega modela praviloma opravlja gradbenik konstruktor. Termin računski model v tej nalogi označuje analitični model, ki predstavlja osnovo za računsko (statično in dinamično) analizo konstrukcije. Računski model se razlikuje od fizičnega. Le-ta v splošnem semantično dopolnjuje IFC model in je namenjen detajlnemu načrtovanju.

Žal razširitev ST4 ne pokriva vseh aspektov računskega modeliranja konstrukcij, ki se pojavljajo v domeni gradbenikov konstruktorjev. Z razširitvijo je mogoče:

- definirati ravninsko in prostorsko analizo konstrukcij, s pomočjo katere se v ustrezni programski opremi lahko tvori računski model. Le-tega je potrebno ustrezno dopolniti/prilagoditi s podatki o: 1.) točkovnih, linijskih in ravninskih elementih, 2.) podpiranju, 3.) povezavi elementov in podpor;
- določiti obtežbe, ki vključujejo točkovne, linijske, ploskovne in temperaturne obtežbe z možnostjo povezovanja v obtežne skupine in kombinacije;

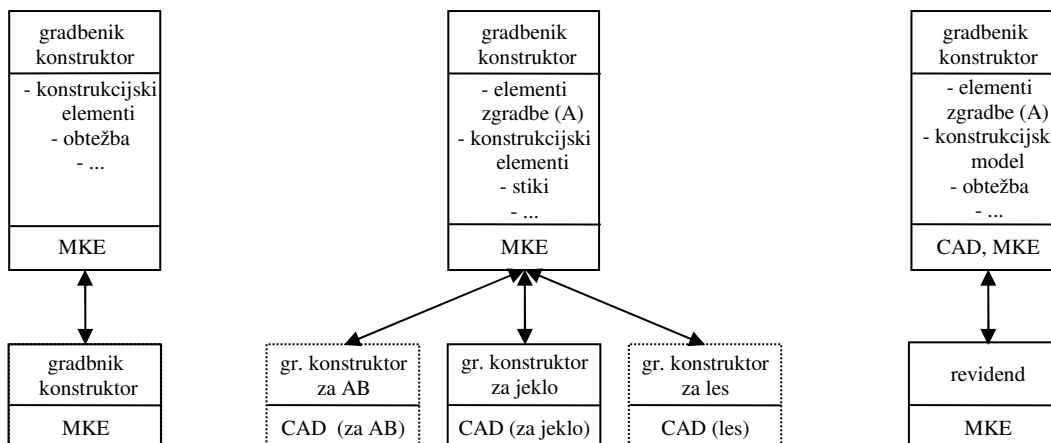
- specificirati različne »podmodele« zgradbe, potrebne za prikaz različnih aspektov oz. delov zgradbe. Odvisnost med modeli je moč tudi shraniti;
- definirati razmerja med obstoječimi elementi zgradbe (arhitektura) in elementi, uporabljenimi pri analizi konstrukcije;
- predstaviti rezultate analize (omejitev na sile in pomike).

Izven dosega razširitve so ostali:

- dinamična analiza,
- obtežba s prednapenjanjem,
- topologija končnih elementov,
- detajlni prikaz rezultatov analize po metodi končnih elementov (deformacije, napetosti).

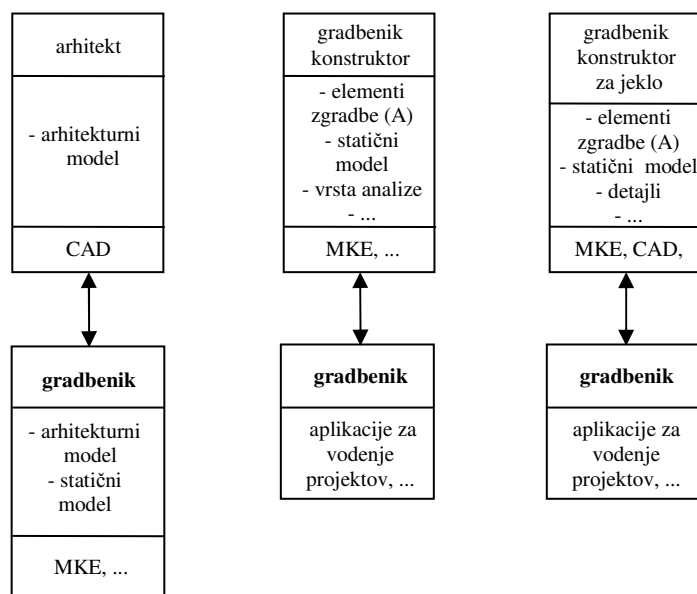
### 2.3.3.1.1 Scenariji uporabe

V skladu z metodologijo razširitev modela zgradb IFC je pred kreiranjem novih razredov in entitet potrebno določiti akterje, ki jih razširitev zadeva, ter pripraviti ustrezne scenarije uporabe. Privzeti sliki 13 ter 14 prikazujeta nekaj najbolj verjetnih scenarijev izmenjave. Predstavljena naloga se osredotoča na prvi primer, prikazan na privzeti sliki 14a.



Privzeta slika 13: Scenariji uporabe znotraj domene analize konstrukcij (Weise et al., 2003)

Adopted figure 13: Typical scenarios inside structural engineering domain (Weise et al., 2003)



Privzeta slika 14: Interoperabilnost s preostalimi domenami (Weise et al., 2003)

Adopted figure 14: Interoperability with other domains (Weise et al., 2003)

### 2.3.3.1.2 Osnovni koncepti razširitve

Osnovne primitive geometrijskega oz. arhitekturnega modela se opisuje z abstraktnim razredom `IfcBuildingElement` oz. njegovimi podrazredi in entitetami (`IfcBeam`, `IfcColumn`, itd). Navedene entitete ter pripadajoči atributi niso vedno primerni za opis računskega modela saj:

- elementi zgradb ne omogočajo neposrednega opisa vseh karakteristik gradnikov računskega modela (teoretično bi sicer lahko manjkajoče karakteristike opisali z dodatnimi lastnostmi),
- med obravnavanima modeloma nastopajo semantične razlike. Primeri: 1.) Monolitna AB plošča z nosilcem je v geometrijskem modelu predstavljena z dvema ločenima elementoma, v računskem modelu pa se predstavljeni sklop med drugim lahko opisuje tudi kot T prerez. 2.) Delitev zgradbe na prostore v arhitekturnem modelu ter s tem delitev kontinuirnih nosilcev na segmente se lahko povsem razlikuje od dejanskih razponov med podporami, ki predstavljajo osnovo statičnega modela.

Razlike med arhitekturnim in računskim modelom prikazuje privzeta slika 16.

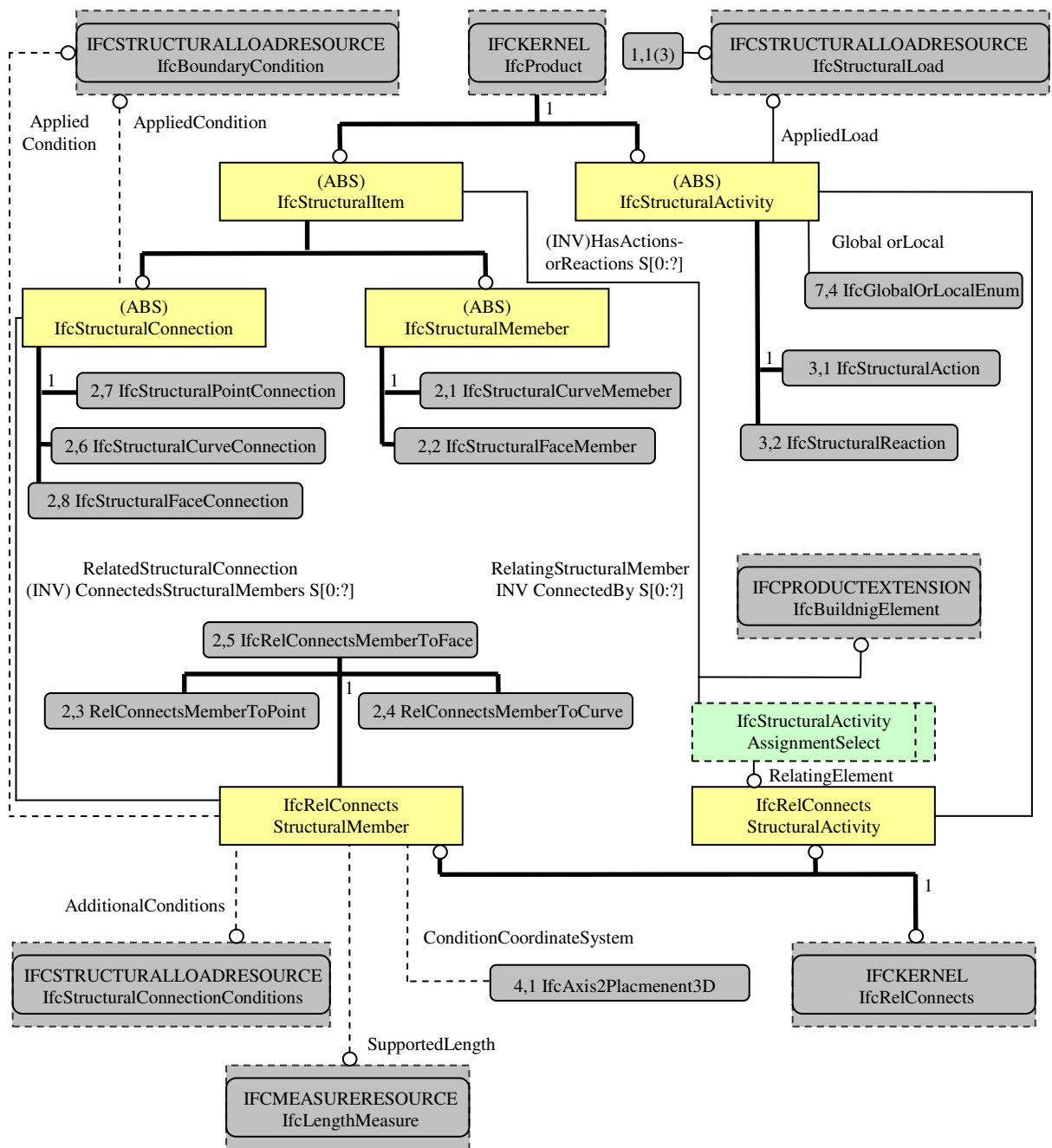
Zaradi opisane neprimernosti elementov zgradb za opis statičnega modela so bile v IFC model vpeljane nove entitete z dvema abstraktnima korenskima razredoma `IfcStructuralItem` (opis konstrukcijskih elementov) ter `IfcStructuralActivity` (opis zunanjih vplivov). Navedena razreda dedujeta lastnosti razreda `IfcProduct`, primitve računskega modela pa se vedno opisuje v globalnem koordinatnem sistemu (privzeta slika 15).

`IfcStructuralItem` je nadalje razdeljen na konstrukcijske elemente (`IfcStructuralMember`) ter stike (`IfcStructuralConnection`), povezuje pa ju entiteta `IfcRelConnectsStructuralMember`. S tem je zagotovljena nedvoumna povezava med elementi in stiki, praktično pa to pomeni, da se ločeno definirajo vozlišča, elementi ter povezave med njimi. Možno je definirati še eno povezavo, in sicer med elementi geometrijskega modela in konstrukcijskimi elementi, in sicer z `IfcRelAssignsToStructuralMembers`, pri čemer kardinalnost povezave opisujejo razmerja 1:1, 1:n ali m:n. V prvem primeru je posamezen element zgradb (npr. `IfcColumn`) predstavljen z enim elementom v računskem modelu (npr. `IfcStructuralCurveMember`). V primeru razmerja 1:n je elementov lahko več. Posamezen element zgradb (npr. dolg nosilec) je lahko opisan z več elementi računskega modela (npr. dejanski razponi nosilca) ali pa v modelu zgradb obstaja več različnih (računskih) predstavitev ter s tem računskih modelov. Razmerja oblike m:n niso dovoljena in jih je potrebno predhodno razdeliti v razmerja 1:n.

Konstrukcijske elemente se nadalje deli na linijske (`IfcStructuralCurveMember`) in ravninske (`IfcStructuralFaceMember`), stike pa na točkovne (`IfcStructuralPointConnection`), linijske (`IfcStructuralCurveConnection`) ter ploskovne (`IfcStructuralFaceConnection`). Stikom lahko predpišemo ustrezno translacijsko oz. rotacijsko togost glede na dane robne pogoje.

Primerjava nekaterih elementov zgradb in elementov računskega modela je prikazana na privzeti sliki 16.

`IfcStructuralActivity` vsebuje dva razreda, `IfcStructuralAction` in `IfcStructuralReaction`, ki omogočata razlikovanje med zunanjimi in notranjimi vplivi. Razmerje med konstrukcijskimi elementi in vplivi podaja razred `IfcRelConnectsStructuralActivity` (podrazred razreda `IfcRelConnects`). Opisano razmerje je potrebno definirati zaradi možnosti podajanja ekscentrične obtežbe.

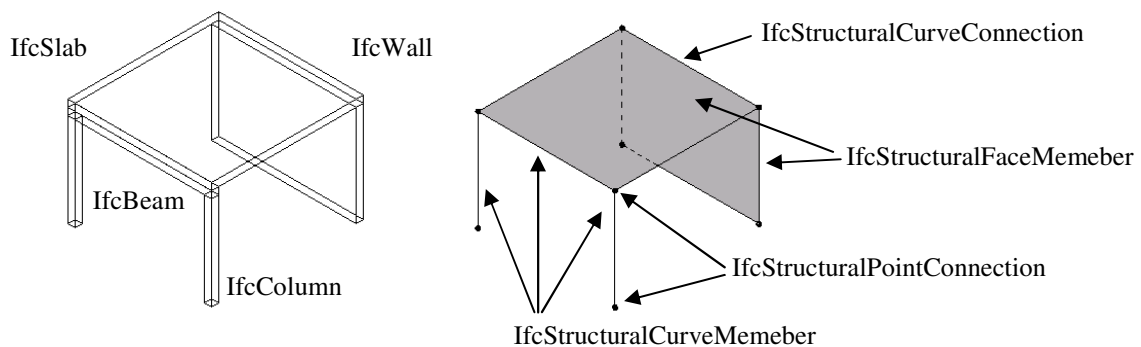


Privzeta slika 15: EXPRESS-G shema razširitve ST4 (IAI, 2008)

Adopted figure 15: EXPRESS-G schema of ST4 extension (IAI, 2008)



Dodatno lahko »aktivnosti« združujemo v `IfcStructuralLoadGroup` (za obtežbo) oziroma v `IfcStructuralResultGroup` (za reakcije), ustvarjene skupine pa lahko kasneje koristno uporabimo pri analizi konstrukcije. Zunanje vplive se lahko poda kot točkovno, linijsko oz. ploskovno obtežbo. Ker je obtežba podana v specifični konstrukcijski shemi virov je uporaba le-te možna tudi v ostalih domenah (podobno je večkratna raba virov rešena pri opisu prečnih prereзов ter pri lastnostih materialov).



Privzeta slika 16: Elementi zgradb: Elementi računskega modela (Weise et al., 2003)

Adopted figure 16: Building elements: Structural elements (Weise et al., 2003)

Znotraj posameznega IFC modela lahko obstaja več računskih modelov. Vsakega izmed njih opisuje entiteta `IfcStructuralAnalysisModel`, s katero med drugim podamo tudi vrsto analize, obtežne primere ter skupine rezultatov pri posamezni obtežbi. Entiteta `IfcRelAssignToGroup` združuje vozlišča, elemente ter obtežne primere, njen zadnji argument pa kaže na pripadajoči model za statično analizo. Dodatno je z uporabo razreda `IfcRelNests` možno vzpostaviti hierarhijo med posameznimi delnimi analizami oz. računskimi modeli.

### 2.3.3.1.3 Razširitve shem

Bistvenega pomena pri definiciji novih entitet je tudi njihova uvrstitev v enega izmed štirih slojev IFC modela.

Pretežni del informacij, ki neposredno zadeva gradbenika konstruktorja (elementi, vozlišča, vplivi na konstrukcijo, definicija računskih modelov), je zbran v domenskem sloju, in sicer v novo definirani domeni analize konstrukcij. Dopolnjen je bil tudi sloj skladnosti, in sicer z

entiteto `IfcStructuralBuildingElementsAssembly`, s katero se združuje konstrukcijske gradnike.

Obsežne dopolnitve so prisotne tudi v sloju virov. Dopolnjeni so bili:

- vir mer: definicija novih količin (npr. masni vztrajnostni moment, temperaturni gradient, toplotni ekspanzijski koeficient, elastična podajnost linearnega elementa na dolžino enote, sprememba nagiba na enoto dolžine, sprememba mase na enoto dolžine ipd.);
- vir profilov: razširitev predstavlja enajst novih razredov za parametrično predstavitev (jeklenih) profilov (asimetrični I profili, C profili, tirnice, kotniki, pravokotni profil z zaobljenimi robovi, T, U in Z profili);
- vir lastnosti profilov predstavlja poleg vira lastnosti materialov in vira obtežb noviteto v sloju virov. Lastnosti profilov so najprej razdeljene glede na specifične konstrukcijske potrebe, ki hkrati predstavljajo osnovo za nadaljnjo delitev (glede na materialne lastnosti). Pri opisu lastnosti profilov je potrebno podati razmerje med virom lastnosti profilov (`IfcProfileDef`) ter virom profilov (`IfcProfileProperties`);
- lastnosti materialov. Ta vir je razdeljen na mehanske lastnosti (Youngov modul, strižni modul, Poissonov koeficient, temperaturni koeficient), mehanske lastnosti jekla (utrjevanje, relaksacija ipd.), splošne materialne lastnosti (specifična teža, poroznost ipd.), razširjene materialne lastnosti, higroskopske lastnosti materiala, optične lastnosti ter toplotne lastnosti materiala;
- viri geometrije, topologije in predstavitve. Navedeni viri vsebujejo minimalne dopolnitve, s katerimi je omogočena predstavitev zapletenih geometrijskih oblik;
- vir obtežb. S petnajstimi entitetami je v viru obtežb opisan celoten spekter zunanjih vplivov na konstrukcijo. Poleg posplošenih točkovnih, linijskih in ploskovnih sil lahko podamo tudi premike in temperaturno obtežbo. Možno je definirati tudi odpoved (nenosilnost) stika v nategu oz. tlaku. Vir obtežb zajema tudi opis robnih pogojev (translacijske oz. rotacijske togosti) elementov oz. stikov (`IfcBoundaryEdgeCondition`, `IfcBoundaryEdgeCondition`, `IfcBoundaryEdgeCondition`).



## **3 PRESLIKAVE V INFORMACIJSKIH MODELIH ZGRADB**

Preobsežnost informacijskih modelov zgradb se rešuje z definiranjem pogledov na model, to je z definicijo domensko usmerjenih podmodelov. Podmodeli ne predstavljajo oddaljevanja od osnovne ideje informacijskega modela zgradbe, temveč le poenostavljajo oz. omogočajo praktično implementacijo kompleksnih specifikacij v praksi. Med podmodeli obstajajo tudi relacije, in sicer 1.) znotraj posamezne discipline, 2.) med posameznimi disciplinami ter 3.) med posameznimi fazami projekta (Nour, 2007).

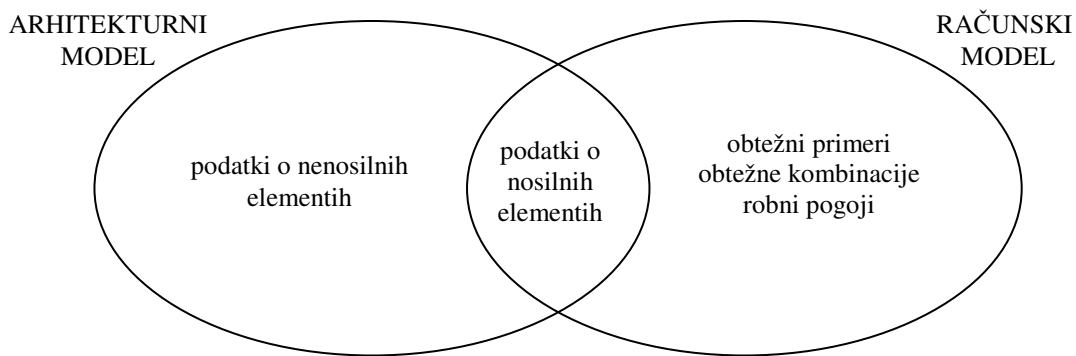
Ta naloga se osredotoča na relacije med posameznimi disciplinami, to je na relacije med arhitekturnimi in računskimi aspekti v IFC specifikaciji. Navedene relacije je zaradi uporabe enotnega informacijskega modela zgradb mogoče vsaj delno avtomatizirati ter jih tudi zato običajno karakteriziramo kot preslikave. Predstavljeni so izsledki raziskav na predstavljenem področju, predlagana je specifikacija pogledov arhitektura – računsko modeliranje (A-RM pogled) ter računsko modeliranje – računska analiza (pogled RM-RA). V sklepnem delu poglavja je izbran najprimernejši algoritem prirejanja in predlagan izvorni pristop k reševanju celotnega problema obravnavanih preslikav med modeli.

### **3.1 Domenski modeli**

Vsaka stroka, vpletena v življenjski cikel zgradbe, si le-to predstavlja z lastnimi domensko specifičnimi modeli. Slednji so postali aktualni z razmahom specifične programske opreme (npr. toplotni odziv konstrukcije, računska analiza konstrukcije ipd.).

Izmed množice domensko orientiranih modelov predmet zanimanja te naloge predstavljata dva modela: arhitekturni model ter računski model (slika 12). Arhitekturni model vsebuje informacije o nosilnih in nenosilnih elementih zgradbe (lokacija, prečni prerez, materialni podatki), medtem ko naj bi v splošnem računski model dodatno vseboval informacije o prenosu vertikalne in horizontalne obtežbe ter o obtežbi in obtežnih kombinacijah (oz. o vplivih na konstrukcijo).

Kot bo prikazano v nadaljevanju pa takšno pojmovanje ni povsem v skladu s smernicami IFC specifikacije. Čeprav posamezni podmodeli v IFC specifikaciji (še) niso striktno definirani, pa slednja načeloma ločuje med računskim modelom (natančen model nosilne konstrukcije) in modelom za analizo (model, ki se uporabi za računsko analizo konstrukcij) (slika 13).



Slika 12: Arhitekturni model – računski model: relacije med modeli

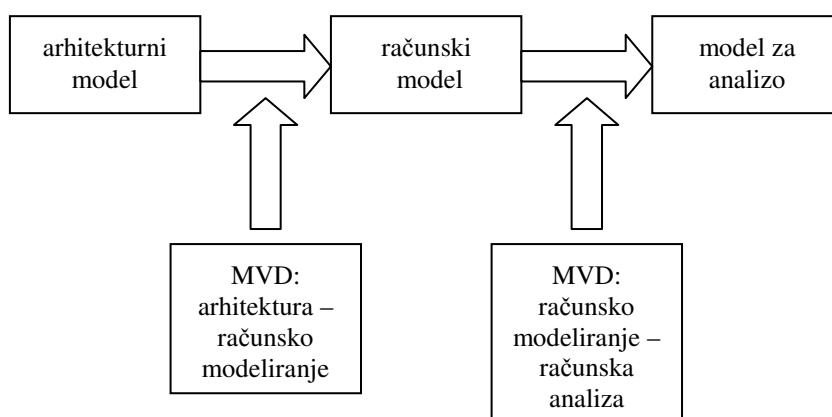
Figure 12: Architectural model – Structural model: relationships

Za domensko specifično modeliranje je značilna samozadostnost modelov – v posamezni domeni je možno z uporabo izbrane specifikacije zgradbe opisati do predpisane stopnje podrobnosti natančno. Ker domene niso popolnoma neodvisne (npr. geometrija zgradb nastopa v večini modelov), takšen pristop običajno vodi k podvajanju informacij. Podvajanje z vidikov optimizacije sicer ni najbolj zaželeno, a je v praksi pogosto namenoma spregledano in celo pogojno uporabno (npr. pri iterativnem načrtovanju). Vsekakor pa je v globalnem modelu zgradbe potrebno zagotoviti konsistentnost ponavljajočih se informacij.

V prejšnjem poglavju sta bila predstavljena dva v IFC specifikacijo vključena domenska modela (arhitektura, model za računsko analizo). Vsakemu izmed navedenih podmodelov pripada določen nabor entitet. Ne glede na izpostavljene nepopolnosti modela za analizo se z definiranimi entitetami vsaj približno skuša opisati konstrukcijo, kot si jo predstavlja gradbenik konstruktor. Podrobnejša analiza konstrukcije (generacija mreže končnih elementov, natančen prikaz rezultatov analize – pomiki, deformacije, napetosti ipd.) je trenutno izven dosega razširitve. Opis trenutno nepokritih področij naj bi bil namenjen ločen razširjeni model za detajlno analizo, čeprav se ob tem postavlja vprašanje smiselnosti drobljenja domen oz. poddomen in uvajanja novih podmodelov. Z razširjenim modelom za

detajlno analizo naj bi gradbenik konstruktor na predpisan način opisal obnašanje konstrukcije pri podani obtežbi in rezultate podrobnega dimenzioniranja zapisal v IFC informacijski model zgradbe.

Eno izmed osnovnih predpostavk razširitve ST4 predstavlja neustreznost entitet arhitekturnega modela za opis računskega modela in uvedba novih entitet za računsko analizo konstrukcij. Problem preslikav med modeloma očitno zahteva vključitev izbranih entitet obeh specifičnih modelov ter določitev relacij med njimi.



Slika 13: Načrtovanje zgradb – relacije med pod modeli v modelu zgradb IFC

Figure 13: Building design – submodels relations in IFC based BIM

Nezadostnost razširitve ST4 naj bi odpravil razširitveni projekt ST7: Končni elementi ter dinamična analiza konstrukcij. Predlog projekta se je pojavil že leta 2004, vendar še ni bil realiziran. V okviru predloga razširitve ST7 je predvideno prirejanje dveh MKE modelov:

- modelu za analizo (definiranem v razširitvi ST4) se priredi mehanski MKE model,
- arhitekturnemu modelu se priredi arhitekturni (oz. fizični) MKE model.

Glede na že opisane pomanjkljivosti elementov zgradb (oz. arhitekturnih modelov) je smiselnost prirejanja MKE modela arhitekturnemu vprašljiva, saj z arhitekturni primitivi ni mogoče zajeti vseh potrebnih informacij za računsko analizo zgradbe.

Ker podrobni model oz. modela za analizo še nista definirana, se z razširitvijo ST4 skuša kar v največji meri opisati aspekte računskega modeliranja. Takšen pristop je privzet tudi v

predstavljeni nalogi, ki sicer neposredno ne obravnava analize konstrukcij, temveč se osredotoča izključno na prirejanja med arhitekturnim in računskim modelom.

### **3.1.1 Domenski modeli in izmenjava**

Delna izmenjava informacij znotraj informacijskega modela zgradb je načeloma možna na dva načina (Lockley et al., 2000):

- delna izmenjava na podlagi sheme,
- delna izmenjava na podlagi instanc.

Delna shema določa nabor relevantnih objektov, s katerimi se lahko opiše določen podmodel oz. se omeji na določeno aktivnost. Primer takšnega opisa predstavlja izmenjava vseh sten, ne glede na morebitne odprtine. Pri takšnem prirejanju je zaželena uporaba temu namenjenega jezika (EXPRESS-X). Definicija izmenjave na podlagi sheme predstavlja kompleksno nalogo, katere rešitev je pogojena z dobrim poznavanjem osnovne sheme in principov produktnega modeliranja. Takšen pristop je praviloma v rabi pri definiranju preslikav celotnih poddomenskih modelov.

Alternativo predlaganem pristopu predstavlja prilagodljivo instančno orientirano okolje za delno izmenjavo (Flexible Instance Oriented Partial Exchange Environment - FIOPE) (Nour, 2007). Takšen pristop naj bi bil mnogo bolj prijazen do končnega uporabnika, saj se s podmodeli operira na nivoju entitet (oz. njihovih instanc) in ne na nivoju sheme. Avtor FIOPE pristopa uporabo le-tega favorizira z višjo stopnjo svobode pri izmenjavi podmodelov in večjo razumljivostjo oz. prijaznostjo do končnega uporabnika, hkrati pa opozarja na nujno zagotavljanje konsistentnosti ter izogibanje redundantnosti. Pri FIOPE pristopu lahko v splošnem končni uporabniki v skladu z načeli IFC specifikacije sami določajo entitete (in attribute), ki se bodo izmenjevali. Na takšen način se lahko zajamejo potrebe posameznih domen pri izmenjavi ter v končni fazi definira delna izmenjava na podlagi sheme.

Ker je narava izmenjave delnih modelov oz. definicija pogledov na model bližje FIOPE metodologiji, bo navedeni pristop uporabljen v tej nalogi.

## 3.2 Preslikave med modeli znotraj informacijskega modela zgradb IFC

Domensko specifične modele se v informacijski model zgradb IFC vključuje z razširitvenimi projekti. Vsaka domenska razširitev prinaša v IFC specifikacijo nove entitete, ki opisujejo domensko specifično razumevanje primitivov in ki so glede na njihovo sporočilnost razvrščene v ustrezen sloj specifikacije. Dodatno so v IFC specifikaciji na voljo entitete za opis relacij med primitivi posameznih modelov. Določitev relacij med entitetami arhitekturnega ter računskega modela ostaja kot specifični problem preslikav med modeli in je izven dosega oziroma izven ožjega namena IFC specifikacije. Po uveljavljenem delotoku obravnavane preslikave in njihove umestitve v informacijski model zgradb ostajajo v domeni gradbenika konstruktorja.

### 3.2.1 Uporabni minimum

Smiselna povezava sorodnih konceptualnih blokov (oz. entitet), s katerimi je moč zadovoljivo opisati večino informacijskih modelov novejša literatura (Lehtinen, Hietanen, 2007) označuje kot »uporabni minimum specifikacije«. Navedeni termin predstavlja sodobno različico »pogledov na model«, ki v praksi niso izpolnili pričakovanj oz. njihov učinek ni bil primerljiv z alternativami (npr. aplikacijsko specifičnimi vmesniki). Podobno kot pri pogledih na model se tudi pri »uporabnem minimumu« obseg modelov krči, vendar pa se viri ohranjajo. Takšen minimum – v primeru, da se v implementacijah izkaže za primernega – bi lahko predstavljal osnovo za ustrezno razširjen doseg.

Bistveni element navedenega pristopa predstavlja določitev realističnega izhodišča. Pri tem je kot vplivne faktorje potrebno upoštevati naslednje spremenljivke (v oklepajih so navedeni akterji):

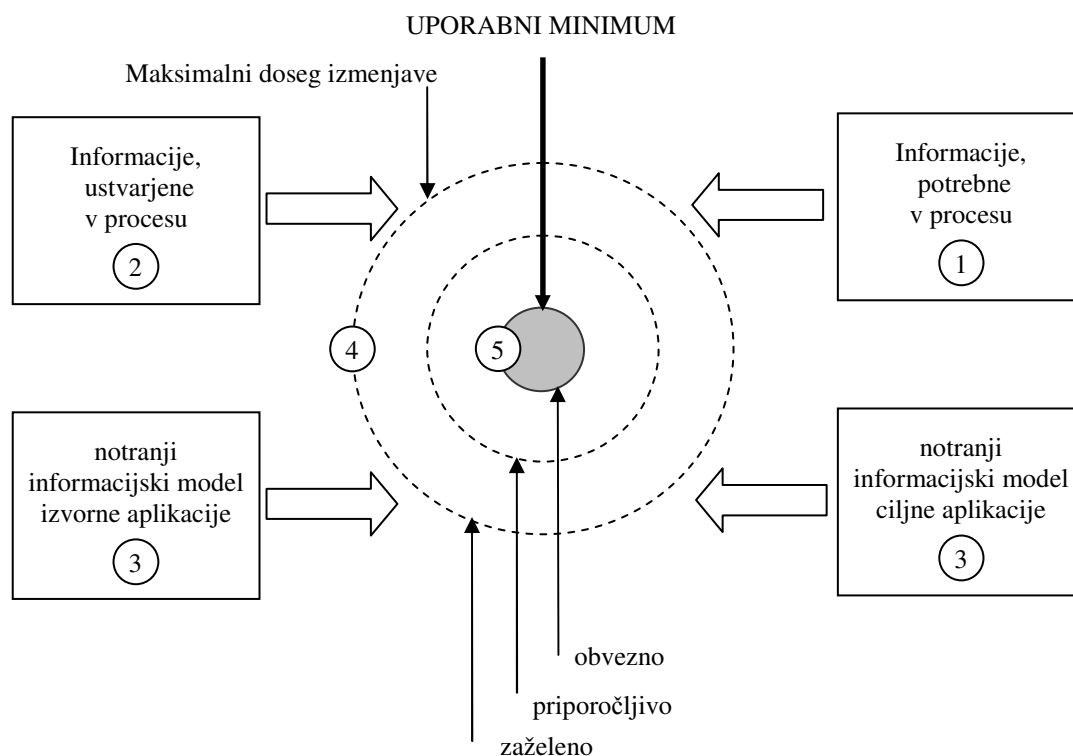
- karakteristike izvornega modela izvorne aplikacije (razvijalci programske opreme),
- konkretne podatke o konkretnem produktu (končni uporabniki),
- karakteristike vmesnika za izvoz modela iz izvorne aplikacije (razvijalci vmesnika),
- karakteristike izvornega modela ciljne aplikacije (razvijalci programske opreme),
- karakteristike vmesnika za uvoz modela v ciljno aplikacijo (razvijalci vmesnika),



- potrebne podatke, ki jih pri uvozu pričakuje ciljna aplikacija (razvijalci vmesnika),
- podatke o produktu, ki jih pričakuje uporabnik ciljne aplikacije (končni uporabniki).

Praktičen primer predstavlja razporeditev prostorov – v arhitekturi je razporeditev prostorov bistveni gradnik načrtovanja, medtem ko v računskem modelu razporeditev prostorov predstavlja le informacijo o namembnosti prostora in posledično o vplivih na konstrukcijo. Nasprotno pa je 3D geometrija prostorov in ovoj zgradbe bistvenega pomena pri izračunu toplotnih izgub.

Izhodišče pri določitvi »uporabnega minimuma« predstavlja specifikacija informacij (oz. entitete), ki bodo predstavljale predmet izmenjave. V veliki večini primerov za opis geometrije zadostuje le nekaj entitet specifikacije (potrebne entitete). Pri konkretnih modelih zgradb je obseg entitet lahko še manjši (uporabljene entitete).



Slika 14: Uporabni minimum

Figure 14: Usefull minimum

Z »uporabnim minimumom« postaja vprašljiva smiselnost ozadja kompleksne IFC specifikacije pri izmenjavi minimalnega nabora informacij. Navedena trditev, ki izključuje integrirani podatkovni model, je povsem neutemeljena: »uporabni minimum« namreč predstavlja izhodišče za širjenje obsega izmenjave. Manjši obseg izmenjave v izhodiščni fazi zanesljivo rezultira v robustnejših vmesnikih, pa tudi širitev dosega že izvedenih vmesnikov je enostavnejša, če je v ozadju izmenjav integrirani podatkovni model.

### **3.3 Preslikave med arhitekturnimi in računskimi modeli v IFC specifikaciji**

#### **3.3.1 IFC skladna programska oprema za analizo konstrukcij**

Ponudba certificiranih IFC vmesnikov za programsko opremo za analizo konstrukcij je relativno skromna (Tekla Structures, SCIA-ESA PT) (preglednica 7). Pri tem je potrebno izpostaviti, da se certifikacija nanaša na razširjen koordinacijski pogled, ki ne vključuje entitet modela za analizo. Obstaja pa kar nekaj necertificiranih vmesnikov (Dlubal, ETABS, AxisVM, ETABS, ipd.). Večina certificiranih in necertificiranih vmesnikov je v razvojni fazi in je zato pred njihovo uporabo potrebno kritično oceniti pravilnost delovanja.

V nasprotju s pričakovanji skladnost programske opreme za analizo konstrukcij z IFC specifikacijo v praksi ne predstavlja interoperabilnosti med navedeno programsko opremo, temveč se nanaša na interpretacijo IFC arhitekturnega modela oziroma na sposobnost prirejanja posameznih elementov arhitekturnega modela primitivom modela za analizo. Takšno ne najbolj posrečeno pojmovanje skladnosti z IFC specifikacijo izhaja iz uveljavljenega delotoka, kjer gradbenik konstruktor na podlagi arhitekture zgradbe določi model za analizo. Pri tem je potrebno izpostaviti, da v vmesnikih ni implementirana celotna IFC specifikacija, temveč le entitete razširjenega koordinacijskega pogleda. V maloštevilnih vmesnikih (Tekla, Dlubal, AxisVM) pa je poleg navedenih entitet delno implementirana razširitev ST4.

Vsak branju namenjen IFC vmesnik programske opreme za analizo konstrukcij ima vgrajen svoj izviren (praviloma neprilagodljiv) način navedenega prirejanja med modeli. V preglednicah 8 in 9 so predstavljene načrtovane preslikave IFC vmesnika programa ETABS.

Trenutno je v prototipu vmesnika implementiran le manjši del načrtovanih preslikav, ki se v večji meri nanašajo le na arhitekturo oz. na geometrijo produkta. Podobno kot pri IFC vmesnikih ostalih aplikacij za računsko modeliranje konstrukcij shranjevanje modela za analizo v IFC zapisu ni možno. Vsi preskušeni vmesniki namreč primitive modela za analizo shranijo kot gradnike arhitekturnega modela. Takšen princip shranjevanja in posledično rezultirajoči model zgradb je z vidika ohranitve informacij semantično šibkejši, kot bi bil model z uporabo gradnikov za računsko analizo konstrukcij (entitete ST4 razširitve).

#### Preglednica 8: Preslikave v programu ETABS – uvoz v model IFC

Table 8: Mapping in ETABS – IFC model import

<b>Ifc entiteta</b>	<b>način predstavitve, pogoji</b>	<b>primitivi programa ETABS</b>
IfcColumn	ekstrudiran pravokotnik	steber
IfcBeam	ekstrudiran pravokotnik	prečka ali povezje (glede na orientacijo)
IfcWallStandardCase	ekstrudiran pravokotnik, konstantna debelina	stena
IfcSlab	ekstrudiran pravokotnik, konstantna debelina	plošča ali rampa (glede na orientacijo)

#### Preglednica 9: Preslikave v programu ETABS – izvoz v model IFC

Table 9: Mapping in ETABS – IFC model export

<b>primitivi programa</b>	<b>pripadajoča Ifc entiteta</b>	<b>opomba</b>
etaža	IfcBuildingStories	
steber	IfcColumn	pravokotni nespremenljiv prerez
prečka	IfcBeam	pravokotni nespremenljiv prerez
povezje	IfcBeam	pravokotni nespremenljiv prerez
stene	IfcWallStandard	ekstrudiran pravokotnik, konstantna debelina
plošče	IfcSlab	ekstrudiranje poljubnega lika, konstantna debelina
rampe	IfcSlab	ekstrudiranje štirivozliščnega lika, konstantna debelina

Vseeno tako pridobljeni arhitekturni model predstavlja pomemben gradnik zaokroženega iterativnega načrtovanja med arhitekti ter gradbeniki konstruktorji. Preglednici 8 in 9 podajata tudi trenutno stopnjo implementacije. Očitno je vmesnik še v začetni razvojni fazi, zato navedeni seznam entitet ne predstavlja »uporabnega minimuma«, kvečjemu osnovo zanj.

### **3.3.2 Raziskave na področju prirejanja med pod modeli IFC specifikacije**

Nadomestitev črtne predstavitve elementov zgradb s tridimenzionalnim objektno orientiranim zapisom (npr. model zgradb IFC) predstavlja temelj raziskav na področju avtomatizacije prirejanja med arhitekturnimi modeli, računskimi modeli ter modeli za analizo. Pri tem je potrebno ponovno izpostaviti, da model zgradb IFC vsebuje le specifikacijo zapisa podmodelov, ne pa tudi algoritmov za dedovanje oz. prirejanje geometrije med modeli.

Število dokumentiranih raziskav, ki obravnavajo predstavljeno področje, je relativno skromno. Izhodišče vseh raziskav predstavlja arhitektura zgradbe, zapisana v informacijskem modelu zgradb IFC. Preslikave so v posameznih raziskavah definirane različno, ravno tako pa se razlikuje način shranjevanja modela za analizo. Nobena izmed opisanih raziskav ne upošteva opisanih temeljnih izhodišč specifikacije za njeno širitev oz. za implementacijo. Tako npr. v nobeni raziskavi ni uporabljena IFC specifikacija za zapis modela za analizo, temveč vse v nadaljevanju predstavljene raziskave uvajajo svoj edinstven način opisa modela za analizo.

#### **3.3.2.1 Uporaba IFC specifikacije pri prefabriciranih armiranobetonskih konstrukcijah**

V okviru raziskav o implementaciji IFC specifikacije ter švedske nacionalne gradbene klasifikacije BSAB je bil izdelan prototip (IMPACT) prirejanja geometrije med različnimi pod modeli pri načrtovanju prefabriciranih armiranobetonskih konstrukcij (Ronneblad, Olofsson, 2003). Vnos geometrije namesto na DWG oz. DXF formatu temelji na informacijah o geometriji, pridobljenimi iz informacijskega modela zgradb IFC. Za omenjeni program je bil izdelan vmesnik, ki pri uvozu omogoča ročno izbiro entitet geometrije modela, vse ostale potrebne parametre (vozlišča, podpore, obtežba) pa je potrebno podati ročno. Z opisano raziskavo ni moč dokazati oz. potrditi vseh prednosti uporabe informacijskih modelov, saj

raziskava ne temelji na generaliziranem objektnem prirejanju, temveč le na prirejanju geometrijskih primitivov.

### 3.3.2.2 Računsko modeliranje konstrukcij na podlagi IFC arhitekturnega modela

V raziskavi Univerze v Šanghaju je bila izvedena študija razmerij oz. razlik med modeli v arhitekturi in v gradbeništvu (Deng, Chang, 2006). Raziskava karakterizira IFC arhitekturni model zgradbe kot fizični model zgradbe, ki se v splošnem lahko precej razlikuje od modela za analizo modeliranega s končnimi elementi. Takšnega modela (še) ni možno predstaviti z zapisom IFC, temveč je njegova predstavitev vezana na uporabljeno programsko opremo.

```

<?xml version="1.0" encoding="GB2312"?>
<DATA>
<TABLE Name="System">
  <Control ID="AppInfo" Name="IFC2SGF" Version="1.00" Producer="SJTU-DXY"/>
  <Control ID="FileInfo" Title="DefaultSGF" Author="DXY" Organization="SJTU"
    TimeStamp="11/28/2005 21:47:59" Description=""/>
  <Control ID="Units" LengthUnit="MM" ForceUnit="KN"/>
</TABLE>
<TABLE Name="Storey">
  <Storey ID="Story2" Elevation="3000"/>
  <Storey ID="Story1" Elevation="0"/>
</TABLE>
<TABLE Name="Material">
  <Material ID="CONC" Mass="2.40277E-012" Weight="2.35631E-008" Type="Isotropic"
    E="24.8211" U="0.2" A="9.9E-006"/>
  <Material ID="STEEL" Mass="7.827E-009" Weight="7.682E-005" Type="Isotropic"
    E="199948" U="0.3" A="1.17E-005"/>
</TABLE>
<TABLE Name="Joint">
  <Joint ID="1" X="-4000" Y="10000" Z="0"/>
</TABLE>
<TABLE Name="FrameSection">
  <FrameSection ID="FSEC1" Material="CONC" Shape="Rectangle" XDim="600" YDim="600"/>
</TABLE>
<TABLE Name="Frame">
  <Frame ID="FRAME1" JointI="21" JointJ="22" Type="Column" Section="FSEC1"
    Angle="0" Jointo="21" Storey="Story1"/>
</TABLE>
<TABLE Name="AreaSection">
  <AreaSection ID="Wall1" Material="CONC" MatAngle="0" AreaType="Shell"
    Type="Shell-Thin" Thickness="380" BendThick="380"/>
</TABLE>
<TABLE Name="Area">
  <Area ID="1" Joint1="1" Joint2="2" Joint3="3" Joint4="4" Type="Wall"
    Section="Wall1" Angle="90" Storey="Story1"/>
</TABLE>
<TABLE Name="Opening">
  <Opening ID="1" Joint1="59" Joint2="60" Joint3="61" Joint4="62" AreaID="4"
    Storey="Story1"/>
</TABLE>
</DATA>

```

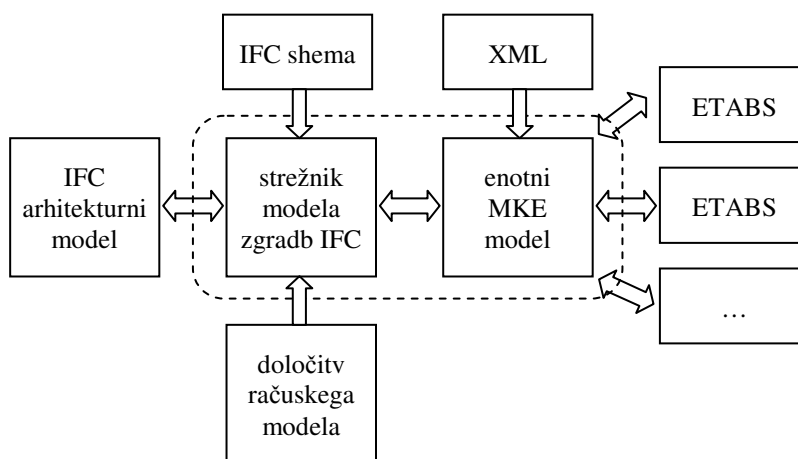
Slika 15: Enotni MKE računski model

Figure 15: Unified finite element model

Nosilne in nenosilne elemente zgradb opisujejo naslednje entitete, uvrščene v sloj interoperabilnosti: IfcColumn, IfcBeam, IfcWall/IfcWallStandardCase, IfcSlab, IfcRamp, IfcStair, IfcRoof, IfcRampFlight, IfcStarFlight. Pri določitvi računskega modela se raziskava omejuje le na prve štiri entitete, ki največkrat predstavljajo preliminarne nosilne elemente. Iz navedenih entitet in pripadajočih atributov obstoječega IFC arhitekturnega modela je možno pridobiti informacije o geometriji (lokacija, prečni prerez, material).

Edinstven prispevek raziskave predstavlja definicija enotnega MKE modela, temelječega na XML zapisu. Struktura zapisa, prikazanega na sliki 15, je zelo podobna tipični strukturi vhodnih podatkov aplikacij za analizo konstrukcij.

Predstavljeni pristop ne omogoča neposredne izmenjave informacij med IFC arhitekturnim modelom ter različnimi aplikacijami za analizo konstrukcij. Slika 16 prikazuje splošno proceduro pri generaciji modela za računsko analizo iz IFC arhitekturnega modela. Le-ta se skupaj s pripadajočo IFC shemo naloži na IFC strežnik. Z uporabo modula za izvleček in združitve se tvori enotni MKE model, kompatibilnost z zapisom obstoječe programske opreme za analizo konstrukcij pa se zagotavlja s specifičnimi vmesniki. Omogočene so povratne preslikave: po zaključku računske analize je možno model preslikati v enotni MKE računski model, le-tega pa je možno integrirati v izvorni arhitekturni model.



Slika 16: Grafikon poteka

Figure 16: Flow chart

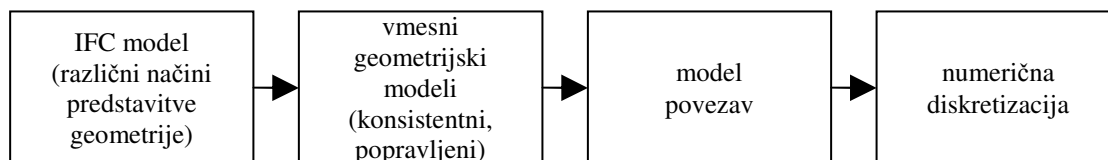
Pri generaciji računskega modela je bistvena medsebojna povezava posameznih gradnikov. V predstavljeni raziskavi je povezava linijskih in ploskovnih elementov predstavljena s t. i. identičnimi vozliščnimi točkami. Po preverjanju pravilnosti prirejanja je potrebno definirati še obtežbo, obtežne primere ter parametre analize.

Posebnost predstavljene raziskave predstavlja uporaba XML formata za opis primitivov računskega modela. IFC specifikacija sicer vsebuje gradnike za opis računskega modela (ST4 razširitev), vendar ne na nivoju končnih elementov. Zato so avtorji raziskave predlagali lasten XML skladen format.

### 3.3.2.3 Računska analiza IFC skladnih modelov zgradb

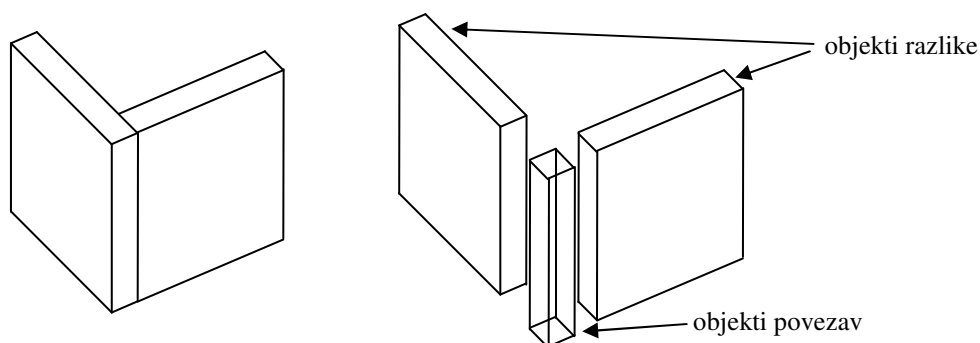
V letu 2004 so bile na Tehnični univerzi v Münchnu (Romberg et al., 2004) opravljene raziskave z namenom izboljšanja računalniško podprtega načrtovanja in upravljanja zgradb. Del raziskav se je nanašal neposredno na načrtovanje arhitekture ter na simulacije, ki se običajno opravljajo pri načrtovanju in ki temeljijo na načrtovani geometriji. V raziskavi je bil predlagan izviren algoritem za pretvorbo geometrije v MKE model (slika 17).

Uporaba tridimenzionalnega modeliranja v okviru informacijskega modeliranja zgradb predstavlja osnovo sodobnih simulacij (npr. računske – statične analize, toplotnega odziva zgradb, itd). Ker IFC specifikacija podpira različne načine predstavitve geometrije se v predlaganem postopku najprej poenoti predstavitev geometrije, in sicer z vmesnima geometrijskima modeloma: v konsistentnem modelu so odpravljene geometrijske vrzeli oz. nekonsistentne odprtine, v popravljenem pa tudi neprimerna križanja elementov. Končni geometrijski model ne temelji na STEP predstavitvi geometrije, temveč se pretvori v ACIS geometrijo (eden izmed najbolj uveljavljenih načinov 3D predstavitve teles oz. 3D modelirnikov).



Slika 17: Podmodeli v računski analizi IFC skladnih modelov zgradb

Figure 17: Submodels in structural analysis of IFC accordant BIMs



Slika 18: Model povezav – objekti povezav, objekti razlike

Figure 18: Connection model – coupling objects, difference objects

Po pretvorbi dobimo »model povezav« (slika 18), kjer je geometrija objekta predstavljena z mejnimi ploskvami (Brep), njegove gradnike pa delimo na: »objekte povezav –  $M_K$ « (mejne ploskve se stikujejo v ravnini) in »objekte razlike –  $M_D$ « (stikovanje po robu ali v vozlišču). Vsak objekt povezave oz. razlike predstavlja zaprt Brep objekt, opisan z vozlišči, robovi ter ploskvami. V splošnem lahko glede na medsebojno lego definiramo naslednje tipe:

- tip NEF – presek objektov predstavljajo ploskve, robovi ter vozlišča,
- tip NE – presek objektov predstavljajo robovi in vozlišča,
- tip N – objekti se sekajo samo v vozliščih.

Pri primerjavi lege objektov razlike in objektov povezav lahko nastopijo naslednji primeri:

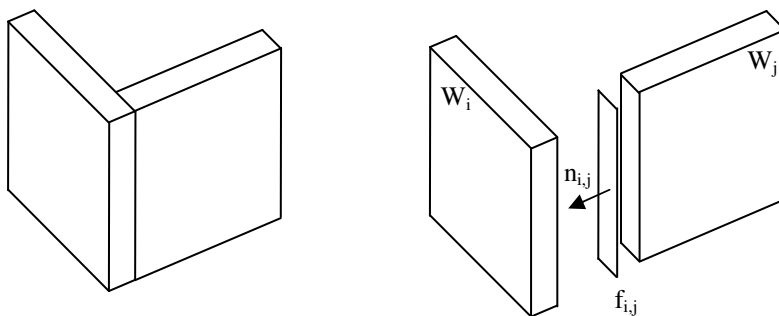
- presek med objekti razlike: N, NE,
- presek med objekti povezave: N, NE, NEF,
- presek med objekti razlike in povezave: N, NE, NEF.

Dekompozicija geometrijskega modela v model povezav za vsak par objektov  $W_i$  in  $W_j$ , za katera obstaja stična ploskev (NEF), poteka po naslednjem algoritmu (vseh objektov je  $W_k$ ) (slika 19):

- definicija stične ploskve –  $f_{i,j}$ ,



- definicija normale na ploskev –  $\mathbf{n}_{i,j}$ ,
- kreiranje kopije objekta  $w_i$ :  $w_i'$ ,
- premik  $w_i'$  vzdolž  $-\mathbf{n}_{i,j}$  za  $\Delta d$ ,
- kreiranje kopije objekta  $w_j$ :  $w_j'$ ,
- $a_1$  predstavlja presek med objektoma  $w_i'$  in  $w_j'$  (objekt  $a_1$  ni nujno končni objekt razlike, saj v stiku lahko v splošnem primeru nastopa več objektov),
- $d_1$  označuje objekt razlike ( $M_D$ ), ki izhaja iz objekta  $w_i$ ,
- $d_2$  označuje objekt razlike ( $M_D$ ), ki predstavlja razliko med objektom  $w_j$  ter objektom povezave  $a_j$ ,
- nadaljevanje dekompozicije z novim parom  $w_i$  in  $w_j$ .



Slika 19: Objekta  $W_i$ ,  $W_j$ , stična ploskev  $f_{i,j}$  in vektor normale  $\mathbf{n}_{i,j}$

Figure 19: Objects  $W_i$ ,  $W_j$ , intersection face  $f_{i,j}$  with normal vector  $\mathbf{n}_{i,j}$

Predstavljen dekompozicija geometrije predstavlja izhodišče za avtomatsko generacijo mreže končnih elementov. Pri formulaciji mreže ni možno uporabiti klasičnih t. i. »h« končnih elementov, saj v splošnem ni mogoče zagotoviti ustreznega razmerja dimenzij končnih elementov (npr. tridimenzionalni končni element bi moral imeti vse robove približno enako velike). Zato so v raziskavi uporabljeni »p« končni elementi, pri katerih lahko nastopajo večja razmerja stranic (tudi do sto). Pomembno je tudi zaporedje generacije mreže končnih elementov. Le-ti se tvorijo v predpisanem zaporedju: najprej se smiselno razdeli

objekte povezave, nato pa še (sosednje) objekte razlike. S takšnim pristopom je mogoče eliminirati neskladnosti na mestu stikov.

Predstavljen način prirejanja predstavlja inovativen pristop k preslikavam znotraj informacijskega modeliranja zgradb. V raziskavi je računski model izdelan vse do nivoja diskretizacije modela na končne elemente. Ker večina sodobne programske opreme za analizo konstrukcij vsebuje dovršene delilnike za generiranje mreže končnih elementov, predstavljena diskretizacija predstavlja nepotrebno podvajanje. Deljenje modela na mrežo končnih elementov znotraj posameznih aplikacij za analizo hkrati rešuje tudi morebitne težave z neustreznostjo izbranega končnega elementa.

#### **3.3.2.4 IFC spletni strežnik za načrtovanje zgradb**

Cilj raziskave Univerze v Singapurju ter Inštituta proizvodnih tehnologij v Singapurju je predstavljala izdelava IFC skladnega spletnega strežnika za načrtovanje zgradb, ki bo omogočal usklajeno načrtovanje arhitektov in gradbenikov konstruktorjev (Chen et al., 2005).

Del celovitega sistema za načrtovanje predstavlja tudi implementacija algoritma za avtomatsko prirejanje (v raziskavi je sicer namesto termina prirejanje uporabljen izraz dedovanje) geometrije med arhitekturnim in modelom za analizo. Slednji mora v splošnem vsebovati vsaj informacije o idealiziranih elementih, povezavah, podporah, mehanskih lastnostih in obtežbah. V raziskavi je bilo predpostavljeno, da je iz arhitekturnega modela možno pridobiti le informacije o idealiziranih elementih ter o povezavah. Obravnavani so le prizmatični elementi (stebri, prečke), ploskovni gradniki (plošče, stene) so izvzeti.

Medsebojna lega dveh izbranih geometrijskih primitivov v prostoru je v splošnem povsem poljubna, za računsko analizo sta zanimiva le primera, ko se elementa sekata oz. ko sta povezana. Preslikava geometrije arhitekturnega modela v model za analizo poteka v treh korakih:

- ugotavljanje povezav med elementi,
- definicija idealiziranih gradnikov modela za analizo,
- določitev stikov med elementi računskih modelov.

Opisano prirejanje geometrije predstavlja del kompleksnejšega sistema za skupno delo arhitektov in gradbenikov pri načrtovanju zgradb. V prvem koraku arhitekt po uspešni prijavi

prenese IFC model na spletni strežnik (komunikacija poteka preko spletnega vmesnika). Po uspešnem prenosu modela na strežnik se po predstavljenem postopku generira računski model, ki se shrani v predlaganem XML formatu. Po uspešni prijavi lahko gradbenik konstruktor preko grafičnega vmesnika pregleda računski model konstrukcije. Vsak element modela lahko gradbenik konstruktor dopolni s komentarji, ki se med drugim nanašajo tudi na spremembo prereza in lege posameznega elementa (neposrednih pravic do sprememb na modelu gradbenik konstruktor nima). Komentarji se shranijo na strežniku in se posredujejo arhitektu. Le-ta z upoštevanjem komentarjev model ustrezno korigira ter ga po že opisanem postopku posreduje v ponovni pregled. Očitno predstavljen način dela rezultira v iterativno načrtovanje, poznano iz prakse.

Sistem je bil v okviru raziskave prototipno implementiran (uporabljeno je bilo javansko okolje).

V raziskavi predlagan pristop je očitno soroden pristopu, navedenem v predlogu teme disertacije. Izpostavi je potrebno nekaj pomanjkljivosti predstavljenega pristopa, ki se jim bo v disertaciji skušalo izogniti:

- Kriteriji oz. vzroki za definicijo lastnega XML formata za zapis računskega modela niso natančno opredeljeni. Definicija lastnega XML formata za zapis računskih modelov oz. neupoštevanje ustrezne razširitve (ST4) ne prispeva k popularizaciji informacijskega modela zgradb IFC.
- Omejenost izključno na linijske elemente.
- Pomanjkljivost raziskave predstavlja tudi neustrezna vključitev specifičnih aspektov načrtovanja (npr. avtomatizacija vključitev rezultatov preliminarnih računskih analiz).
- Predstavljen prototip je v celoti zasnovan na neobjavljeni lastni izvorni kodi. Zato predstavljenega sistema ni bilo mogoče preskusiti oz. kritično oceniti.

### **3.3.3 Zahteve pri preslikavah med arhitekturnimi in računskimi modeli in scenarij uporabe**

Na osnovi podane hipoteze, analize IFC specifikacije ter opisa relevantnih raziskav lahko zberemo zahteve, katerim mora ustrezati načrtovani sistem za zaokroženo načrtovanje. Navedene so le ključne zahteve, ugotovljene med raziskavo. Številne manjše zahteve je moč

najti znotraj pomembnejših zahtev ali pa so že kar ne popolnoma eksplicitno vključene v predlagano rešitev. Kot bistvene zahteve lahko izpostavimo:

- Interoperabilnost mora temeljiti na uveljavljeni in razširjeni specifikaciji (IFC).
- Posamezne faze načrtovanja oz. informacije, ki opisujejo le-te, morajo biti natančno opredeljene.
- Porazdeljena narava dela pri načrtovanju zgradb zahteva porazdeljeni vmesnik za sodelovanje vseh vpletenih.
- Možnost uporabe različne strojne in programske opreme.
- Zagotavljanje varnosti (avtentičnost, avtorizacija oz. opredelitev lastništva informacij).
- Ključni procesi morajo biti čim bolj avtomatizirani in izvedeni v doglednem času.
- Predlagani sistem ne sme bistveno posegati v delo arhitektov in gradbenikov.
- Omogočeno mora biti varno in trajno hranjenje posameznih modelov.

### **3.3.4 Pogledi na model**

Izhodišče predlaganega sistema predstavlja specifikacija nabora informacij, ki natančno opredeljuje entitete, ki predstavljajo predmet izmenjave. S tem bi podobno kot pri izmenjavi arhitekture (tam je definiran razširjen koordinacijski pogled) opredelili izmenjavo pri zaokroženem načrtovanju konstrukcij.

Pri določitvi pogledov na model bodo upoštevana priporočila organizacije IAI oz. skupine MVD. Zato bosta v nadaljevanju definirana dva ločena pogleda:

- pogled arhitektura – računsko modeliranje (pogled A-RM),
- pogled računsko modeliranje – računska analiza (pogled RM-RA).

#### **3.3.4.1 Pogled na model: Arhitektura – računsko modeliranje (načrtovanje) nosilne konstrukcije**

Razširjeni koordinacijski pogled, ki opredeljuje nabor informacij pri izmenjavi arhitekturnih modelov, ni povsem primeren za opis obravnavanega pogleda, saj ne vsebuje konceptov, primernih za računsko modeliranje konstrukcije ter za opis analize konstrukcije. V skladu z

uveljavljenim pristopom pri razvoju pogledov znotraj IFC specifikacije (IAI, 2008), FIOPE pristopa pri določitvi entitet izmenjave (Nour, 2007) ter po priporočilih relevantnih virov (Lehtinen, 2005) je smiselno aktualni pogled na model definirati skladno z načelom »uporabnega minimuma«: specificira se potreben nabor entitet, ki zadostuje za opis večine splošnih primerov. Izbran pristop omogoča razširljivost, ki se z dodajanjem entitet lahko izvede po potrditvi koncepta. V nadaljevanju uporabljeni koncepti se v življenjskem ciklu pogleda (osnutek, predlog, kandidat, uradno veljavni, nadomeščen) lahko smatrajo kot predlog pogleda. Le-ta je pripravljen na podlagi smernic delovne skupine za definicijo pogledov (IAI-MVD, 2007): generičnemu opisu pogleda sledi analiza zahtev pri izmenjavi. Po priporočilih organizacije IAI specifikacija pogledov na model poteka na dveh nivojih: na konceptualnem (neodvisnem od implementacije in celo od IFC specifikacije) ter na IFC specifičnem. Konceptualni nivo dejansko predstavlja opis zahtev pri izmenjavi relevantnih informacij, medtem ko je IFC specifični nivo bolj tehnično podroben (Lehtinen, Hietanen, 2007).

#### **3.3.4.1.1 Generični opis A-RM pogleda**

Razširjeni koordinacijski pogled vsebuje precej več informacij o zgradbi, kot jih pri svojem delu potrebuje gradbenik konstruktor, hkrati pa ne omogoča popolnega semantičnega opisa primitivov računskega modela. Zato je potrebno definirati nov pogled, ki bo omogočal izmenjavo informacij med aplikacijami za načrtovanje arhitekture ter aplikacijami za računsko modeliranje (domena analize konstrukcij ni zajeta v tem pogledu).

Izhodišče modeliranja predstavlja geometrija, ki jo v informacijski model zgradbe vnese arhitekt in ki se lahko preslika v primitive aplikacije za računsko modeliranje. Po najbolj naprednem scenariju je moč navedene preslikave avtomatizirati.

A-RM pogled naj bi po predlogu skupine IAI-MVD pokrival le informacije, ki jih ustvari arhitekt. Takšna predpostavka ustreza informacijskem toku, saj model izhaja iz aplikacije za arhitekturno načrtovanje.

Načrtovani pogled se od razširjenega koordinacijskega pogleda med drugim razlikuje tudi po:

- predpisani natančni opredelitvi razmerij med primitivi (združevanje elementov v sklope – npr. streha, etaža, prostor, ipd.),

- predpisani definiciji materialov za vsak element,
- predpisanih dodatnih lastnosti (Pset), ki dopolnjujejo nepopolno semantiko elementov zgradb (nosilno-nenosilno, zunanje-notranje, ipd.).

Semantika modela zgradb omogoča precej več kot le izmenjavo geometrijskih entitet. S povezavo objektov IFC modela ter z vključitvijo standardov s področja analize konstrukcij (Evrokod) lahko deloma avtomatiziramo tudi določitev zunanjih vplivov na konstrukcijo.

Uporaba navedenega pogleda je v skladu s trenutnim dosegom omejena na objekte visokogradnje, ni pa omejena z načinom gradnje oz. z uporabljenimi materiali.

#### **3.3.4.1.2 Konceptualni opis**

S konceptualnim zapisom je možno na razumljiv način opisati informacije, ki predstavljajo predmet zanimanja posameznega pogleda. V obravnavanem primeru je smiselno obravnavati oba konca informacijskega kontinuuma ločeno: arhitekturo ter računsko modeliranje konstrukcij.

Skladno z osnovno idejo informacijskega modeliranja je najprej potrebno opisati strukturo prostorskih elementov ter relacij med njimi:

- projekt (kontekst informacij, ki se izmenjujejo),
- gradbišče (območje, kjer se bo odvijala gradnja),
- zgradba (načrtovani objekt),
- etaža (horizontalna agregacija prostorov na določeni višini),
- prostor (smiselno zaključen volumen dela zgradbe).

Za zadovoljivo natančen opis objektov visokogradnje je v opisu za računsko modeliranje potrebno zajeti vsaj naslednje primitive:

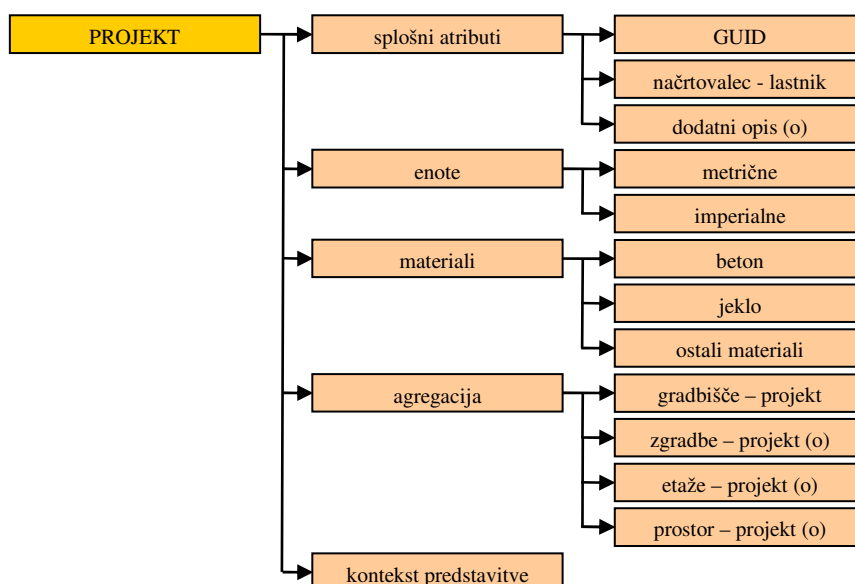
- steber (vertikalni oz. skoraj vertikalni ne nujno nosilni linijski element),
- prečka (horizontalni oz. skoraj horizontalni ne nujno nosilni linijski element),
- element (splošen, ne nujno nosilen linijski element),

- plošča (splošen ploskovni element s konstantno debelino in poljubno lego (ne nujno horizontalno), ki običajno vertikalno zapira prostor),
- stena (vertikalni ne nujno nosilen ploskoven element s funkcijo ločevanja prostorov),
- kritje konstrukcijskih elementov (npr. plošče, stene ipd.),
- odprtine odprtih v ploskovnih elementih,
- streha (vsebovalnik, ki združuje vse elemente ostrešja),
- klančina (gradniki v nagibu, ki omogočajo komunikacijo med prostori. Klančina lahko zajema vmesno ploščo, ne pa tudi stopnišča. Rama klančine predstavlja posamezen segment klančine),
- stopnišče (gradnik vertikalne komunikacije. Stopnišče lahko zajema tudi vmesno ploščo – podest. Stopniščna rama predstavlja segment stopnišča),
- proksi elementi kot vsebovalniki morebitnih dodatno definiranih objektov,
- odprtine v elementih zgradb,
- povezovalni elementi (vsebovalnik primitivov, ki povezujejo zgoraj navedene elemente zgradb),
- elementi transporta (vsebovalniki primitivov, namenjenih transportu ljudi ali dobrin v zgradbi),
- navidezni prostori (elementi, namenjeni navidezni ločitvi primitivov),
- cone (združevanje (navideznih) prostorov),
- nabor lastnosti deljenih elementov (Pset\_BeamCommon, Pset\_ColumnCommon, Pset\_CurtainWallCommon, Pset\_RampCommon, Pset\_RampFlightCommon, Pset\_RoofCommon, Pset\_SlabCommon, Pset\_StairCommon, Pset\_StairFlightCommon, Pset\_WallCommon).

Materiali in koncepti geometrijske predstavitve ne bodo podrobneje obravnavani, saj se je na predstavljenih področjih smiselno nasloniti na predlagane omejitve razširjenega koordinacijskega pogleda.

V nadaljevanju bodo s predpisano grafiko (IAI-MVD, 2007) predstavljene konkretne entitete, ki predstavljajo uporabni minimum A-RM pogleda (slike 20 – 27). Ob vsakem diagramu bo skladno z EXPRESS shemo podan tudi primer zapisa osnovnega gradnika v fizični STEP datoteki (zaradi obsega reference entitet ne bodo eksplicitno navedene). S krepkim tiskom so označene po specifikaciji obvezne entitete, s krepkim ležečim pa dodatne obvezne entitete, ki jih predpisuje obravnavani A-RM pogled. Svetli tisk označuje vrsto atributa (npr. atribut »description« opišemo z entiteto IfcText).

### PROJEKT (IfcProject)



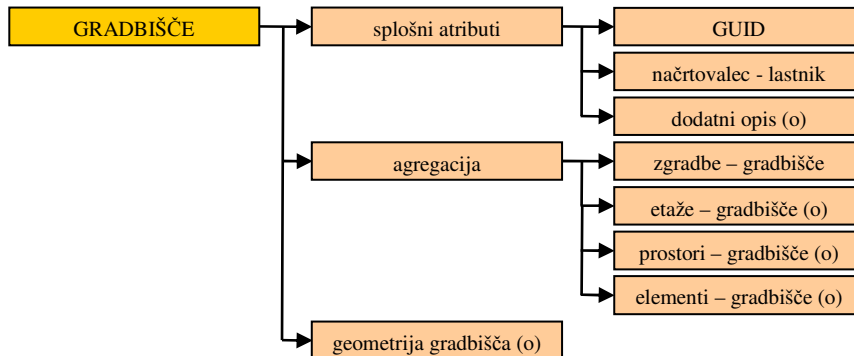
```
IfcProject (GlobalID-IfcGloballyUniqueId, OwnerHistory-IfcOwnerHistory,  
Name-IfcLabel, Description-IfcText, ObjectType-IfcLabel, LongName-IfcLabel,  
Phase-IfcLabel, RepresentationContexts-IfcRepresentationContext,  
UnitsInContext-IfcUnitAssignment);
```

Slika 20: A-RM pogled: IfcProject

Figure 20: A-RM view: IfcProject



## GRADBIŠČE (IfcSite)

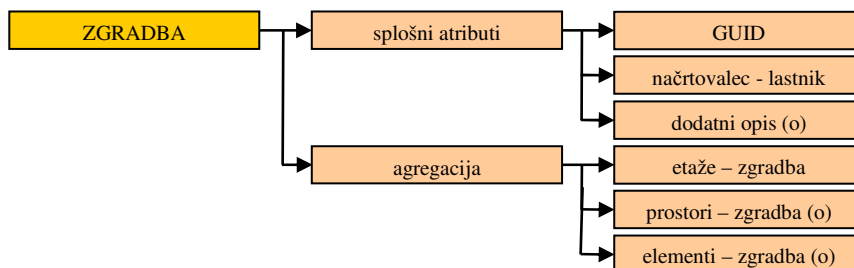


IfcSite(**GlobalID**-IfcGloballyUniqueId, **OwnerHistory**-IfcOwnerHistory, Name-IfcLabel, Description-IfcText, ObjectType-IfcLabel, ObjectPlacement-IfcObjectPlacement, Representation-IfcProductRepresentation, LongName-IfcLabel, **CompositionType**-IfcElementCompositionEnum, RefLatitude-IfcCompoundPlaneAngleMeasure, RefLongitude-IfcCompoundPlaneAngleMeasure, RefElevation-IfcLengthMeasure, LandTitleNumber-IfcLabel, SiteAddress-IfcPostalAddress);

Slika 21: A-RM pogled: IfcSite

Figure 21: A-RM view: IfcSite

## ZGRADBA (IfcBuilding)

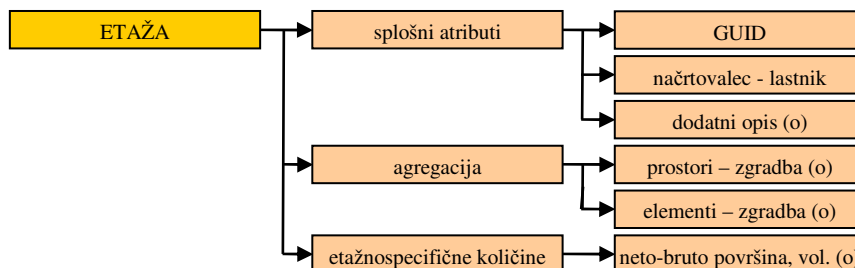


IfcBuilding(**GlobalID**-IfcGloballyUniqueId, **OwnerHistory**-IfcOwnerHistory, Name-IfcLabel, Description-IfcText, ObjectType-IfcLabel, ObjectPlacement-IfcObjectPlacement, Representation-IfcProductRepresentation, LongName-IfcLabel, **CompositionType**-IfcElementCompositionEnum, ElevationOfRefHeight-IfcLengthMeasure, ElevationOfTerrain-IfcLengthMeasure, BuildingAddress-IfcPostalAddress);

Slika 22: A-RM pogled: IfcBuilding

Figure 22: A-RM view: IfcBuilding

## ETAŽA (IfcBuildingStorey)

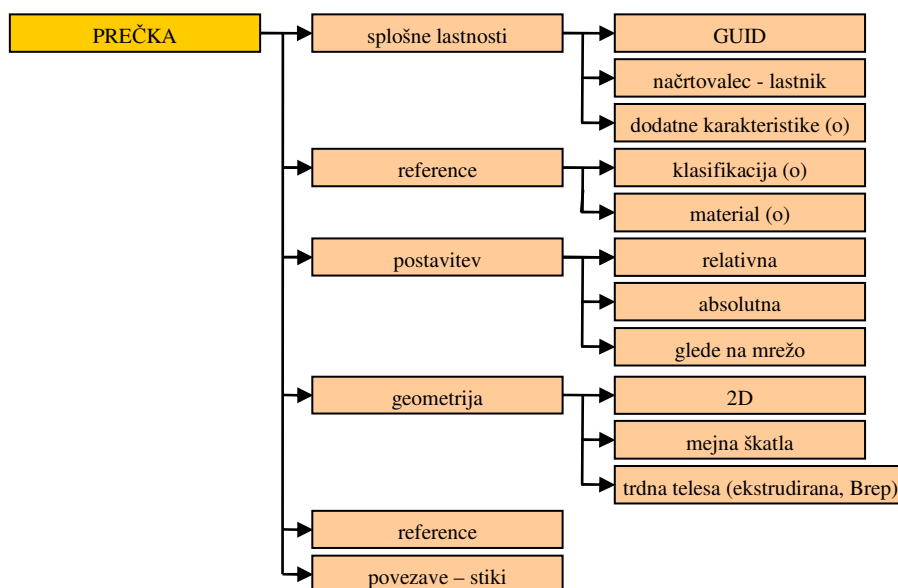


```
IfcBuildingStorey(GlobalID-IfcGloballyUniqueId, OwnerHistory-  
IfcOwnerHistory, Name-IfcLabel, Description-IfcText, ObjectType-IfcLabel,  
ObjectPlacement-IfcObjectPlacement, Representation-  
IfcProductRepresentation, LongName-IfcLabel, CompositionType-  
IfcElementCompositionEnum, Elevation-IfcLengthMeasure);
```

Slika 23: A-RM pogled: IfcBuildingStorey

Figure 23: A-RM view: IfcBuildingStorey

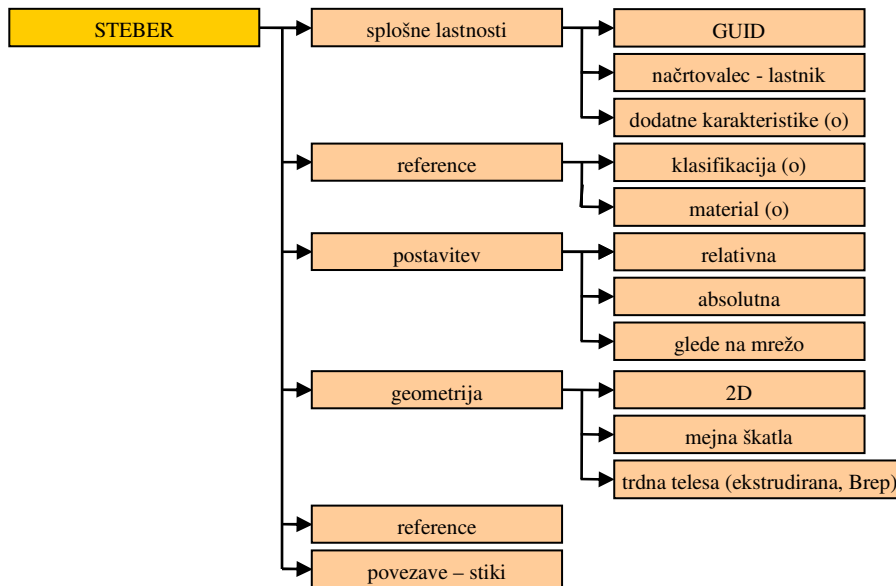
## PREČKA (IfcBeam)



```
IfcBeam(GlobalID-IfcGloballyUniqueId, OwnerHistory-IfcOwnerHistory, Name-  
IfcLabel, Description-IfcText, ObjectType-IfcLabel, ObjectPlacement-  
IfcObjectPlacement, Representation-IfcProductRepresentation, Tag-  
IfcIdentifier);
```

Slika 24: A-RM pogled: IfcBeam

Figure 24: A-RM view: IfcBeam

**STEBER (IfcColumn)**

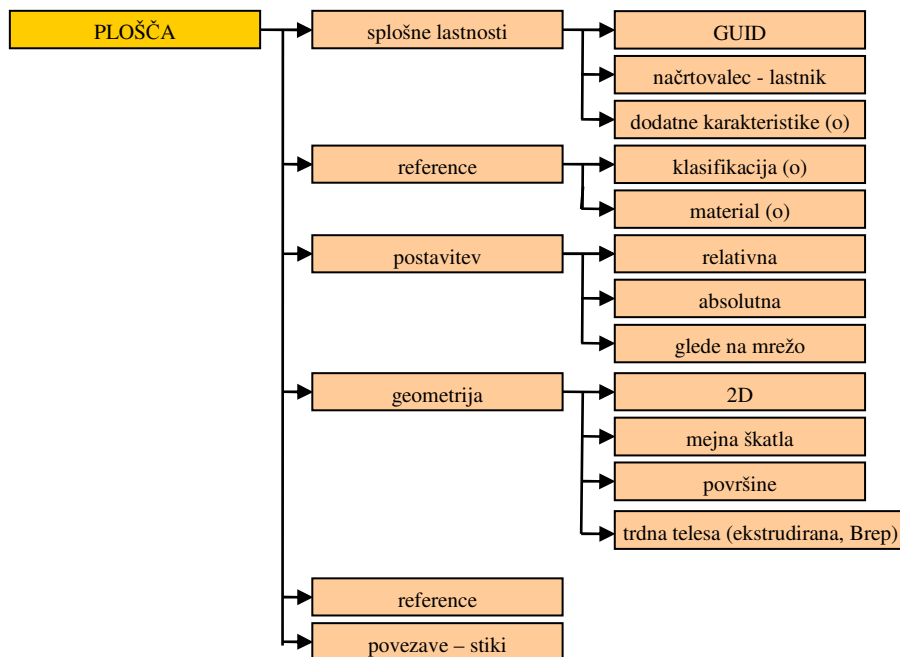
```

IfcColumn(GlobalID-IfcGloballyUniqueId, OwnerHistory-IfcOwnerHistory, Name-
IfcLabel, Description-IfcText, ObjectType-IfcLabel, ObjectPlacement-
IfcObjectPlacement, Representation-IfcProductRepresentation, Tag-
IfcIdentifier);
  
```

Slika 25: A-RM pogled: IfcColumn

Figure 25: A-RM view: IfcColumn

## PLOŠČA (IfcSlab)

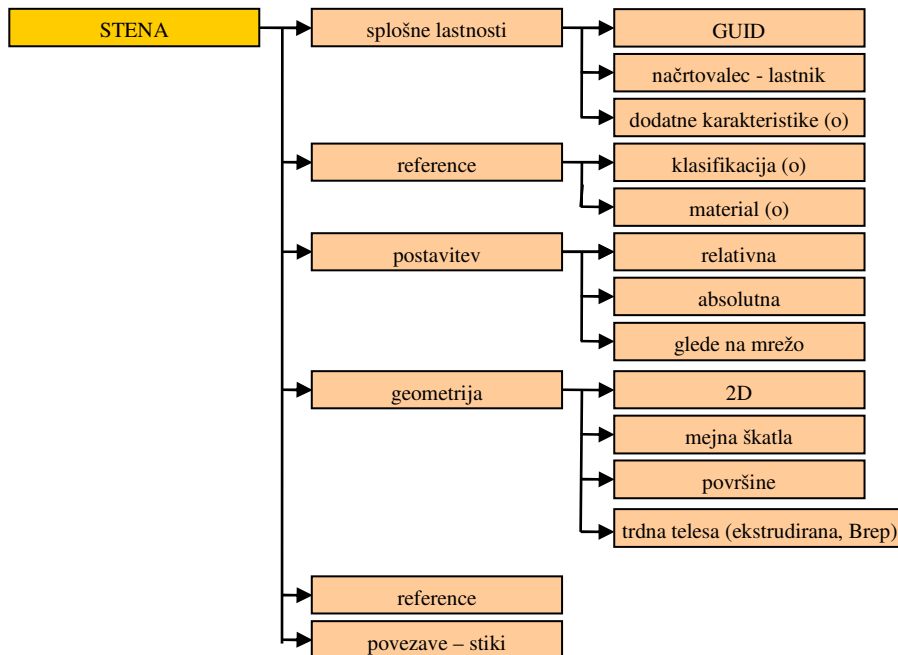


```
IfcSlab(GlobalID-IfcGloballyUniqueId, OwnerHistory-IfcOwnerHistory, Name-  
IfcLabel, Description-IfcText, ObjectType-IfcLabel, ObjectPlacement-  
IfcObjectPlacement, Representation-IfcProductRepresentation, Tag-  
IfcIdentifier, PredefinedType-IfcSlabTypeEnum);
```

Slika 26: A-RM pogled: IfcSlab

Figure 26: A-RM view: IfcSlab

## STENA (IfcWall)



```

IfcWall(GlobalID-IfcGloballyUniqueId, OwnerHistory-IfcOwnerHistory, Name-
IfcLabel, Description-IfcText, ObjectType-IfcLabel, ObjectPlacement-
IfcObjectPlacement, Representation-IfcProductRepresentation, Tag-
IfcIdentifier);
  
```

Slika 27: A-RM pogled: IfcWall

Figure 27: A-RM view: IfcWall

## PSET\_ImeElementa

Nabor lastnosti zgoraj predstavljenih deljenih elementov (Pset\_ImeElementa) prav tako uvrščamo med uporabni minimum pogleda A-RM. Z navedenim naborom je namreč možno identificirati, določiti nosilne elemente.

Primer:

```

Pset_BeamCommon(Reference-IfcIdentifier, Span-IfcPositiveLengthMeasure,
Slope-IfcPlaneAngleMeasure, IsExternal-IfcBoolean, LoadBearing-IfcBoolean,
FireRating-IfcLabel);
  
```

### **3.3.4.2 Pogled na model: Računsko modeliranje (načrtovanje) nosilne konstrukcije – Analiza konstrukcije**

Navedeni pogled sledi delotoku pri načrtovanju konstrukcij. Gradbenik konstruktor mora po izdelavi podrobnega računskega modela slednjega prenesti v aplikacijo za računsko analizo konstrukcij.

#### **3.3.4.2.1 Konceptualni opis**

V splošnem primeru lahko informacije, potrebne za opis računskega modela, razdelimo v pet konceptualnih skupin:

- Geometrija konstrukcije:
  - mreža oz. raster za načrtovanje,
  - linijski elementi (oznaka elementa, lega, sprostitev, segmenti, prerez),
  - ploskovni elementi (oznaka elementa, lega, prerez),
  - vozlišča (ime, lokacija, tip),
  - podpore (ime lokacija, tip).
- Prerezi elementov:
  - lastnosti prereza: oznaka, prečni prerez, torzijska konstanta, odpornostni moment (os 2, 3), strižni prerez (os 2, 3), ipd.
- Materiali:
  - ime, vrsta (izotropični, ortotropični, anizotropični),
  - specifična masa, teža, modul elastičnosti, Poissonov koeficient, strižni modul, koeficient termičnega raztezka.
- Obtežba:
  - vrsta obtežbe: posplošene sile, posplošeni premiki, enakomerna sprememba temperature, neenakomerna sprememba temperature ipd.,
  - vozliščna obtežba (ime, vrsta, smer, velikost),
  - obtežba linijskih elementov (ime, vrsta, smer, velikost),

- obtežba ploskovnih elementov (ime, vrsta, smer, velikost),
- obtežni primeri (ime, vrsta obtežbe, tip (stalna koristna, sneg, veter, potres, ostalo), faktorji),
- obtežne kombinacije (ime, vrsta, faktorji).
- Ostale informacije:
  - med ostale informacije zaradi odsotnosti modela za analizo lahko uvrstimo količine, ki opisujejo rezultate analize (npr. za posplošene sile in posplošene premike, napetosti, armatura). Opis navedenih količin je izven dosega te naloge.

### 3.3.4.3 Splošne zahteve pri specifikaciji računskega podmodela

IFC specifikacija in pripadajoča dokumentacija (implementacijski vodnik, razširitev ST4) podajata nekaj pravil oz. omejitev, ki jih je potrebno upoštevati pri implementaciji:

- prisotna mora biti vsaj ena entiteta IfcProject – projekt (1), IfcSite – gradbišče (0...n), IfcBuilding (modelirane zgradbe), IfcBuildingStories (etaže modelirane zgradbe);
- gradniki računskega modela morajo biti zbrani v vsaj enem modelu za analizo IfcStructuralAnalysisModel;
- opredeljena mora biti relacija med IfcBuilding in IfcStructuralAnalysisModel;
- pri opisu gradnikov (IfcStructuralItems) in obtežbe (IfcStructuralActivities) mora biti uporabljen en koordinatni sistem (omejitev velja znotraj posameznega modela, opredeljenega z IfcStructuralAnalysisModel);
- prisotna mora biti vsaj ena entiteta IfcGeometricRepresentationContext (v splošnem je znotraj IFC modela lahko uporabljenih več predstavitev oblik);
- povezave med gradniki morajo biti definirane kot podpore ali vozlišča. Vsaka podpora oz. vozlišče morata biti povezana vsaj z enim elementom (nepovezana vozlišča niso dovoljena);
- vsi objekti, ki se stikajo v istem vozlišču, morajo uporabljati isto instanco entitete IfcVertexPoint. Enako velja tudi za stike robov (IfcEdge);

- vplivi na konstrukcijo (IfcStructuralAction) morajo biti združeni v vsaj eno obtežno skupino (IfcStructuralLoadGroup);
- izmed vseh obtežb, ki delujejo na konstrukcijo (ki jih predstavlja abstraktni razred IfcStructuralLoad), mora biti natančno opredeljena vsaj ena (IfcStructuralAction).

## 3.4 Prirejanje geometrije

### 3.4.1 Hevristika

Slovar slovenskega knjižnega jezika označuje hevrstiko kot nauk o metodah raziskovanja in pridobivanja novih spoznanj. Torej je hevrstika v splošnem zelo širok pojem in ga je težko natančno definirati. V splošnem bi ga lahko označili kot način reševanja problemov, ki se izogiba očitnim napakam. V gradbeništvu je večina praktičnih računskih problemov poznanih že vrsto let in jih zato enostavno označimo kot problem intuitivnega reševanja »na recept z upoštevanjem zdrave pameti«.

Kot ekstremne primere hevrstike nekateri avtorji (Kos, 2000) opisujejo ekspertne sisteme, ki rešujejo probleme na bazi znanja, ki izkazuje znanje strokovnjakov. Tak pristop teži k temu, da bi bilo reševanje vsaj tako dobro, kot to delajo strokovnjaki, rezultati pa naj bi bili boljši od rezultatov strokovnjakov.

V prejšnjem podpoglavju je bilo predstavljenih več različnih hevrstičnih pristopov pri pretvorbi arhitekturnega modela v računski model. Implementacijo algoritmov preslikav, ki se z omejitvijo njihovega dosega ter s podajanjem dodatnih parametrov omejuje na specifične primere konstrukcij, lahko označimo za računalniško kodiranje hevrstičnih pravil.

Hevrstične metode prirejanja v veliki večini temeljijo na enostavnem principu delitve informacij na osnovne in kompleksne. Slednje lahko s pomočjo preslikav in upoštevanjem dedovanja izpeljemo iz enostavnih. V obravnavanem primeru enostavne informacije predstavljajo geometrijski gradniki elementov zgradb, informacije o materialu, kompleksne pa topološke informacije, mreže končnih elementov, ipd. Le s tako definiranimi modeli je mogoče zagotoviti konsistenco in popolnost modelov.



### 3.4.2 Algoritem preslikav med arhitekturnimi modeli in modeli za analizo

Vsem neabstraktnim gradnikom arhitekturnega modela sta lahko pripisana dva atributa: postavitev gradnika ter njegova geometrijska predstavitev.

Standard STEP oz. v IFC-je privzeti del načeloma opredeljuje več možnih predstavitev geometrije (glej poglavje 2.3.2.1.2). Pri analizi IFC vmesnikov aplikacij za načrtovanje arhitekture je bilo ugotovljeno, da se za predstavitev geometrije nosilnih elementov zgradbe (stebri, prečke, stene, plošče) uporabljajo naslednji načini predstavitev: robna predstavitev (Brep), širjena telesa (SweptSolid), CSG trdna telesa (Constructive Solid Geometry) in mejna škatla (BoundingBox). Predstavitev gradnikov z mejnimi škatlami je praviloma namenjena samostojni predstavitvi geometrije objekta (samostojni arhitekturni podmodel v modelu zgradb IFC). Pri analizi vmesnikov je bilo dodatno ugotovljeno, da praktično implementirana predstavitev linijskih elementov temelji na robni predstavitvi in širjenih telesih.

Nasprotno pa predstavitev elementov računskega modela (kot kompleksni primer) temelji na topološki predstavitvi. To pomeni, da se gradniki predstavitve delijo na geometrijski del z geometrijskimi elementi (površina – »surface«, krivulja – »curve« in točka – »point«) in na topološki del s topološkimi elementi (ploskev – »face«, rob – »edge«, vozlišče – »vertex«).

Preglednica 10: Možna stikovanja med elementi zgradb

Table 10: Possible connections between building elements

steber	steber	točkovni
steber	prečka	linijski
steber	plošča	točkovni
steber	stena	linijski
prečka	prečka	točkovni
prečka	plošča	linijski
prečka	stena	linijski
stena	stena	linijski
stena	plošča	linijski
plošča	plošča	linijski

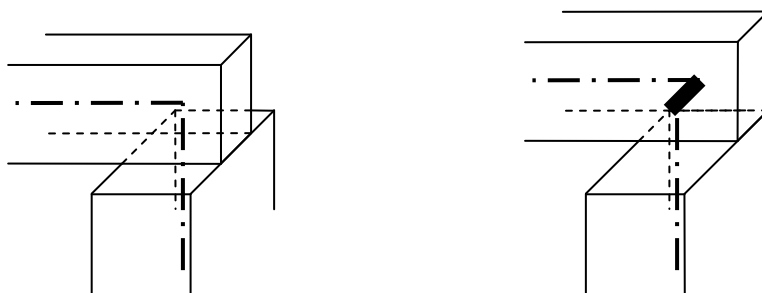
Opis stebra predstavlja najenostavnejši primer: v računskem modelu je steber predstavljen kot linijski segment, ki leži v osi geometrijske osi in ki povezuje dvoje vozlišč. Vozlišča določajo nove točke, izpeljane iz točk, ki definirajo geometrijski model.

Določevanje točkovnih stikov elementov (preglednica 10) iz tako definiranih gradnikov računskih modelov je možno samo v primeru, če osi obeh elementov ležijo v ravnini, ki je vzporedna obema elementoma (slika 28). Ker temu v splošnem ni tako, je potrebno povezanost elementov ugotavljati na nivoju elementov zgradb.

Nguyen in Oloufa (2002) klasificirata medsebojne relacije elementov zgradb v pet kategorij:

- elementa sta sosednja,
- elementa sta nepovezana,
- prvi element leži v drugem elementu (in obratno),
- prvi element seka drugega (in obratno),
- prvi element je povezan z drugim (in obratno).

Izmed navedenih relacij sta za določitev stikov elementov zgradb zanimivi le zadnji razmerji, pri čemer lahko presek smatramo za posebno obliko povezave elementov.



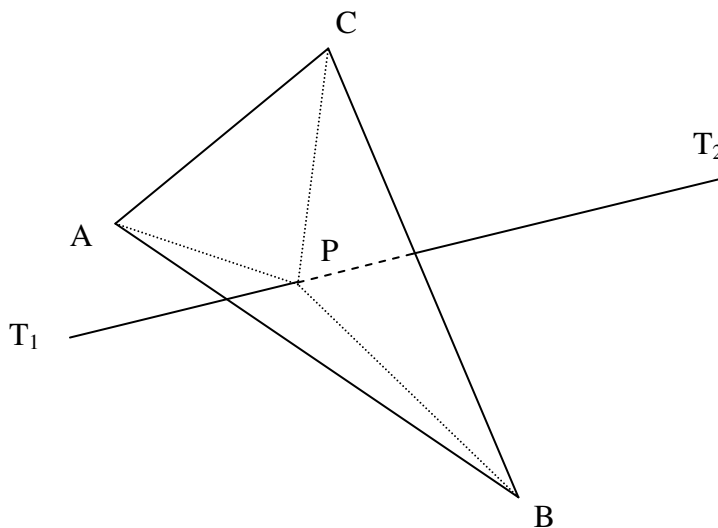
Slika 28: Stikovanje elementov

Figure 28: Mechanical connectivity

Ugotavljanje povezave dveh prizem temelji na dejstvu, da v primeru povezave med prizmama rob ene prizme leži oz. seka površino druge prizme. Geometrijsko se predstavljeni problem prevede na iskanje presečišča daljice (robu) s presečiščem ravnine, ki jo v splošnem definirajo

točke, ki omejujejo izbrano ploskev geometrijskega telesa. Ravnino sicer v splošnem določajo tri točke, zato je potrebno mejne ploskve ustrezno triangulirati. Nadaljnji algoritem poteka po naslednjem postopku:

- določi se enačba ravnine, ki jo določajo vozlišča A, B in C;
- določi se enačba premice, ki jo določata točki T1 in T2;
- če je koeficient premice enak 0, potem je premica vzporedna ravnini (ni presečišča). Če je koeficient med 0 in 1, potem presečišče premice in ravnine obstaja, vendar ni nujno, da obstaja v trikotniku ABC. Slednje se najenostavneje preverja s ploščinami trikotnikov: če je vsota ploščin trikotnikov ABP, ACP in BCP enaka ploščini trikotnika ABC, potem se elementa stikata.



Slika 29: Algoritem za ugotavljanje stikov

Figure 29: Connectivity algorithm

Predstavljeni algoritem določa medsebojno lego elementov v točkovnem stiku. Za splošen primer definicije linijskega stika bi bilo potrebno algoritem dopolniti.

Za okvirne konstrukcije s ploščami je globalno zato predlagan naslednji algoritem:

- določitev povezav na nivoju elementov zgradb (na podlagi predlaganega algoritma);

- določitev nivoja etaž (srednja višina plošče) in medetaž ter preslikava plošč v ploskovne elemente modela za analizo;
- preslikava stebrov v računske linijske elemente (glede na nivoje etaž);
- preslikava prečk v računske linijske elemente;
- določitev povezav med vozlišči računskega modela.

## **3.5 Predlagana rešitev zaokroženega načrtovanja znotraj modela IFC**

### **3.5.1 Temeljni principi**

Z opredelitvijo algoritma preslikav med arhitekturnimi modeli in modeli za analizo so bili postavljeni temelji za zaokroženo načrtovanje (roundtrip engineering) na podlagi informacijskega modela zgradb IFC. Zaokroženo načrtovanje predstavlja relativno dobro uveljavljen termin v industriji, ki oblikuje grajeno okolje. Pri omejitvi na domeno arhitekture in računske analize konstrukcij navedeni termin označuje iterativno delo arhitekta ter gradbenika konstruktorja.

Obstoječe rešitve sodobne programske opreme že nekaj let ponujajo dvosmerno delujoče vmesnike domensko specifičnih aplikacij, ki olajšajo delo pri predstavljenem iterativnem sodelovanju. Slabost navedenih vmesnikov predstavlja omejenost na specifično programsko opremo (običajno posledica kapitalskih povezav) ter njihova neprilagodljivost. Čeprav bi primerljivo rešitev dosegli z uporabo IFC specifikacije ter ustrezno prilagodljivih aplikacijsko specifičnih vmesnikov, je v tej nalogi uporabljena sodobnejša rešitev. Iterativno načrtovanje ne poteka več na podlagi izmenjave datotek, temveč je delo koordinirano s pomočjo spletnega strežnika IFC. Korektno uporabo IFC strežnika priporoča večina najnovejših raziskav (npr. Kiviniemi et al., 2006) rešuje večje število pomanjkljivosti datotečne izmenjave:

- spremljanje razvoja modela (sposobnost upravljanja z različicami),
- zagotavljanje integritete, kompatibilnosti in persistence,
- definirano je lastništvo informacij,

- upravljanje z različnimi vlogami akterjev na projektu,
- možno je opredeliti različne nivoje dostopa glede na vloge akterjev,
- notifikacija akterjev o spremembah modela.

### 3.5.2 Opredelitev problema

Dokumentacija projekta ST4 (Weise et al., 2002) predpisuje vrsto informacij, ki jih mora vsebovati iz arhitekturnega modela izpeljan računski model:

- idealizirane računske elemente,
- vozlišča oz. povezave med elementi ter podpore,
- mehanske lastnosti,
- obtežbe,
- ostale informacije.

Vse dosedanje raziskave s področja prirejanja ((Chen et al., 2005), (Deng, Chang, 2006), (Romberg et al., 2004)) so temeljile na predpostavki, da je iz arhitekturnega modela, zapisanega v formatu IFC, mogoče izluščiti le idealizirane elemente računskega modela ter deloma tudi povezave med njimi. Navedena predpostavka je seveda napačna. IFC arhitekturni model zgradbe namreč vsebuje precej več informacij o zgradbi (npr: namembnost prostorov, sestava sten oz. fasade, sestava medetažne konstrukcije, sestava strehe ipd.), ki jih je mogoče koristno uporabiti pri generaciji računskega modela.

Ta naloga se poleg prirejanja geometrije osredotoča tudi na prirejanje vplivov na konstrukcijo, ki jih predpisuje standard Evrokod (ENV 1991-1, 2004). Enega izmed najbolj očitnih primerov, ki bo uporabljen tudi pri implementaciji prototipa, predstavlja avtomatsko prirejanje koristne obtežbe etaž (prirejanje temelji na definiciji prostorov oz. con znotraj zgradbe).

IFC arhitekturni model kot izhodišče raziskave je lahko poljubno natančen, običajno pa njegovo natančnost določa razširjeni koordinacijski pogled. Vendar pa minimum entitet navedenega pogleda, ki je praviloma implementiran v IFC vmesnikih, zadostuje le za izmenjavo geometrije. Sklicevanje na semantično vsebino modela zahteva natančno

opredelitev razmerij med primitivi arhitekturnega modela, ki so nujno potrebna za opredelitev obtežbe na konstrukcijo:

- določena morajo biti razmerja med prostori in elementi zgradb (prostor – stene, plošče, stebri, prečke),
- definirana mora biti natančna sestava konstrukcijskih sklopov (medetažne konstrukcije, stene, strehe),
- opredeljena mora biti geografska lega konstrukcije,
- stene (vsaj zunanje) in strehe morajo biti orientirane,
- opredeljeni morajo biti vsi materiali (za določitev materialnih karakteristik potrebnih v modelu za analizo).

Izmed navedenih razmerij so le redka implementirana v IFC vmesnikih aplikacij za arhitekturno modeliranje. Ker na razvoj vmesnikov neposredno ni mogoče vplivati, bo izdelava prototipa ter s tem potrditev koncepta omejena na edino implementirano razmerje, odkrito v fazi testiranja (razmerje med namembnostjo prostorov ter pripadajočimi elementi zgradb).

### **3.5.2.1 Vplivi na konstrukcijo**

Vplive na konstrukcijo (pogovorno običajno govorimo kar o obremenitvi konstrukcije) predstavljajo njihova velikost, razpored po konstrukciji in vrsta – način delovanja (Duhovnik, 1998). Vplive na konstrukcijo delimo na stalne in spremenljive. Glede na izvor vplive delimo na:

- mehansko obtežbo (razporejena ali koncentrirana, pomična ali nepomična, posredna ali neposredna),
- spremembo temperature (enakomerna ali neenakomerna), s spremembo temperature se opisuje tudi krčenje materiala,
- premike podpor.

Najbolj pogosti vplivi na konstrukcijo so določeni s statističnimi metodami in zbrani v predpisih. Trenutno veljavne predpise, ki določajo vplive na konstrukcijo, predstavlja družina standardov SIST ENV 1991 (preglednica 11).

## Preglednica 11: Evrokod 1: Vplivi na konstrukcijo

Table 11: Eurocode 1: Actions on structures

oznaka	naziv
SIST EN 1991-1-1:2004	Evrokod 1: Vplivi na konstrukcije – 1-1. del: Splošni vplivi – Gostote, lastna teža, koristne obtežbe stavb
SIST EN 1991-1-1:2004	Evrokod 1: Vplivi na konstrukcije – 1-1. del: Splošni vplivi – Prostorninske teže, lastna teža, koristne obtežbe stavb
SIST EN 1991-1-1:2004/A101:2005	Evrokod 1: Vplivi na konstrukcije – 1-1. del: Splošni vplivi – Prostorninske teže, lastna teža, koristne obtežbe stavb – Nacionalni dodatek
SIST EN 1991-1-2:2004	Evrokod 1: Vplivi na konstrukcije – 1-2. del: Splošni vplivi – Vplivi požara na konstrukcije
SIST EN 1991-1-2:2004	Evrokod 1: Vplivi na konstrukcije – 1-2. del: Splošni vplivi – Vplivi požara na konstrukcije
SIST EN 1991-1-2:2004/A101:2006	Evrokod 1: Vplivi na konstrukcije – 1-2. del: Splošni vplivi – Vplivi požara na konstrukcije – Nacionalni dodatek
SIST EN 1991-1-3:2004/oA101:2007	Evrokod 1: Vplivi na konstrukcije – 1-3. del: Splošni vplivi - Obtežba snega - Nacionalni dodatek
SIST EN 1991-1-3:2004	Evrokod 1: Vplivi na konstrukcije – 1-3. del: Splošni vplivi – Obtežba snega
SIST EN 1991-1-3:2004	Evrokod 1: Vplivi na konstrukcije – 1-3. del: Splošni vplivi – Obtežba snega
SIST EN 1991-1-3:2004/A101:2008	Evrokod 1: Vplivi na konstrukcije – 1-3. del: Splošni vplivi - Obtežba snega - Nacionalni dodatek
SIST EN 1991-1-4:2005	Evrokod 1: Vplivi na konstrukcije – 1-4. del: Splošni vplivi – Obtežbe vetra
SIST EN 1991-1-4:2005	Evrokod 1: Vplivi na konstrukcije – 1-4. del: Splošni vplivi – Obtežbe vetra
SIST EN 1991-1-4:2005/A101:2008	Evrokod 1: Vplivi na konstrukcije – 1-4. del: Splošni vplivi - Obtežbe vetra - Nacionalni dodatek
SIST EN 1991-1-4:2005/oA101:2007	Evrokod 1: Vplivi na konstrukcije – 1-4. del: Splošni vplivi – Obtežbe vetra- Nacionalni dodatek
SIST EN 1991-1-5:2004	Evrokod 1: Vplivi na konstrukcije – 1-5. del: Splošni vplivi – Toplotni vplivi
SIST EN 1991-1-6:2005	Evrokod 1: Vplivi na konstrukcije – 1-6. del: Splošni vplivi – Vplivi med gradnjo

se nadaljuje

nadaljevanje

SIST EN 1991-1-7:2006	Evrokod 1: Vplivi na konstrukcije – 1-7. del: Splošna pravila – Nezagodni vplivi
SIST EN 1991-2:2004	Evrokod 1: Osnove projektiranja in vplivi na konstrukcije – 2. del: Prometna obtežba mostov
SIST EN 1991-3:2006	Evrokod 1: Vplivi na konstrukcije – 3. del: Vpliv žerjavov in drugih strojev
SIST EN 1991-4:2006	Evrokod 1: Vplivi na konstrukcije – 4. del: Silosi in rezervoarji
SIST ISO 1991-1:1996	Zelenjava - Nomenklatura - Prvi spisek
SIST ISO 1991-2:1996	Zelenjava - Nomenklatura - Drugi spisek

### 3.5.2.2 ENV 1991-1-1

ENV 1991-2-1 (ENV 1991, 2004) predpisuje splošne vplive, ki jih mora gradbenik konstruktor upoštevati pri računski analizi konstrukcij. V prvem delu so predpisane karakteristične vrednosti za oceno lastne teže gradbenih elementov. Le-te se v IFC model vključujejo preko materialnih lastnosti.

V drugem delu standarda so predpisane karakteristične vrednosti, koristne obtežbe za stanovanja, pisarne, javne prostore, garaže in prostore, kjer se gibljejo vozila, skladišča in industrijske prostore ter strehe.

Kombinacije vplivov na konstrukcijo niso predmet standarda ENV 1991-1-1. Ker predlagani sistem za polavtomatsko pretvorbo obravnava le nekatere izmed možnih vplivov, določevanje njihovih kombinacij z upoštevanjem ugodnega/neugodnega vpliva ni smiselno in v nalogi tudi ni posebej obravnavano.

### 3.5.2.3 Transformacija ENV 1991-1-1 v XML obliko

XML format predstavlja najprimernejšo obliko transformacije standarda ENV 1991-1-1 v računalniško berljiv zapis.

Za vsako vrsto vpliva na konstrukcijo znotraj posameznega Evrokod standarda je smiselno predlagati svojo XML datoteko. Le takšen pristop rezultira v obvladljive XML datoteke. Slika



30 prikazuje primer XML zapisa za koristno obtežbo. Žal pretvorba ostalih obtežb (npr. vetra, snega) ni tako enostavna.

```
<?xml version="1.0"?>
<DATA>
<EC loads - simple example>
  <category ID="A">
    <subcategory ID_S=A1>
      <load>2.0</load>
    </subcategory>
    <subcategory ID_S=A2>
      <load>2.0</load>
    </subcategory>
    <subcategory ID_S=A3>
      <load>2.5</load>
    </subcategory>
  </category>
  <category ID="B">
    <subcategory ID_S=B1>
      <load>3.0</load>
    </subcategory>
  </category>
  <category ID="C">
    <subcategory ID_S=C1>
      <load>3.0</load>
    </subcategory>
    <subcategory ID_S=C2>
      <load>4.0</load>
    </subcategory>
    <subcategory ID_S=C3>
      <load>5.0</load>
    </subcategory>
    <subcategory ID_S=C4>
      <load>5.0</load>
    </subcategory>
    <subcategory ID_S=C5>
      <load>5.0</load>
    </subcategory>
  </category>
  <category ID="D">
    <subcategory ID_S=D1>
      <load>4.0</load>
    </subcategory>
    <subcategory ID_S=D2>
      <load>5.0</load>
    </subcategory>
  </category>
</EC loads - simple example>
</DATA>
```

Slika 30: XML zapis koristne obtežbe

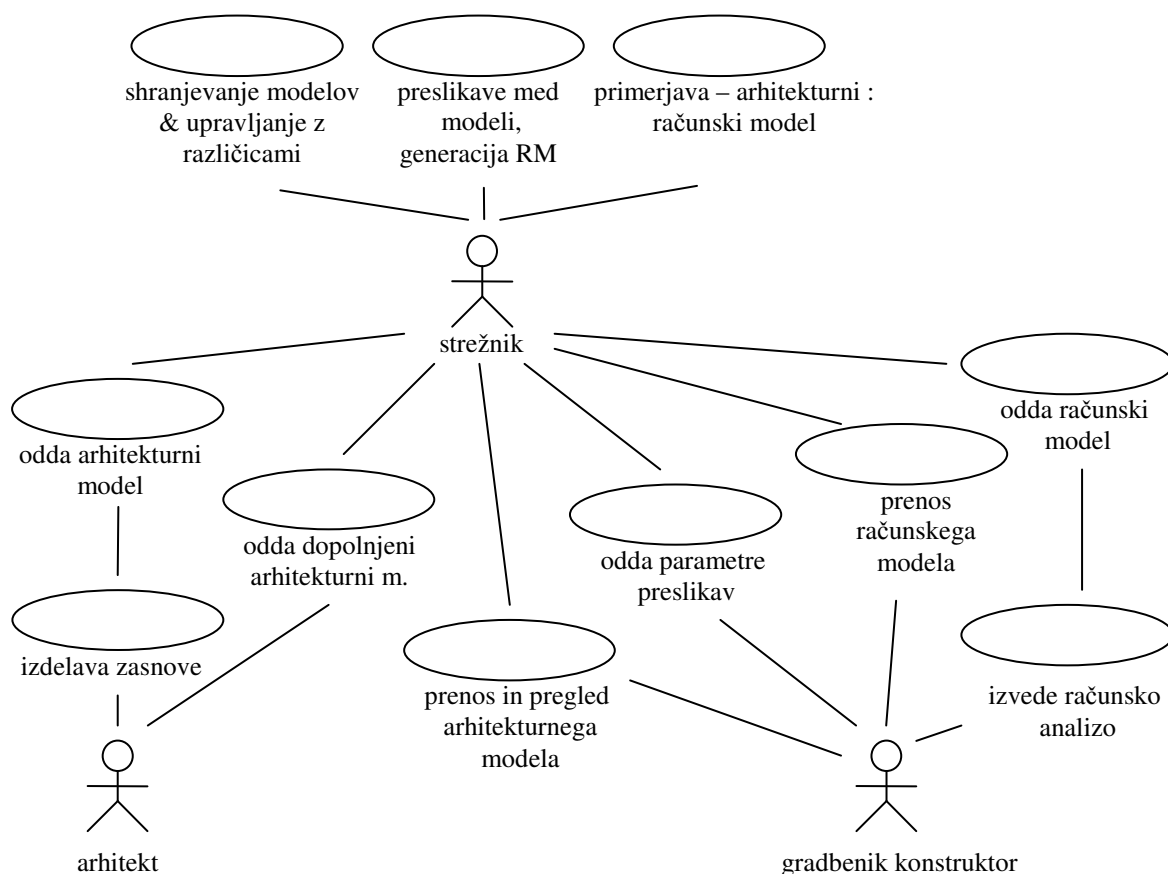
Figure 30: Imposed loads in XML form

### 3.5.3 Arhitektura predlaganega sistema

Predlagana rešitev skuša z uporabo spletnih tehnologij in baz podatkov pozitivno vplivati na proces načrtovanja gradbenih konstrukcij. Kot je prikazano na sliki 31 predlagani sistem sestavljajo trije glavni akterji: arhitekt, gradbenik konstruktor ter strežnik modela zgradb IFC. Večina trenutnih implementacij IFC specifikacije v praksi strežnika ne implementira. Zato – praviloma nezadovoljivo – njegovo vlogo prevzemata ostala dva akterja.

Arhitekturni model predstavlja osnovo načrtovanega sistema. Arhitekt lahko pri načrtovanju uporabi poljubno z IFC specifikacijo kompatibilno aplikacijo. Glede na definicijo pogledov na model ter na obravnavani delotok bi morali IFC vmesniki aplikacij za arhitekturno načrtovanje omogočati zapis modela v t. i. A-RM pogledu. Implementirane rešitve se žal razlikujejo: aplikacije za arhitekturno modeliranje večinoma implementirajo le razširjeni koordinacijski pogled. Ker slednji pokriva skoraj ves nabor entitet A-RM pogleda in ker

trenutno ne obstaja primernejša opcija, bo kot zasilna rešitev uporabljen implementiran razširjen koordinacijski pogled.



Slika 31: Diagram (upo)rabe

Figure 31: Use-case diagram

Arhitekt lahko po uspešni identifikaciji naloži IFC model na strežnik. Za zagotavljanje ažurnosti pri načrtovanju je predvideno obveščanje akterjev s pomočjo elektronskih sporočil.

Po prejetem sporočilu se gradbenik konstruktor preko spletnega brskalnika prijavi na strežnik modela zgradb IFC. Po predogledu arhitekture modela gradbenik konstruktor poda dodatne parametre za pretvorbo modela (npr. dimenzije modela za analizo (2D, 3D), izbira nosilnih elementov). Glede na podane parametre je možna implementacija različnih hevrističnih algoritmov. Za obravnavani primer geometrijskih preslikav med 3D podmodeli IFC specifikacije sta najprimernejši algoritem predlagala Nguyen in Oloufa (2002).

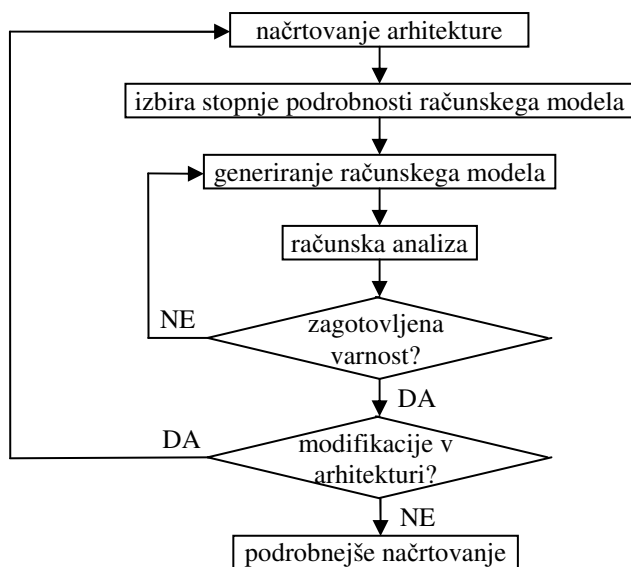
Vnos dodatnih parametrov s strani gradbenika konstruktorja (3D ali 2D model, izbira nosilnih elementov) sproži preslikave med modeloma. Zaradi spremljanja razvoja modela oz. njegove evolucije se tvori dopolnjena različica obravnavanega modela zgradb. Do zaključka računske analize konstrukcije z informacijskim modelom zgradb lahko upravlja le gradbenik konstruktor, pri čemer je upravljanje domensko specifično. Spreminjati oz. urejati je mogoče le tiste dele modela, ki se navezujejo na domeno konstruktive.

Ob upoštevanju elementov računskega modela za ploskovne (IfcStructuralSurfaceMember) in linijske elemente (IfcStructuralCurveMemeber) dobimo računski model, sestavljen iz ploskev in linij, ki so med seboj povezane v točkah oz. linijah. Možen je tudi opis stika dveh površin, vendar pa le-ta v nalogi ni bil obravnavan. Pri načrtovani preslikavi je predvideno, da so vsi stiki med elementi togi. Natančnejša opredelitev stikov neposredno iz arhitekturnega modela pri obravnavanem poteku načrtovanja namreč ni mogoča. Mogoča bi bila le v primeru obstoja vmesnega računskega modela za načrtovanje nosilne konstrukcije, kjer bi bil stik natančneje opredeljen.

Ob opisanem prirejanju se v informacijski model zgradb vnese dodatne entitete, ki določajo nosilnost posameznih gradnikov. Stebrom, prečkam, stenam in ploščam se pripiše dodaten nabor entitet (P\_setColumnCommon, P\_setBeamCommon, P\_setWallCommon, P\_setSlabCommon,), ki med drugim karakterizira (ne)nosilnost gradnika.

Bistvenega pomena za načrtovano zaokroženo načrtovanje je tudi ohranitev relacij med arhitekturnim in računskim modelom. Načeloma je za ohranitev relacij med gradniki potrebno implementirati entiteto, ki opisuje razmerja – IfcRelAssignToProduct. Zaradi specifičnosti IFC vmesnikov za analizo konstrukcij ni nujno, da bo navedeno razmerje v vseh primerih zadostovalo za ohranitev relacij. Zato se dodatno predlaga definiranje razmerij, ki temeljijo na identifikatorju GUID, ki predstavlja edinstveni kvalifikator posameznega primitiva. Ohranitev razmerja med gradniki je bistvena tudi z vidika določitve vplivov na konstrukcijo. Standard ENV 1991-1-1 podaja vplive na konstrukcijo glede na primitive računskega modela (lokacija zgradbe, balkon, pisarna, ipd.). Enostaven nazoren primer: Evrokod predpisuje za pisarne obtežbo  $3 \text{ kN/m}^2$ . Obtežbo lahko z ustreznim razmerjem povežemo le z elementi zgradb, posredno preko razmerij pa tudi s pripadajočim gradnikom računskega modela.

Po uspešno zaključenih preslikavah lahko gradbenik konstruktor prenese računski model (sicer vsebovan v skupnem modelu zgradb) na lokalni računalnik, ga uvozi v aplikacijo za računsko analizo, ustrezno dopolni/korigira ter opravi zahtevano analizo. Sicer je možen tudi prenos celotnega modela zgradb, vendar pa slednje ni priporočljivo oz. smiselno. Vmesniki aplikacij za analizo konstrukcij namreč implementirajo le ožji nabor entitet informacijskega modela zgradb, vse ostale entitete se pri uvozu modela ne ohranijo.



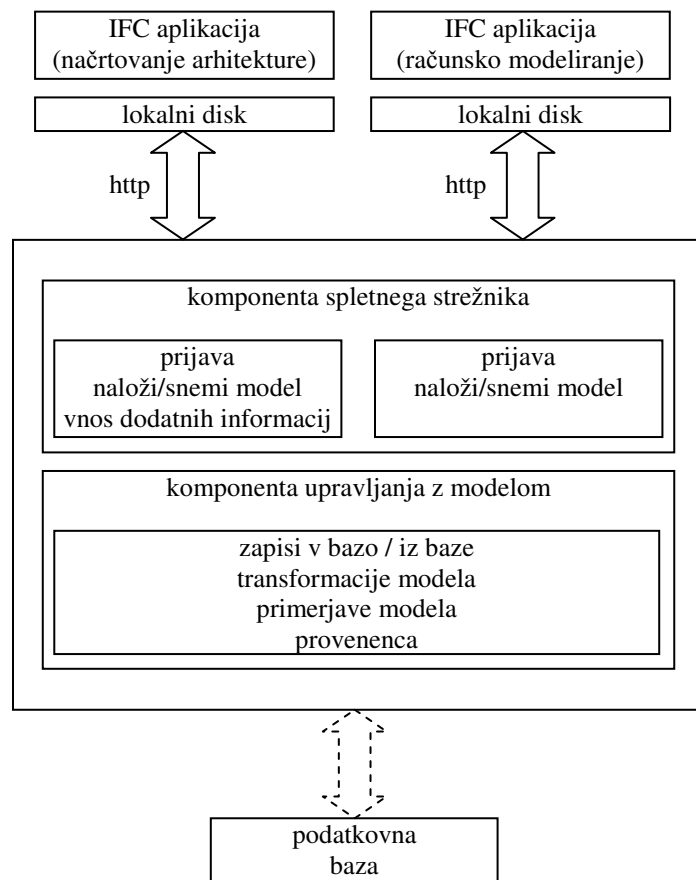
Slika 32: Diagram poteka predlaganega zaokroženega načrtovanja

Figure 32: Flowchart of suggested roundtrip engineering

Po opravljeni analizi gradbenik konstruktor rezultate računske analize shrani v zapisu IFC ter le-tega prenese na strežnik. Zaporedje entitet v vhodnem in izhodnem modelu za analizo je v splošnem lahko različno, preskusi vmesnikov pa so pokazali, da se pri manipulaciji z računskim modelom ne ohranjajo definirana razmerja med elementi zgradb in elementi računskega modela. Navedeno pomanjkljivost je teoretično možno zaobiti z upoštevanjem identifikatorjev GUID. Med analizo, ki jo opravlja gradbenik konstruktor se nosilnim elementom lahko spremeni lega v prostoru, dimenzije ali material. Zato se pri posredovanju rezultatov analize na strežnik navedene količine za vsako entiteto posebej primerjajo s pripadajočimi vrednostmi, ki jih vsebuje model na strežniku. Ob ugotovljenem neujemanju

(npr. spremeni se prečni prerez stebra) se ne izvršijo samodejne korekcije modela na strežniku, temveč se z elektronskim sporočilom obvesti arhitekta o potrebnih spremembah. Slednji nato popravi izvorni model in ob morebitnih dodatnih spremembah konstrukcije model ponovno posreduje v obravnavo gradbeniku konstruktorju. Navedeno iterativno delo (slika 32) očitno rezultira v zaokroženo načrtovanje, poznano iz vsakodnevne prakse.

Slika 33 prikazuje predlagano arhitekturo IFC strežnika modela zgradb.



Slika 33: Arhitektura IFC strežnika modela zgradb

Figure 33: IFC model server architecture

## **4 PROTOTIPNA IMPLEMENTACIJA PREDLAGANE REŠITVE**

Kompleksnost v prejšnjem poglavju predstavljene splošne konceptualne rešitve onemogoča njeno implementacijo v celoti. Predstavljeni koncepti bodo zato prikazani na omejenem prototipnem primeru, ki sicer verodostojno potrjuje tezo naloge. Opisu predpostavk in omejitev prototipa sledi kratek opis uporabljenih orodij. Osrednji del poglavja predstavlja opis implementacije predlagane rešitve ter prikaz delovanja predlaganega sistema na enostavnem testnem primeru. V sklepnem delu bodo predstavljene ugotovitve testiranja ter možne izboljšave prototipne implementacije.

### **4.1 Predpostavke in omejitve**

Nepopolna implementacija predlagane rešitve zahteva natančno opredelitev predpostavk in omejitev. Le-te morajo biti dovolj splošne, da bo s prototipom vseeno možno potrditi v prejšnjem poglavju predstavljene koncepte. Predpostavke in omejitve je smiselno razdeliti v dve skupini: v prvo sodijo tiste, ki so pogojene z omejitvijo dosega in preslikav, medtem ko v drugo skupino uvrščamo tiste, ki jih pogojuje izbrana programska oprema.

#### **4.1.1 Predpostavke in omejitve dosega in preslikav**

Prve omejitve dosega predstavljajo v prejšnjem poglavju predstavljeni »pogledi na model«. Dodatno se prototipna implementacija omejuje na naslednje nosilne elemente zgradb: stebri, prečke, stene in plošče.

Tudi spekter vplivov na konstrukcijo, ki jih predpisuje standard Evrokod, je bilo za potrebe prototipne implementacije potrebno zožiti. Trenutno je implementirano prirejanje koristne obtežbe v zgradbah, kot jo predpisuje standard ENV 1991-1-1:2001, poglavje 6.3.1.

Kategorije uporabe in vrednosti obtežbe so podane v preglednicah 12 in 13.

V skladu s standardom Evrokod se v primeru večnamenske rabe prostorov upošteva najbolj neugoden primer enakomerno porazdeljene obtežbe ( $q_k$ ) oz. točkovne obtežbe ( $Q_k$ ). Če je potrebno (balkoni, stopnišča), se namesto priporočenih podčrtanih vrednosti privzame večje

vrednosti. Obtežbo se lahko reducira s faktorjem  $\alpha_A$ , ki se določi po enačbi ( $A_0$  predstavlja referenčno površino  $10 \text{ m}^2$ ,  $A$  pa dejansko površino,  $\psi_0$  pa kombinacijski faktor):

$$\alpha_A = \frac{5}{7}\psi_0 + \frac{A_0}{A} \leq 1 \quad (1)$$

Faktor  $\alpha_A$  mora biti pri kategorijah C in D večji od 0,6.

Prirejanje stalne obtežbe sicer ni implementirano, saj se le-ta v večini programov za analizo konstrukcij avtomatsko določi iz karakteristik prečnih prerezov ter iz karakteristik materialov. Proces prirejanja dodatne stalne obtežbe (npr. estrihi, talnem obloge) bi bilo z uporabo entitete IfcCovering ter malenkostnimi spremembami programske kode tudi mogoče avtomatizirati.

Prototip se omejuje na preslikave, kjer je dimenzija prostora izvornih elementov enaka dimenziji prostora ciljnih elementov.

#### Preglednica 12: Kategorije uporabe

Table 12: Categories of use

<b>kategorija</b>	<b>specifična raba</b>	<b>primer</b>
A	stanovanja	sobe v stanovanjih, hišah, bolnišnicah, hotelih
B	pisarniški prostori	
C	kongregacija množice ljudi	C1: prostori z mizami (šole, kavarne, restavracije, recepcije)
		C2: prostori s fiksnimi sedeži (cerkve, kinodvorane, predavalnice, konferenčne sobe, čakalnice)
		C3: prostori brez ovir pri gibanju (muzeji, razstavniki prostori, hodniki v upravnih zgradbah, muzejih, hotelih, postajah)
		C4: območja, kjer so možne fizične aktivnosti (plesne dvorane, odri, gimnastične sobe)
		C5: območja, kjer se lahko pojavijo velike množice: zgradbe, kjer se odvijajo množične prireditve (koncertne dvorane, športne hale – stojišča, terase, območja dostopa, ipd.)
D	trgovine	D1: območja v manjših trgovinah
		D2: območja v nakupovalnih centrih

Preglednica 13: Koristna obtežba prostorov, balkonov in stopnišč v zgradbah

Table 13: Imposed loads on floors, balconies and stairs in buildings

kategorije	$q_k$ [kN/m <sup>2</sup> ]	$Q_k$ [kN]
A – etaže	1,5 – <u>2,0</u>	<u>2,0</u> – 3,0
A – stopnišča	<u>2,0</u> – 4,0	<u>2,0</u> – 4,0
A – balkoni	<u>2,5</u> – 4,0	<u>2,0</u> – 3,0
B	2,0 – <u>3,0</u>	1,5 – <u>4,5</u>
C – C1	2,0 – <u>3,0</u>	3,0 – <u>4,0</u>
C – C2	3,0 – <u>4,0</u>	2,5 – 7,0 ( <u>4,0</u> )
C – C3	3,0 – <u>5,0</u>	<u>4,0</u> – 7,0
C – C4	4,5 – <u>5,0</u>	3,5 – <u>7,0</u>
C – C5	<u>5,0</u> – 7,5	3,5 – <u>4,5</u>
D – C1	<u>4,0</u> – 5,0	3,5 – 7,0 ( <u>4,0</u> )
D – C2	4,0 – <u>5,0</u>	3,5 – <u>7,0</u>

#### 4.1.2 Predpostavke in omejitve programske opreme

Eno izmed poglavitnih prednosti IFC specifikacije predstavlja njena prosta dostopnost. Žal predstavljenemu pristopu ne sledijo IFC vmesniki. Slednji so vključeni v programsko opremo ali pa jih je potrebno dokupiti. V predhodnih raziskavah (Pazlar, Turk, 2006) ter ob analizi delovanja najnovejših IFC vmesnikov je bilo ugotovljeno, da večina vmesnikov najbolj pogosto uporabljene programske opreme za načrtovanje še vedno ni zadosti robustna. Ne glede na predpisan pogled na model se obseg implementacije lahko razlikuje, podobno velja tudi za interpretacijo specifikacije. Zato je bilo pred vključitvijo v prototipno rešitev potrebno preveriti sposobnost oz. pravilnost delovanja vmesnika pri izmenjavi entitet, ki so predmet zanimanja.

##### 4.1.2.1 Programska oprema za arhitekturno načrtovanje konstrukcij

Velika večina najbolj uveljavljene programske opreme za arhitekturno načrtovanje konstrukcij omogoča shranjevanje modelov v zapisu IFC. Takšno stanje je pričakovano, saj je domena arhitekture v IFC specifikacijo vključena že od prve različice specifikacije dalje. Poleg ustreznega nabora entitet ter pravilnosti delovanja dodatni kriterij pri izbiri vmesnika



predstavlja zmožnost opisa razmerij med elementi zgradbe in namembnostjo prostorov. Navedeno preskušanje sicer zavrača osnovno zamisel interoperabilnosti, vendar pa se je glede na ugotovljeno stanje implementacij IFC specifikacije izkazala za nujno potrebno.

Preskušena je bila naslednja programska oprema: Autodesk Architectural Desktop 2006 z INOPSO IFC vmesnikom (IFC 2x2), Graphisoft Archicad 9 (IFC2x2), Graphisoft Archicad 10 (IFC2x2, IFC 2x3), Nemetschek Allplan 2006 (IFC2x2, IFC 2x3). Najbolj zanesljivo, vendar še vedno ne povsem zadovoljivo delovanje vmesnikov, je bilo ugotovljeno pri programu Archicad 10 (različica IFC 2x3).

#### **4.1.2.2 Programska oprema za računsko modeliranje in analizo konstrukcij**

Večina programske opreme za analizo konstrukcij vsebuje zadosti zmogljive geometrijske modelirničke za računsko modeliranje konstrukcij. Na trgu se sicer pojavlja posebna programska oprema, namenjena izključno detajlnemu modeliranju, s katero bi bilo mogoče detajlno opisati nosilno konstrukcijo in posledično izdelati detajlni računski model (npr. Autodesk Revit), vendar pa se takšen pristop vsaj v začetnih fazah iterativnega načrtovanja, kjer se določajo globalne dimenzije zgradbe ter prerezov, običajno ne uporablja.

Število zadovoljivo delujočih IFC vmesnikov programske opreme za analizo konstrukcij je relativno skromno (uveljavljenih programov za analizo konstrukcij je namreč precej več kot pa programov za načrtovanje arhitekture). Glede na predlagano arhitekturo sistema mora izbrana aplikacija za računsko načrtovanje oz. analizo operirati z v prejšnjem poglavju predstavljenimi entitetami arhitekturnega modela in modela za analizo.

Preskušena je bila naslednja programska oprema: AxisVM 9 (IFC2x3), Tekla R13 (IFC 2x3), Dlubal RSTAB 6 (IFC2x3), SCIA ESA PT (IFC2x2), CSI ETABS 9 (IFC 2x2). Izmed navedene programske opreme so le prve tri navedene aplikacije sposobne operirati tako z elementi zgradb kot tudi z gradniki modela za analizo. Po natančnem preskusu vmesnikov je bil izbran program AxisVM. Le-ta ima najbolj dodelan IFC vmesnik, hkrati pa ima vse karakteristike sodobne programske opreme za načrtovanje oz. računsko analizo konstrukcij.

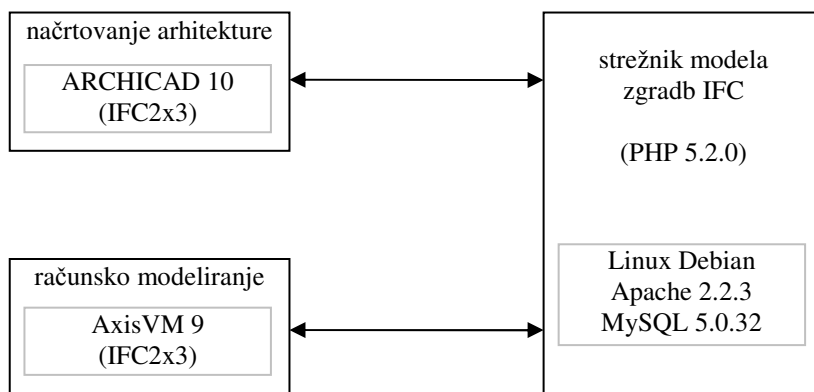
#### **4.1.2.3 Strežnik modela zgradb IFC**

Pri izbiri strežnika modela zgradb so prisotne še večje omejitve kot pri že opisani IFC skladni programske opremi, še posebej ob omejitvi na prostodostopne strežnike. Trenutno obstajajo le

trije IFC skladni strežniki, in sicer Eurostep strežnik (EMS – Eurostep ModelServer for IFC), EPM strežnik (EDM server) ter IMSvr strežnik. Izmed navedenih je le IMSvr prostodostopen (Adachi, 2002). Navedenemu strežniku sicer ni mogoče oporekati tehničnih pomanjkljivosti (omogoča upravljanje s celotnim/delnim modelom, shranjevanje IFC modela v podatkovni bazi, uporaba standardiziranih spletnih tehnologij (XML, spletne storitve/SOAP), prosta dostopnost), vendar je izredno pomanjkljivo dokumentiran, dodatno pa je delovanje nekaterih komponent sistema pogojeno s preživeto programsko opremo. Razvoj IMSvr strežnika uradno sicer ni bil nikoli zaključen, vendar pa datum zadnjih korekcij (06/2002) nazorno dokazuje njegovo usodo.

Zato predstavlja razvoj lastnega strežnika modela zgradb edino utemeljeno rešitev. Pri razvoju je bil uporabljen operacijski sistem Linux Debian, na katerem teče strežnik Apache 2.2.3. Na strežniku teče tudi upravljalnik relacijske zbirke podatkov MySQL (različica 5.0.32), ki omogoča kompleksne poizvedbe po – v podatkovno bazo preslikanem – informacijskem modelu zgradb. Uporabljen je bil skriptni programski jezik PHP (različica 5.2.0) .

## 4.2 Implementacija rešitve



Slika 34: Implementiran prototip

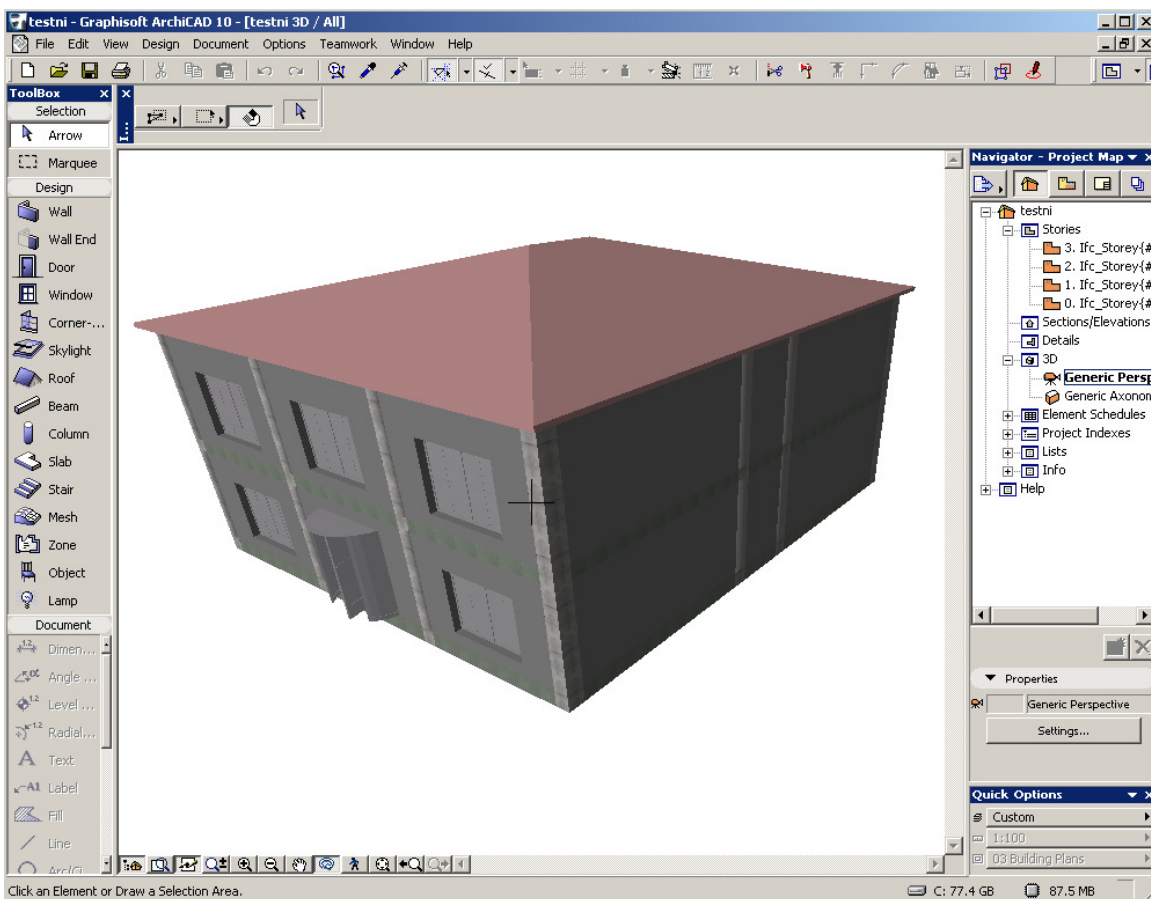
Figure 34: Implemented prototype

Implementacija rešitve (slika 34) temelji na predpostavljenih omejitvah dosega preslikav ter na omejitvah programske opreme. V nadaljevanju bodo predstavljeni posamezni koraki zaokroženega načrtovanja zgradb, temelječega na predlaganemu prototipu.

#### 4.2.1 Modeliranje arhitekture

Načeloma je pri načrtovanju arhitekture zgradbe (vključno z ureditvijo okolice) možno uporabiti povsem poljubne gradnike s poljubno natančno stopnjo opisa. V prejšnjem poglavju navedene omejitve je potrebno upoštevati le za nosilne gradnike konstrukcije.

Slika 35 prikazuje obravnavani testni primer: dvoetažna nepodkletena okvirna konstrukcija s ploščo le v prvi etaži. Streha je štirikapnica z leseno nosilno konstrukcijo, ki se analizira ločeno.

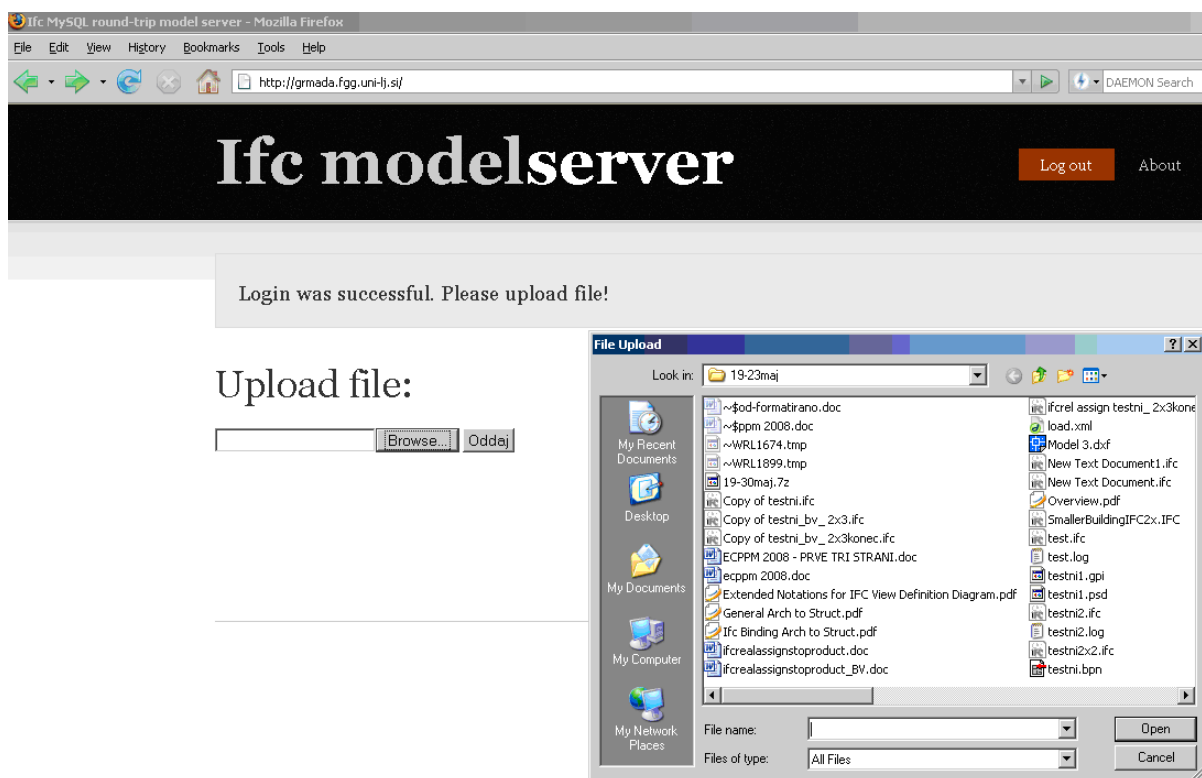


Slika 35: Arhitekturno načrtovanje (Graphisoft Archicad 10)

Figure 35: Architectural design (Graphisoft Archicad 10)

Za dokaz predstavljenih konceptov je pri arhitekturnem modeliranju konstrukcije nujno potrebno določiti namembnost prostorov. Pri analizi delovanja IFC vmesnika programa Archicad se je izkazalo, da opis namembnosti prostorov ne temelji na semantičnih primitivih zgradbe (stenah, ploščah) oz. na njihovi geometrijski predstavitvi (točke, linije oz. ploskve), temveč se izključno za obravnavani namen definirajo novi geometrijski primitivi. Posledično ne obstaja neposredna povezava med elementi zgradb in namembnostjo prostora, zato je potrebno v IFC model vključiti dodatna razmerja (entiteta `IfcRelAssignToProduct`).

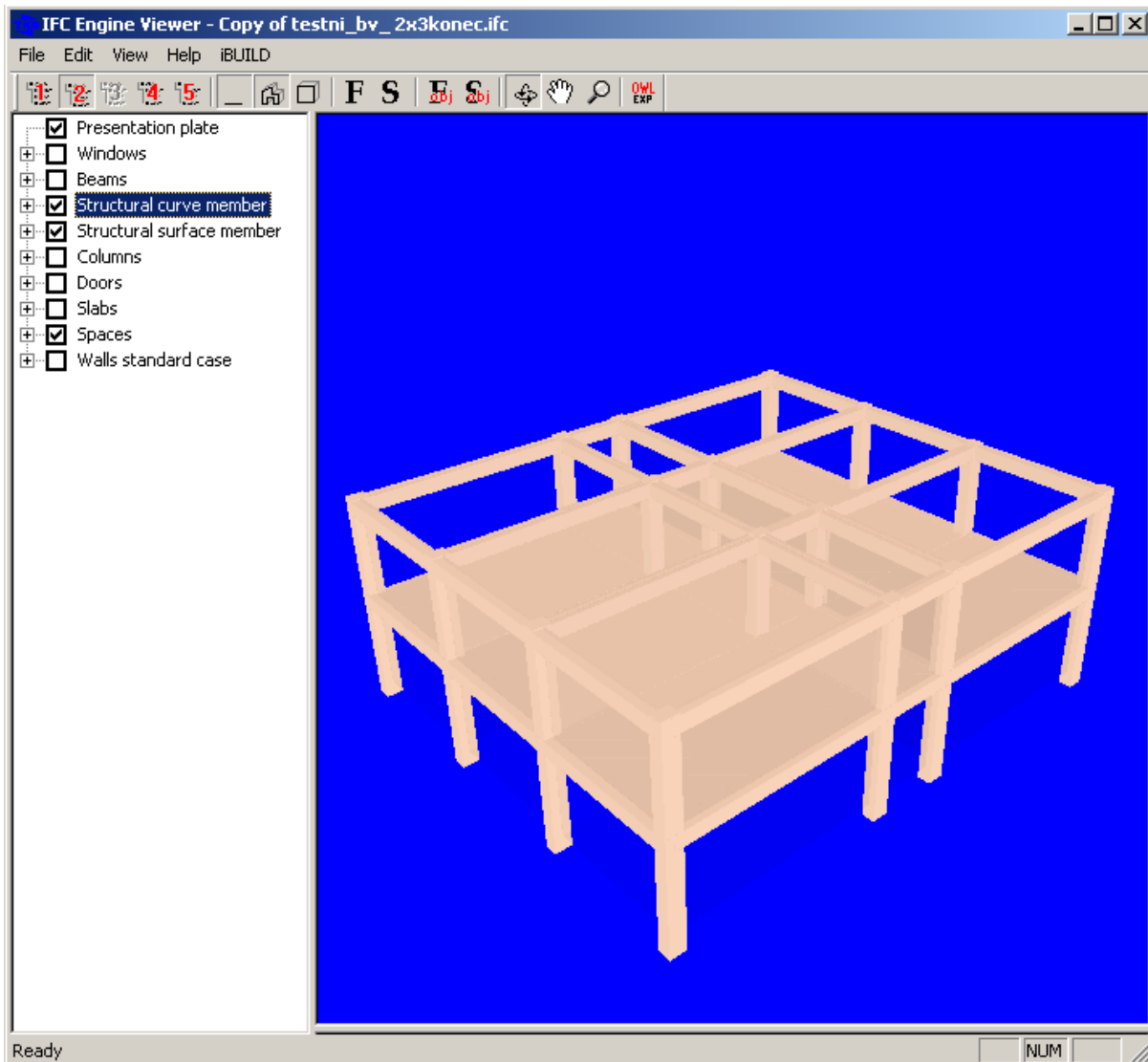
Pri shranjevanju IFC modela se sicer opcijsko lahko poda še podatke o avtorju, izbere ustrezne enote in natančneje opredeli geometrijsko predstavitev elementov zgradb.



Slika 36: Strežnik modela zgradb IFC – nalaganje arhitekturnega modela

Figure 36: IFC model server – uploading architectural design

Uspešno naložen arhitekturni model predstavlja zaključek prve faze zaokroženega načrtovanja (slika 36). Podobno kot arhitektu je tudi gradbeniku konstruktorju omogočeno delo šele po uspešni domensko vezani prijavi. Gradbenik konstruktor najprej pregleda arhitekturni model ter na podlagi pridobljenih informacij sprejeme odločitev o računskem modelu.



Slika 37: Arhitekturni model z elementi zgradb (IfcBeam, IfcColumn, IfcSlab) in računski model (IfcStructuralCurveMemeber, IfcStructuralSurfaceMember)

Figure 37: Architectural model with building elements (IfcBeam, IfcColumn, IfcSlab) and structural model (IfcStructuralCurveMemeber, IfcStructuralSurfaceMember)

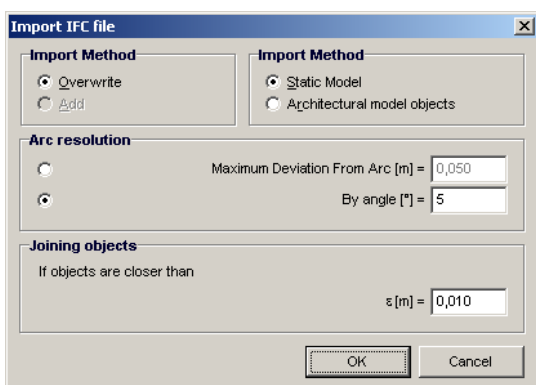
S potrditvijo povratnih informacij se sproži polavtomatska preslikava arhitekturnega v računski model. Pri pretvorbi je bistvenega pomena ohranitev relacij med elementi zgradb in novoustvarjenimi elementi računskega modela. Pri tem se elementom zgradb, ki so nosilni, pripiše dodatna entiteta (Pset), ki tudi v arhitekturnem modelu zgradb karakterizira njihovo nosilno naravo. V drugi fazi prirejanja se na podlagi semantike arhitekturnega modela in informacij o vplivih na konstrukcijo (XML zapis) gradnikom računskega modela pripiše še obtežba (v predstavljenem primeru je to plošča).

Po opravljenih preslikavah se iz aktualne baze podatkov generira nova IFC datoteka, v kateri sta zapisana tako arhitekturni kot tudi model za analizo (slika 37).

#### 4.2.2 Računsko modeliranje/analiza

Po opravljeni polavtomatski pretvorbi arhitekturnega modela v računski model lahko gradbenik konstruktor dostopa do modela zgradb, v katerem sta shranjena dva podmodela: arhitekturni in model za analizo. Po pretvorbi je celoten IFC model potrebno prenesti s strežnika in nadalje operirati z modelom, ki se nahaja na lokalnem računalniku.

Na sliki 38 je prikazan IFC vmesnik programa AxisVM, ki prikazuje tudi njegovo bistveno prednost. Vmesnik namreč omogoča uvoz tako arhitekturnega modela kot tudi modela za analizo.

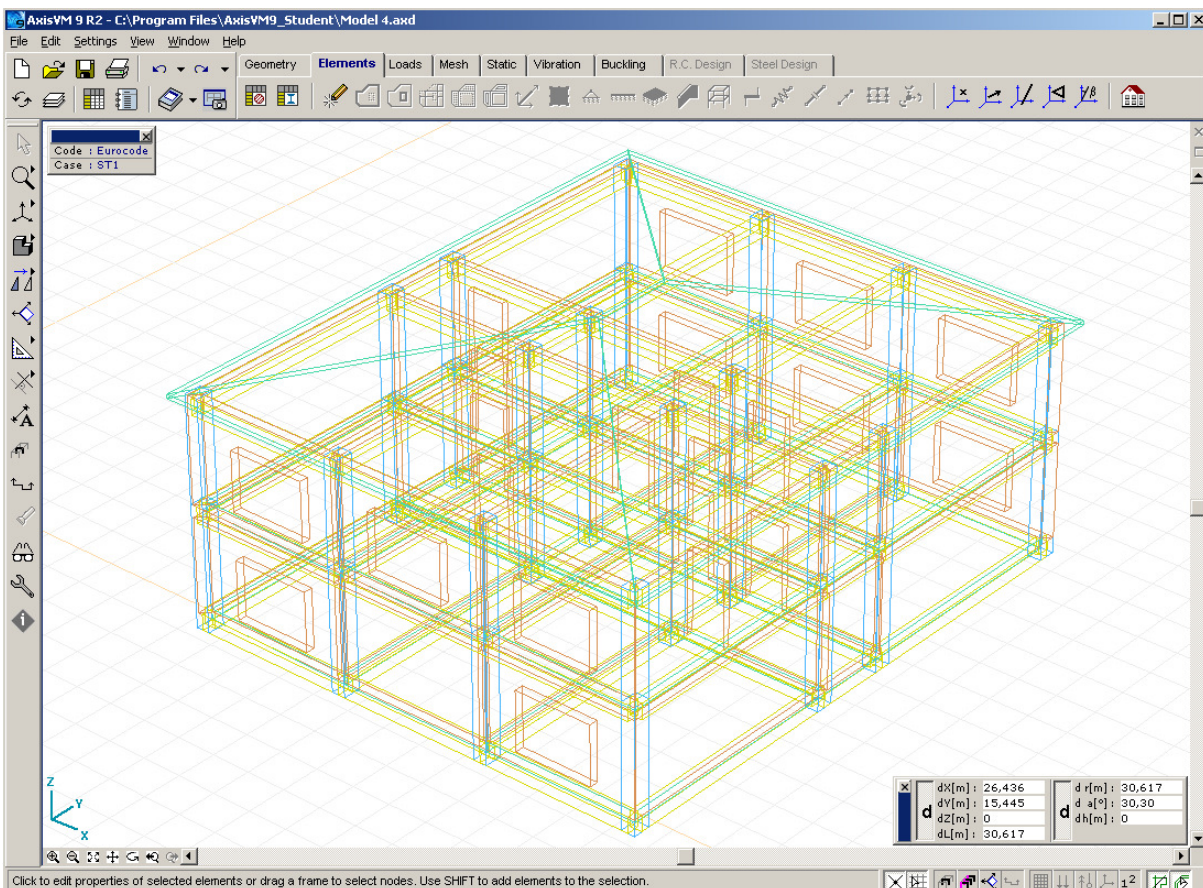


Slika 38: IFC vmesnik programa AxisVM za uvoz informacijskega modela zgradb

Figure 38: AxisVM IFC interface for BIM import

Pri uvozu arhitekturnega modela (slika 39) vmesnik iz IFC datoteke prebere vse entitete, ki so kakorkoli povezane z nosilno konstrukcijo (stebri, prečke, plošča, streha, ipd.), eliminira pa vse gradnike, ki nimajo karakteristik nosilnih elementov (npr. okna, vrata, pohištvo, ipd.). Uvoženih gradnikov je tako lahko mnogo več, kot je pa nosilnih elementov, ki se upoštevajo v izračunu. Tako se na primer na predstavljenem primeru v računsko konstrukcijo preslikajo tudi vse stene, čeprav le-te v predstavljenem primeru nimajo nosilnih karakteristik.

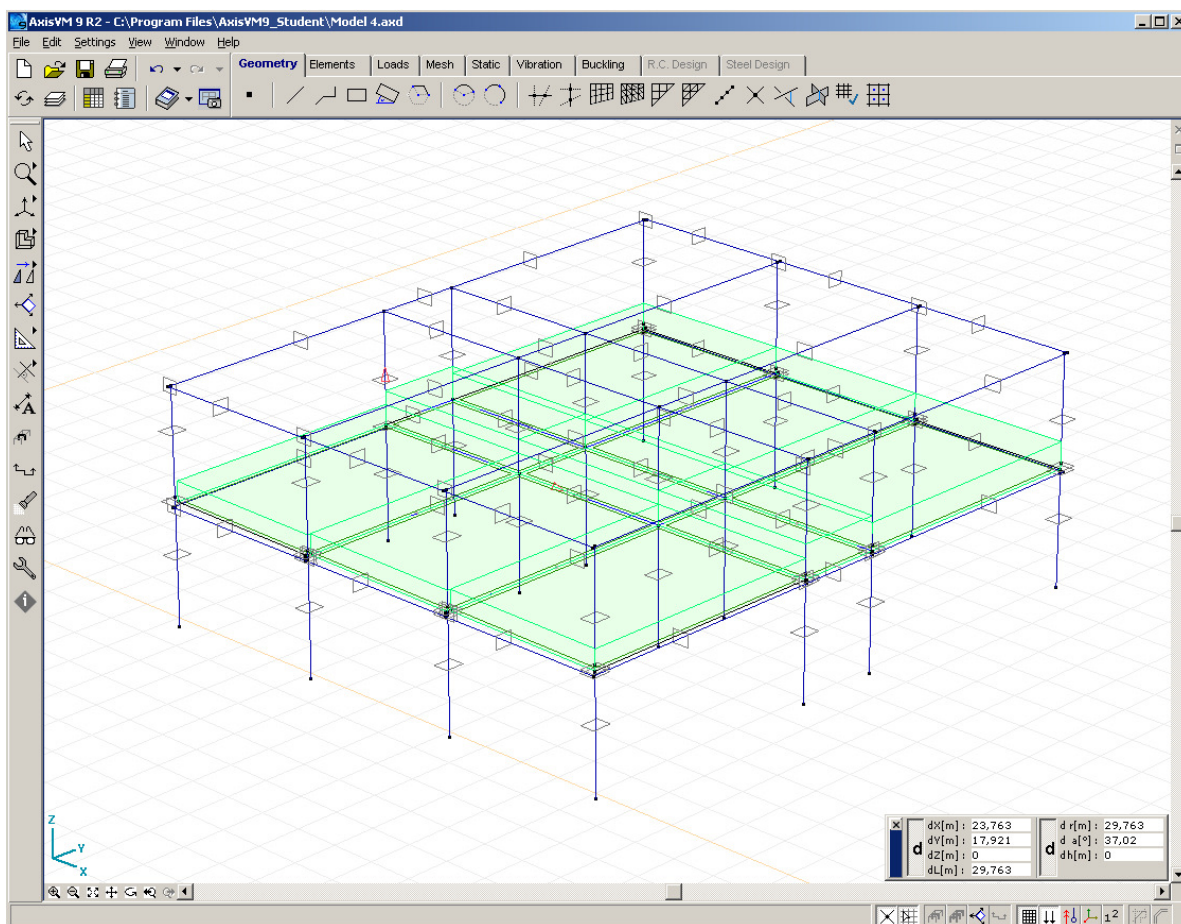
Uvoženi arhitekturni model lahko predstavlja le geometrijsko osnovo za generacijo novega neodvisnega modela, možno pa je uporabiti vgrajeni algoritem, ki uvoženi model razbije, njegove dele pa priredi v primitive računskega modela. Praktično je bil preskušen tudi navedeni pristop, njegovo poglavitno slabost pa predstavlja izguba relacij med elementi zgradb in elementi modela za analizo, s čimer je praktično onemogočeno zaokroženo načrtovanje.



Slika 39: Uvoz arhitekturnega modela v program AxisVM

Figure 39: Importing architectural model into AxisVM

Za primernejšega se izkaže uvoz računskega modela (slika 40). V tem primeru se – vsaj v prvi fazi zaokroženega načrtovanja – ohranjajo relacije med gradniki obeh modelov. Ker je mogoče naravo nosilne konstrukcije opredeliti že pred prirejanjem, je takšen računski model primernejši. Dodatno računski model vsebuje že nekatere obtežbe na konstrukcijo, ki bi jih bilo sicer potrebno podajati ročno.



Slika 40: Uvoz računskega modela v program AxisVM

Figure 40: Importing structural model into AxisVM

Gradbenik konstruktor po zaključenem uvozu preveri pravilnost pretvorb geometrije, prirejanja vplivov na konstrukcijo in ustrezno dopolni računski model. Po izvedeni računski analizi se model zopet shrani v zapisu IFC, in sicer kot računski model. Možno je sicer shranjevanje arhitekturnega modela, vendar se v navedenem primeru ne shranijo vse



karakteristike, potrebne za zaokroženo načrtovanje. Pri tem je potrebno izpostaviti, da se pri obeh variantah v IFC model shranijo le gradniki nosilne konstrukcije (stebri, prečke, plošče, ipd.), vsi ostali primitivi so bili izgubljeni že pri uvozu modela.

### **4.2.3 Zaokroženo načrtovanje**

Po zaključeni računski analizi mora gradbenik konstruktor arhitektu posredovati informacije o morebitnih spremembah načrtovane konstrukcije. Spremembe, ki se nanašajo na (1) lego primitivov, (2) na prečni prerez elementov ter na (3) eventualno spremembo materiala, bi bilo po predlaganem algoritmu mogoče aplicirati polavtomatsko.

Identifikacija oz. relacija med elementi zgradb in elementi računskega model poteka na osnovi identifikatorja GUID. Žal se je izkazalo, da IFC vmesnika obeh uporabljenih aplikacij (AxisVM, Archicad 10) nezadovoljivo upravljata z identifikatorji GUID, in sicer ne samo pri obravnavanih gradnikih (podentitetami IfcBuildingElement pri načrtovanju arhitekture ter IfcStructuralMember v modelu za analizo), temveč pri vseh primitivih modela zgradb. Pri preslikavi gradnikov v katerokoli izmed navedenih aplikacij se identifikatorji ne ohranjajo, temveč se pri ponovnem izvozu modela gradnikom enostavno predpišejo nove vrednosti GUID.

Vsi poskusi, da bi navedeno pomanjkljivost odpravili s predpisovanjem opsijskih atributov (npr. Name, Description), so se izkazali za neuspešne, saj se tudi njihove vrednosti v večini primerov spreminjajo.

Žal v navedenem primeru ni mogoče uporabiti niti identifikacije gradnikov glede na njihovo lego v prostoru, saj se le-ta v postopku načrtovanja lahko spreminja, dodatno pa se lahko dodajajo novi elementi.

Zaradi navedenih pomanjkljivosti IFC vmesnikov (na katerih razvoj in delovanje neposredno ni mogoče vplivati) predlaganega sistema ni bilo mogoče v celoti implementirati.

## **4.3 Testiranje implementirane rešitve**

Implementirana rešitev je bila preskušana že med samim razvojem posameznih komponent. Po združitvi komponent v celoto je bilo izvedeno testiranje celotnega sistema, vključno z delovanjem IFC vmesnikov za načrtovanje arhitekture ter računsko modeliranje.

Testiranje je bilo izvedeno na dveh nivojih. Z enostavnimi testnimi primeri se je ugotavljala predvsem pravilnost delovanja sistema, medtem ko se je pri kompleksnejših testnih primerih opazovala tudi časovna učinkovitost predlaganega sistema za realno delo v praksi.

Scenarij testiranja je bil pri vseh modelih enak in sovпада s prvo fazo opisanega zaokroženega inženirskega načrtovanja.

Testiranje prototipa se je nanašalo predvsem na opazovanje implementiranega prirejanja med arhitekturnim in računskim modelom. Globalni izsledki testiranja celotnega sistema (vključno z delovanjem IFC vmesnikov) bodo podani v naslednjem podpoglavju.

### **4.3.1 Testni primeri**

Vsi preskušeni enostavni modeli so potrdili pravilnost delovanja tako geometrijskega prirejanja kot tudi prirejanja obtežbe. Odstopanja so se pojavila le v primerih, kjer zgradba arhitekturnih modelov ni bila konsistentna (npr. pri neustreznem stikovanju elementov, to je pri prekrivanju gradnikov arhitekturnega modela).

Pri testiranju obsežnejših primerov se nismo ukvarjali z načrtovanjem modelov, temveč so bili le-ti pridobljeni iz raznih virov, predvsem preko spleta. Izbrani so bili predvsem produktni modeli s pravilno geometrijo, to je takšni, ki ustrezajo predpostavkam prototipne implementacije. Tudi pri kompleksnejših testnih primerih je implementirani prototip deloval zadovoljivo, še najbolj moteča elementa sta predstavljala nekonsistentnost geometrije in časovna komponenta pri manipulaciji z modelom IFC.

### **4.3.2 Ugotovitve pri testiranju**

Testiranja, ki so potekala po opisanih scenarijih, so pokazala, da predlagana rešitev v veliki meri zadovoljuje pričakovanja. Pravilnost delovanja prototipa je pogojena z natančnostjo oz. korektnostjo arhitekturnega modeliranja. V primeru neupoštevanja napotkov, podanih v poglavju 3.5, rezultati preslikav niso zadovoljivi in so potrebne ročne korekcije generiranega modela.

Pojavile so se le malenkostne zahteve pri definiranju preslikav med modeloma oz. pri prirejanju obtežbe, ki so bile naknadno upoštevane in vključene v sistem. Večje težave so bile

identificirane le pri učinkovitosti delovanja sistema pri kompleksnih testnih primerih. Pri tem je potrebno izpostaviti dejstvo, da je bil prototip preskušan na neobremenjenem strežniku.

Manipulacija s celotnim IFC modelom rešuje težave z zagotavljanjem konsistentnosti modela. Pri enostavnih testnih primerih prenos celotnih modelov ne predstavlja ovir, medtem ko pa se je pri kompleksnih testnih primerih in pri nižji pasovni širini povezave prenos časovno izkazal kot moteč element.

#### **4.4 Predlagane izboljšave prototipa**

Kot dokaz koncepta je bil implementiran le del predlaganega algoritma. Ne glede na relativno popolno definicijo »uporabnega minimuma« izmenjave med modeli pa bi bilo potrebno več pozornosti nameniti algoritmom prirejanja, funkcionalnost prototipa pa bi bilo mogoče povečati tudi z vključitvijo dodatnih obtežb.

Večjo pozornost bi veljalo nameniti izbiri programske opreme. Temeljito bi bilo potrebno preveriti karakteristike in delovanje zadnjih različic IFC vmesnikov aplikacij za načrtovanje. Možno bi bilo izbrati enega izmed komercialnih sistemov (Eurostep IFC Model Server ali EPM EDMServer), ki bi sicer zagotavljal večjo robustnost sistema, vendar pa bi ob tem eliminirali temeljno predpostavko načrtovanega sistema (prosto dostopnost).

Izboljšati bi bilo potrebno tudi grafično podobo, ki je trenutno omejena le na elemente, ki so nujno potrebni za dostop do vseh implementiranih funkcionalnosti.

Časovno potratno izmenjavo bi bilo mogoče vsaj delno eliminirati z delno izmenjavo domensko specifičnih podmodelov, za katere bi na strežniku obstajal zbirni model. Za učinkovito delovanje predlaganega pristopa bi bilo potrebno implementirati na SOAP protokolu temelječ spletni servis. Istočasno bi bilo potrebno določiti vloge in upravljanje z zbirnim modelom oz. podmodeli.

## 5 SKLEP

### 5.1 Glavne ugotovitve

V predstavljeni nalogi so bile raziskane možnosti za učinkovito sodelovanje arhitektov in gradbenikov konstruktorjev na podlagi uporabe informacijskega modela zgradb IFC. V raziskavi je bila najprej opredeljena terminologija informacijskega modeliranja zgradb v AEC-FM industriji in izdelan pregled razvoja informacijskih modelov. Podrobneje so bila opisana področja, ki se navezujejo na predstavljeno sodelovanje oz. na prirejanja med arhitekturnimi modeli in modeli za računsko analizo.

Predstavljena raziskava je zasnovana na predpostavki, da sodobni procesi zasnove, načrtovanja, gradnje, upravljanja in odstranitve zgradb temeljijo na informacijskem modelu zgradb, ki predstavlja zbirnik vseh informacij, ki so kadarkoli na voljo vsem udeležencem navedenih procesov. V okviru naloge je bila izdelana arhitektura sistema za skupno delo arhitektov in gradbenikov konstruktorjev, ki za razliko od predhodno definiranih arhitektur temelji na informacijskem modelu zgradb IFC in na polavtomatskih preslikavah med arhitekturnimi modeli in modeli za računsko analizo.

Arhitektura predlaganega sistema temelji na semantičnem kontekstu informacijskih modelov zgradb, ki predstavlja teoretično osnovo za več kot le transformacijo oz. izvlečke geometrije. Znotraj informacijskega modela zgradb sta bila opredeljena dva nova pogleda, ki natančneje določata nabor informacij, ki so predmet izmenjave. Za zagotavljanje konsistentnosti je bil v sistem vključen spletni strežnik, ki zagotavlja ustrezno upravljanje z modelom. Osnovna funkcionalnost sistema je na voljo preko spleta in je zato udeležencem načrtovanja praktično dostopna kadarkoli in kjerkoli. Avtorizacija se opravlja na podlagi uporabniškega imena in gesla.

Obravnavani koncepti preslikav in domenskega sodelovanja se v predstavljenem sistemu sicer omejujejo na domeno arhitekture in konstruktive, vendar pa je lahko njihov doseg (oz. doseg predlaganega strežnika modela zgradb) ob vključitvi ostalih domen in ustreznih prilagoditvah precej večji.

Večino konceptov, ki jih predstavlja zasnovani sistem za skupno načrtovanje, je uporabljenih tudi v implementiranem prototipu, s katerim smo želeli preveriti predlagano hipotezo. Prototipna implementacija sistema temelji na odprtokodnih aplikacijah oz. orodjih (Linux, Apache, MySQL, PHP) ter na odprtih spletnih komunikacijskih protokolih (HTTP). Slednje zagotavlja platformo neodvisnost ter tudi s tem ob že dokazanih prednostih sistema favorizira njegovo uporabo v praksi.

Testiranje prototipne rešitve je načeloma pokazalo, da predlagana rešitev oz. njena implementacija rešuje ključne probleme, identificirane v začetni fazi raziskave. Delno k temu prispevajo tudi omejitve prototipne rešitve.

Žal globalna ocena karakteristik predlaganega sistema ni tako navdušujoča, predvsem zaradi omejenosti IFC vmesnikov oz. zaradi nekorektne implementacije IFC specifikacije.

Takšni rezultati testiranja so bili glede na predstavljena poročila o delovanju IFC vmesnikov tudi pričakovani. Specifikacija in posledično implementacije bodo v sedanji formi le s težavo zadostile vedno strožjim zahtevam AEC-FM industrije po natančnosti, konsistenci, integraciji, koordinaciji in sinhronizaciji. Dodatni oviri pri razvoju ter pri implementaciji predstavljata togost in neprofesionalnost organizacije IAI, pri čemer neprofesionalnost nikakor ne predstavlja sinonima za nestrokovnost avtorjev specifikacije. Že v opisu razvoja informacijskega modeliranja je bilo prikazano, da so bili vsi poskusi definiranja informacijskega modela zgradb v okviru raziskovalnih projektov oz. neprofitnih združenj neuspešni (izjemo pri tem predstavlja model CIS2, ki je po nastal po aktivni vključitvi industrijskih partnerjev v sicer neuspešen raziskovalni projekt). Zato enega izmed upravičenih zaključkov naloge predstavlja naslednja trditev: tudi po več kot desetletnem obstoju IFC specifikacije njene implementacije – kljub podeljenim certifikatom skladnosti – ne zagotavljajo zadostne ravni interoperabilnosti. Slednja se enostavno ne more primerjati z rešitvami zaokroženega načrtovanja, ki jih ponujajo (kapitalsko) povezane programske hiše. Poglavitne pomanjkljivosti specifikacije, izpostavljene v tej nalogi, predstavljajo nezadostno upravljanje z identifikatorjem GUID, različni pristopi pri modeliranju ter nedoločeni pogledi na model.

Povsem nerealno je pričakovati, da se bodo navedene napake oz. nekonsistentnosti v IFC modelu in v implementacijah zgradb odpravile postopoma oz. same po sebi, kot so to

napovedali nekateri raziskovalci (Bazjanac, 2002). Desetletje obstoja specifikacije in ugotavljanje enakih napak, kot so navedene v prvih implementacijski poročilih, potrjuje navedeno trditev. Zato bi bilo namesto vključevanja novih domenskih razširitev IFC specifikacije bolj primerno na osnovi implementacijskih poročil oz. analiz opraviti ponoven pregled celotne IFC specifikacije, izdelati natančnejšo specifikacijo oz. navodila za implementacijo ter skupaj z razširitvami izdelati natančno opredeljene in vse do nivoja entitet (oz. eventualno vse do nivoja atributov) opredeljene poglede na model. Le tako izpopolnjena specifikacija bi lahko predstavljala osnovo kakovostnejših implementacij, sposobnih konkurirati lastniškimi rešitvam zaokroženega načrtovanja. Takšen pristop bi obenem omogočal implementacijo načrtovanega prototipa.

## 5.2 Povzetek bistvenih prispevkov naloge

- Izdelan je bil natančen pregled zgodovine informacijskega modeliranja v gradbeništvu. Predstavljeni so bili začetki rabe informacijskih in komunikacijskih tehnologij v gradbeništvu, posebej podrobno pa je bil izdelan pregled razvoja informacijskih modelov zgradb.
- Podrobneje je bil predstavljen informacijski model zgradb IFC. Poleg pregleda razvoja je bila predstavljena tudi struktura specifikacije in domenska razširitev na področje računske analize konstrukcije.
- Definirana sta bila dva nova »pogleda na model«, ki na osnovi IFC modela zgradb natančneje opredeljujeta izmenjavo informacij med arhitekturo in računskim modeliranjem ter računskim modeliranjem in računsko analizo. Pri določitvi pogledov so bili za zagotavljanje ustrezne robustnosti uporabljeni najnovejši izsledki raziskav na predstavljenem področju (»uporabni minimum«, FIOPE pristop).
- Na praktičnem primeru zaokroženega načrtovanja konstrukcij je bilo dokazano, da s trenutno veljavno specifikacijo informacijskega modela zgradb IFC (oz. s trenutno implementiranim razširjenim koordinacijskim pogledom) ni mogoče doseči zelene interoperabilnosti pri zaokroženem načrtovanju konstrukcij. Za doseganje cilja je potrebna natančnejša opredelitev nabora entitet ki bodo predstavljale predmet

zanimanja oz. izmenjave in sicer v obliki »uporabnega minimuma« ali v obliki pogledov.

- Za učinkovito obvladovanje informacijskega modela zgradb so bile na predlaganem primeru prikazane prednosti uporabe strežnika modela zgradb.
- Na primeru domene računskega modeliranja in računske analize konstrukcij je bila v nalogi praktično utemeljena nujnost zagotavljanja robustnosti specifikacije in modela pred njegovo širitvijo na vseobsegajoči opis življenjskega cikla zgradbe.
- Preskušeni so bili najbolj pogosto uporabljeni IFC vmesniki aplikacij za arhitekturno in računsko modeliranje konstrukcij. Z izpostavitvijo njihovih bistvenih pomanjkljivosti (npr. upravljanje z identifikatorjem GUID) se opozarja razvijalce programske opreme po nujnem izboljšanju njihovih produktov.
- Teoretično in praktično je bila utemeljena in prikazana ponovna raba informacij, ki izvirajo iz semantične narave informacijskega modela zgradb. Poleg implementacije prirejanja geometrije med arhitekturnimi modeli in modeli za analizo je v nalogi predlagan izviren avtomatiziran pristop pripisovanja vplivov na konstrukcijo. S prikazanim načinom prirejanja se znotraj informacijskih modelov zgradb poleg prirejanja geometrije z naslonitvijo na standarde s področja projektiranja konstrukcij delno avtomatizira tudi prirejanje obežbe.
- V nalogi je bil razvit prototip strežnika modela zgradb IFC. Pri izdelavi je bila uporabljena le prosto dostopna programska oprema ter odprti spletni komunikacijski protokoli. S prototipno implementacijo je bila postavljena osnova sodobnega prosto dostopnega IFC strežnika modela zgradb.

### **5.3 Nadaljnje raziskave**

Stopnja implementacije IFC specifikacije je ne glede na več kot desetletno zgodovino relativno skromna oziroma močno odstopa od pričakovanj. Vzroki za takšno stanje so znani in so bili predstavljeni tudi v tej nalogi. Namesto domenske širitve modela bi bilo bolj upravičeno vložiti napore v natančno opredelitev »uporabnih minimumov« ter nadalje v implementacijo predvidljivih rešitev v obliki vmesnikov oziroma v strežnike informacijskega

modela zgradb. Tudi v prihodnje gre pričakovati, da bo izmenjava čiste geometrije produkta ostala v domeni uveljavljenih formatov (npr. dwg). Nesmiselno je pričakovati, da bo na opisanem področju specifikacija IFC izpodrinila navedene uveljavljene načine zapisa. Priložnost za uveljavitev IFC specifikacije oz. za implementacije le-te predstavljajo področja, kjer dodatni pomen oz. semantika objektnih primitivov zagotavlja dodano vrednost pri zapisu. Zato bi morale biti raziskave na področju IFC specifikacije najprej usmerjene v iskanje oz. definicijo nedvoumnega »uporabnega minimuma« informacij, ki bi se ga na podlagi povratnih informacij iz prakse lahko ustrezno razširilo. Vključitev novih domen v model IFC bi morala biti zato že v osnovi pogojena z opredelitvijo »uporabnega minimuma« izmenjave.

Področje računske analize konstrukcij v informacijskem modelu zgradb IFC trenutno ni zadovoljivo pokrito (pomanjkljivosti razširitve ST4 so bile v nalogi podrobno izpostavljene). Podobno velja tudi za računsko modeliranje konstrukcij. Nadaljnje raziskave na obravnavanem področju bi morale nujno v navedenem vrstnem redu vsebovati vsaj:

- natančno opredelitev računskega modela in modela za analizo,
- ustrezno dopolnitev specifikacije računskega modela in modela za analizo za področje visokogradnje,
- glede na predlagane dopolnitve ustrezno korigirati v tej nalogi predlagane poglede na model,
- na podlagi implementacij oceniti ustreznost na osnovi uporabnih minimumov, definiranih pogledov na modele ter ustrezno korigirati.

V prototip vključeni algoritem predstavlja le eno izmed možnih preslikav med arhitekturnimi in računskimi modeli. Preslikave, pri katerih se med drugim spremenijo tudi dimenzije modela (npr. iz prostorske konstrukcije v ravninski računski model), zahtevajo vključitev kompleksnejših hevrističnih pravil. V nalogi je predstavljen le izsek mnogo kompleksnejšega področja, ki ga je možno označiti za samostojno raziskovalno področje.

Standard Evrokod v prvem delu natančno specificira vrste vplivov na konstrukcijo, ki jih je potrebno upoštevati pri računski analizi konstrukcije. Podobno kot definicija pogledov tudi definicija XML zapisa vplivov temelji na principu uporabnega minimuma. Temeljita analiza vseh predpisanih vplivov na konstrukcijo (ENV 1991-1-1) predstavlja osnovo za preslikavo standarda v XML zapis.



Funkcionalnost trenutno implementiranega prototipa je relativno omejena. Izboljšave bi bile potrebne predvsem na strežniški strani. Implementirati bi bilo potrebno protokol SOAP, ki bi z ustreznim klientom omogočil delno izmenjavo informacij oz. modelov.

Dodano vrednost koncepta ponovne uporabe informacij, zajetih v informacijskih modelih zgradb, bi bilo z nadaljnjimi raziskavami možno dokazati tudi v preostalih domenah (npr. arhitektura – toplotni odziv zgradb). Uspeh predlaganih raziskav bo v veliki meri odvisen od podrobnosti specifikacije na obravnavanem področju ter od ustreznosti pripadajočih pogledov na model.

## 6 POVZETEK

Gradbeništvo v najširšem pomenu besede se običajno kvalificira kot klasični primer razdrobljene industrije z unikatnimi produkti, kjer je izmenjava informacij med posameznimi akterji pomembno, a hkrati netrivialno opravilo. Kompleksnost sodobnih gradbenih objektov zahteva sodelovanje strokovnjakov različnih strok. Ne glede na dejstvo, da vsi obravnavajo isti objekt oz. zgradbo, ima vsak izmed udeležencev svoj pogled nanj(o), svojo specializirano programsko opremo in posledično svoj digitalni model zgradbe. V izogib neskladnosti med posameznimi modeli ter napakam pri transformacijah med modeli so se raziskave na področju gradbene informatike usmerile v celovito informacijsko modeliranje zgradb. Pri takšnem pristopu si model zgradbe delijo vsi, ki informacije o zgradbi ustvarjajo in potrebujejo.

Disertacija se osredotoča na trenutno najbolj uveljavljen informacijski model zgradb, to je na temeljne razrede za industrijo (IFC - Industry Foundation Classes). Ker nedvoumna predstavitev in deljenje informacij predstavljata bistveni element informacijskega modeliranja, izbrani pristop temelji na sistematičnem kronološko razvojnem opisu splošnega informacijskega modeliranja s posebnim poudarkom na standardu ISO 10303 (STEP), to je specifikaciji za opis in izmenjavo informacij o produktu skozi njegov življenjski cikel.

Glede na razvojno stopnjo specifikacije IFC se naloga omejuje na domensko specifična podmodela arhitekture in računske analize konstrukcij ter na relacije med njima. V klasičnem delotoku načrtovanja izdelkov industrije ki oblikuje grajeno okolje gradbenik konstruktor na podlagi arhitekturnega modela izdelava ustrezen računski model, to je izbere primerno nosilno konstrukcijo za prevzem zunanjih vplivov ter določi potrebne dimenzije nosilnih elementov. Ker se pri modeliranju uporablja enotni informacijski model je smiselno preučiti relacije in možna prirejanja med modeloma. V disertaciji je postavljena hipoteza da je predstavljeni problem ob uporabi informacijskega modela zgradb IFC ter računalniškega kodiranja hevrističnih pravil možno poenostaviti ter delno avtomatizirati.

Obstoječe študije preslikav med arhitekturnimi in računskimi aspekti temeljijo izključno na prirejanju geometrije. V skladu s hipotezo naloge predlagana rešitev za meddomensko sodelovanje temelji na semantičnem kontekstu informacijskih modelov zgradb, ki poleg izmenjave geometrije omogoča vključitev oz. delno avtomatizacijo preostalih aspektov

izmenjave informacij med arhitektom in gradbenikom konstruktorjem. Na primeru prirejanja vplivov na konstrukcijo je prikazana praktična uporaba semantike informacijskih modelov zgradb. Predlagani sistem zaradi zagotavljanja konsistentnosti temelji na spletnem strežniku, ki edini omogoča ustrezno upravljanje z modelom. Hipotezo potrjuje tudi delna prototipna implementacija, in sicer z uporabo izključno odprtokodnih rešitev in odprtokodnih spletnih komunikacijskih protokolov.

Predlagana arhitektura in demonstracijski prototip prikazujeta eno izmed številnih možnosti, ki jih ponuja strukturirana izmenjava informacij, temelječa na informacijskem modelu zgradb IFC. V raziskavi je bila podana hipoteza načeloma sicer potrjena, vendar je bilo pri izdelavi prototipa identificiranih kar nekaj težav: nepopolnost specifikacije na obravnavanih področjih, nedoločen nabor entitet za izmenjavo oz. nedoločeni pogledi na model, omejenost obstoječih IFC vmesnikov ter njihove nepopolne oz. nekoherentne implementacije. Zaradi navedenih pomanjkljivosti specifikacije in pripadajočih vmesnikov načrtovanega prototipa žal ni bilo mogoče izdelati v predvidenem obsegu.

## 7 SUMMARY

Architecture, engineering, construction and facility management sector is commonly qualified as a crumbled industry branch with unique products where information exchange is important but at the same time nontrivial procedure. The complexity of modern AEC-FM products requires collaboration of many domain specific experts. Regardless of the common point of interest, each participant has its own domain specific view on the building, its own specialized software and consequently its own digital building model. In order to avoid discordance between individual models and faults in model transformations, the contemporary construction informatics focuses on structured information exchange where all relevant information about the building in its lifecycle are gathered in a single building information model. Relevant information can therefore be used by everyone involved.

Presented thesis focuses on the Industry Foundation Classes (IFC), currently the most popular building information model. Selected approach is based on systematic, chronological development description since unambiguous information description and information sharing present the key elements in the information modeling. The role of ISO 10303 (STEP) specification in product description and information exchange is specially emphasized.

Regarding to the IFC specification development stage presented thesis is limited on the domain specific architectural and structural submodels and on corresponding relations between them. Within the common AEC-FM workflow the architectural design presents the origin point of structural design. Structural engineer has to select an appropriate load bearing structure which would be able to take over imposed external loads. All relevant structural design information is gathered in the structural model. It is reasonable to examine the relations between presented submodels when a unified building information model (IFC) is used. This thesis defends the hypothesis that presented problem can be simplified and partly automated if IFC specification and appropriate computer encoded heuristic rules are used.

All so far existing studies are based only on simple geometry mapping. According to the hypothesis suggested solution for inter domain collaboration is based on a semantic context of building information models. Beside geometry exchange presented approach allows inclusion or partial automation of other relevant aspects in information exchange between architects and

structural engineers. To prove the concept, capabilities of semantic based information exchange is demonstrated on a simple example of semi-automatic external load assignment. Proposed system is based on the IFC web server which offers the appropriate level of management and at the same time assures model consistency. The prototype is implemented using only non-proprietary tools and non-proprietary web based communication protocols.

Proposed system for collaborative building design between architects and structural engineers and the partially implemented prototype present only one of many possibilities offered by structured IFC based information exchange. Although presented research work mainly proves the thesis hypothesis, the difficulties identified in prototype implementation (incomplete IFC specification, indefinite entities in information exchange – model views, limitations of current IFC interfaces – incomplete and incoherence implementations) confirms the anticipations about immature IFC specification and appurtenant interfaces. Consequently currently proposed system can not be fully implemented.

## LITERATURA

- Adachi, Y. 2002. Overview of IFC model server framework. V: Turk, Ž., Scherer, R. J. (ur.). Proceedings of the 4th European Conference on Product and Process Modelling – ECPPM 2002 – eWork and eBusiness in Architecture, Engineering and Construction, 8. – 11. September 2002. Portorož, Slovenia. Leiden, Netherlands, Taylor & Francis / Balkema: str. 367-372.
- Amor, R., Ma, H. 2006. Preservation of meaning in mapped IFCs. V: Martinez, M., Scherer, R. J. (ur.). Proceedings of the 6th European Conference on Product and Process Modelling – ECPPM 2006 – eWork and eBusiness in Architecture, Engineering and Construction, 13. – 15. September 2006. Valencia, Spain. Leiden, Netherlands, Taylor & Francis / Balkema: str. 223-236.
- Augenbroe, G. 1995. The COMBINE project: A global assessment. V: Fischer, M. (ur.). Modeling of Buildings Through Their Life-Cycle, CIB W78 Workshop Proceedings, 22. – 24. August 1995. Stanford, USA: str. 163-171.  
<http://itc.scix.net/cgi-bin/works/Show?w78-1995-163> (22. 1. 2006).
- Backas, S. 2002. SPADEX – Project final report.  
[http://cic.vtt.fi/vera/Projects/e\\_spadex.htm](http://cic.vtt.fi/vera/Projects/e_spadex.htm) (15. 2. 2006).
- Bazjanac, V. 2002. Early lessons from deployment of the IFC compatible software, Keynote paper. V: Turk, Ž., Scherer, R. J. (ur.). Proceedings of the 4th European Conference on Product and Process Modelling – ECPPM 2002 – eWork and eBusiness in Architecture, Engineering and Construction, 8. – 11. September 2002. Portorož, Slovenia. Leiden, Netherlands, Taylor & Francis / Balkema: str. 9-16.
- Behrman, W. 2002. Best practices for the development and use of XML data interchange standards, CIFE Technical Report 131, Stanford University, Department of Civil and Environmental Engineering, Center for Integrated Facility Engineering: 27 str.  
<http://cife.stanford.edu/online.publications/TR#131.pdf> (4. 7. 2007).
- Björk, B.-C. 1992. A conceptual model of spaces, space boundaries and enclosing structures. Automation in Construction, 1, 3: 193-214.

Björk, B.-C. 1995. Requirements and information structures for building product data model. Dissertation for the degree of Doctor of Technology. Espoo, Finland, VTT Building Technology: 88 str.

<http://lib.tkk.fi/Diss/199X/isbn9513860302/> (4. 7. 2007).

Björk, B.-C., Penttilä H. 1989. A scenario for the development and an implementation of the building product model standard. *Advances in Engineering Software*, 11, 4: 176-187.

BLIS, 2004.

<http://www.blis-project.org> (25. 1. 2006).

Cerovšek, T., Turk, Ž., Duhovnik, J. 2002. Informacijski modeli zgradb. V: Saje, F. in Lopatič, J. (ur.). Zbornik 24. zborovanja gradbenih konstrukterjev, Bled, Festivalna dvorana, 14. – 15. november 2002. Ljubljana, Slovensko društvo gradbenih konstruktorjev: str. 311-318.

Cerovšek T., Turk, Ž. 2002. ICCI ICT Common Glossary. ICCI Project IST-2000-33022, Deliverable D211.

<http://itc.fgg.uni-lj.si/projects/icci/glossary.cgi> (4. 12. 2007).

Cerovšek, T. 2003. Distribuirana računalniško podprta gradnja pri pogojih necelovitosti. Doktorska disertacija. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo, Oddelek za gradbeništvo, Konstrukcijska smer: 307 str.

Chen, P.-H, Cui, L., Wan, C., Yang, Q., Ting, S. K., Tiong, R. L. K. 2005. Implementation of IFC-based web server for collaborative building design between architects and structural engineers. *Automation in Construction*, 14, 1: 115-128.

Dayal, M. 2004. Analyse des 3D – datenaustausches via IFC-modell am beispiel komplexer objekt-dokumentation in der automobilindustrie mit dem ziel der optimierung von planungsprozessen. Diploma thesis. München, Technische Universität München: 159 str.  
<http://www.graphisoft-muenchen.de/pub/Dayal-IFC.pdf> (21. 7. 2007).

Debras, P., Zarl, A., Amar, V., Poyet, P. 1997. The distributed information service in the VEGA project: An approach towards the harmonisation of STEP, SGML and EDIFACT information standards for the support of integrated and distributed construction project information systems. V: Amor, R. (ur.). CIB W78 Workshop Proceedings, 9. – 11. July 1997. Cairns, Queensland, Australia: str. 123-137.

[http://itc.scix.net/cgi-bin/works/Show?\\_id=w78-1997-123](http://itc.scix.net/cgi-bin/works/Show?_id=w78-1997-123) (4. 7. 2007).

Deng, X.Y., Chang, T.-Y. P. 2006. Construction of structural model from IFC-based architectural model. V: Proceedings of joint international conference on computing and decision making in civil engineering, 14. – 16. June 2006. Montreal, Canada.

[www.icccbexi.ca/PDF/tf577.pdf](http://www.icccbexi.ca/PDF/tf577.pdf) (14. 8. 2007).

Duhovnik, J. 1998. Statika linijskih konstrukcij 1. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 224 str.

Eastman, C. 1999. Building product models: Computer environments supporting design and construction. Florida, USA, CRC Press, Boca-Raton: 424 str.

Eastman, C. 2000. Overview of CIS2 (extended). Georgia Tech Document Library: 33 str.

<http://www.coa.gatech.edu/~aisc/index.php?cat1=3#Tutorial%20List> (5. 8. 2006).

Ekholm, A. 1994. A systemic approach to building modelling – Analysis of some object oriented building product models. V: Proceedings of CIB W78 Workshop on Computer Integrated Construction, 22. – 24. August 1994. Esbo, Finland.

[http://itc.scix.net/cgi-bin/works/Show?\\_id=w78-1994-30](http://itc.scix.net/cgi-bin/works/Show?_id=w78-1994-30) (5. 9. 2007).

ENV 1991-1-1. 2004. Evrokod 1: Vplivi na konstrukcije – 1-1. del: Splošni vplivi – gostote, lastna teža, koristne obtežbe stavb: 44 str.

Fischer, M., Cam, K. 2002. PM4D Final Report. CIFE Technical report 143. Stanford University, Department of Civil and Environmental Engineering, Center for Integrated Facility Engineering: 50 str.

<http://cife.stanford.edu/online.publications/TR143.pdf> (5. 6. 2006).

Flower, J. 1995. STEP for data management, exchange and sharing. Twickenham, UK, Technology Appraisals Ltd.: 226 str.



- Forester, J. 2002. Information modelling – Introduction, history and languages. Seminar on Data Modelling for Building Operation, ASHARE Annual Meeting. Gaithersburg, ZDA, National Institute of Standards and Technology: 65 str.  
[www.nist.gov/tc75/ForresterASHRAE%20Information%20Modeling%20Seminar.pdf](http://www.nist.gov/tc75/ForresterASHRAE%20Information%20Modeling%20Seminar.pdf), (20. 6. 2007)
- Froese, T. 1996. Models of construction process information. *Journal of Computing in Civil Engineering*, 10, 3: 183–193.
- Gallaher, M. P., O'Connor, A. C., Dettbarn, J. L., Gilday, L. T. 2003. Cost analysis of inadequate interoperability in the U.S. capital facilities industry. Research report. Maryland, USA, National Institute of Standards and Technology – US Department of Commerce Technology of Standards and Technology, Advanced Technology Program, Information Technology and Electornics Office Gaithersburg: 210 str.  
<http://www.bfrl.nist.gov/oe/publications/gcrs/04867.pdf> (25. 1. 2008).
- Geiger, A. 2001. Ausführliche testergebnisse zur IFC-schnittstelle version 1.5.1. datenaustausch von einfachen wandobjekten bis hin zu komplexen beispielen aus dem CAD-alltag. Karlsruhe, Forschungszentrum Karlsruhe Institut für Angewandte Informatik: 46 str.  
<http://www.iai.fzk.de/www-extern/index.php?id=1134&L=1> (24. 6. 2007).
- Ghang, L. 2004. A new formal and analytical process to product modelling (PPM) method and its application to the precast concrete industry. Doctoral thesis. Atlanta, Georgia Institute of Technology: 271 str.
- Goh, B. H. 2005. IT barometer 2003: Survey of the Singapore construction industry and a comparison of results. *ITcon*, 10(2005): 1–13,  
<http://www.itcon.org/2005/1> (11. 3. 2007).
- Goldstein, B. L. M., Kemmerer, S. J., Parks, C. H. 1998. A Brief History of Early Product Data Exchange Standard. Research report. Maryland, USA, National Institute of Standards and Technologies: 27 str.  
<http://www.mel.nist.gov/msidlibrary/summary/9828.html> (1. 8. 2007).
- Graphisoft, 2004. Graphisoft IFC support.  
<http://www.graphisoft.com/products/ifc> (1. 8. 2006).

Haymaker, J., Sutler, B., Kuntz, J., Fischer, M. 2003. PRESPECTORS: Automating the construction and coordination of multidisciplinary 3D design representation. CIFE technical report 145. Stanford University, Department of Civil and Environmental Engineering, Center for Integrated Facility Engineering: 11 str.

<http://cife.stanford.edu/online.publications/TR145.pdf> (15. 1. 2006).

Hietanen, J. 2006. IFC model view definition format. IAI – International Alliance for Interoperability.

<http://www.blis-project.org/IAI-MVD/> (24. 9. 2007).

IAI, 2008. International Alliance for Interoperability.

<http://www.iai-international.org/> (22. 12. 2007).

IAI-MSG, 1996a. The EXPRESS definition language for IFC development.

[http://www.iai-international.org/Model/IFC\(ifcXML\)Specs.html](http://www.iai-international.org/Model/IFC(ifcXML)Specs.html) (22. 1. 2006).

IAI-MSG, 1996b. Data modelling using EXPRESS-G for IFC Development.

[http://www.iai-international.org/Model/IFC\(ifcXML\)Specs.html](http://www.iai-international.org/Model/IFC(ifcXML)Specs.html) (22. 1. 2006).

IAI-MVD, 2007. International Alliance for Interoperability – Model View Definition Group.

<http://www.blis-project.org/IAI-MVD/> (23. 2. 2007).

IAI-NA, 2007. International Alliance for Interoperability – North American Chapter.

<http://www.buildingsmartalliance.org/> (23. 2. 2007).

IGES. 2006. IGES – Initial Graphics Exchange Specification IGES 5.3. National Institute of Standards and Technologies.

<http://ts.nist.gov/standards/iges/> (22.5.2006)

IGES5. 2007. The IGES 5x Preservation Society (IPS).

<http://www.iges5x.org/> (14. 2. 2007)

ISO 10303-42. 1992. Industrial automation systems and integration – Product data representation and exchange – Part 42: Integrated generic resources: Geometric and topological representation: 328 str.

ISO 10303-21. 1994. Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure: 72 str.

ITAA, 2007. Information Technology Association of America.

<http://www.ita.org/> (22. 5. 2007).

Junge, R., Beetz, K., Liebich, T. 1997. Product data model and implementation strategies for a distributed environment – Developments in the VEGA project. V: Amor, R. (ur.). CIB W78 Workshop Proceedings, 9. – 11. July 1997. Cairns, Queensland, Australia: str. 181-198.  
<http://itc.scix.net/cgi-bin/works/Show?w78-1997-181> (30.6.2006).

Katranuschkov, P., Scherer, R. J. 1996. Schema mapping and object matching: a STEP-based approach to engineering data management in open integrated environments. V: Proceedings of CIB W78 Workshop on Computer Integrated Construction, 10. – 12. June 1996. Bled, Slovenia.

<http://itc.scix.net/cgi-bin/works/Show?w78-1996-335> (4. 12. 2007).

Khemlani, L. 2004. The IFC building model: A look under the hood. AECBytes – Analysis, Research and Reviews of AEC Technology.

<http://www.aecbytes.com/feature/2004/IFCmodel.html> (13. 8. 2007).

Kiviniemi, A., Fischer, M., Bazjanac, V. 2005. Integration of multiple product models – IFC model servers as potential solution. V: Scherer, R. J., Katranuschkov, P., Schapke, S. E. (ur.). Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction, 19. – 21. June 2005. Dresden, Germany: str. 37-40. <http://itc.scix.net/data/works/att/w78-2005-KN-w2-1-Kiviniemi.pdf> (30.5.2007).

Kos, L. 2000. Optimizacija razreza linijskih elementov. Interno poročilo 1/1-2000. Ljubljana, Univerza v Ljubljani, Fakulteta za strojništvo, Laboratorij za CAD – LECAD: 32 str.

Lehtinen, S., Hietanen, J. 2007. The useful minimum. Poročilo. Zasebna korespondenca o inovativnem pristopu definiranja IFC pogledov na model.

Lehtinen, S. 2005. Generic AEC/FM view description: Architectural design to structural design.

<http://www.blis-project.org/IAI-MVD/MVD> (9. 12. 2007).

Lieblich, T., Wix., J. 2000. IAI technical guide. International Alliance for Interoperability. [http://www.iai-international.org/Model/files/20030630\\_Ifc2x\\_ModelImplGuide\\_V1-6.pdf](http://www.iai-international.org/Model/files/20030630_Ifc2x_ModelImplGuide_V1-6.pdf) (25. 5. 2006).

Liebich, T., Wix, J. 2002. Standard analysis – Current AEC situation – building models.  
ProdAEC Project. IST-2001-32035, Deliverable D4.1.2: 78 str.

Lieblich, T. 2004, ifcXML Implementation guide. International Alliance for Interoperability.  
[http://www.iai-international.org/Model/files/20030630\\_Ifc2x\\_ModelImplGuide\\_V1-6.pdf](http://www.iai-international.org/Model/files/20030630_Ifc2x_ModelImplGuide_V1-6.pdf)  
(10. 4. 2007).

Nguyen, T.-H., Oloufa, A. A. 2001. Computer – generated building data: Topological  
information. *Journal of Computing in Civil Engineering*, 15, 4: 268-274.

Nisbet, T., Liebich, T. 2007. ifcXML implementation guide, Version 2.0. International  
Alliance for Interoperability.  
[http://www.iai-international.org/IFCXML/Ifc2x/Ifc2x\\_FINAL](http://www.iai-international.org/IFCXML/Ifc2x/Ifc2x_FINAL), (12. 12. 2007).

Nour, M. 2007. Manipulating IFC sub-models in collaborative teamwork environment. V:  
Rebolj, D. (ur.). *Proceedings of the 24th CIB W78 Conference on Information Technology in  
Construction*, 26. – 29. June 2007. Maribor, Slovenia. Faculty of Civil Engineering,  
University of Maribor: str. 111–117.

Opendesign, 2006. Why isn't DXF good enough, Open Alliance White Paper.  
<http://www.opendesign.com/about/whtpaper> (12. 12. 2007).

Pazlar, T., Dolenc, M., Duhovnik, J. 2004. Implementation of the ICT in the Slovenian AEC  
sector. V: Dikbas, A., Scherer, R. J. (ur.). *Proceedings of the 5th European Conference on  
Product and Process Modelling – ECPPM 2004 – eWork and eBusiness in Architecture,  
Engineering and Construction*, 8. – 10. September 2006. Istanbul, Turkey. Leiden,  
Netherlands, Taylor & Francis / Balkema: str. 161–169.

Pazlar, T., Turk, Ž. 2006. Analysis of the geometric data exchange using the IFC. V:  
Martinez, M., Scherer, R. J. (ur.). *Proceedings of the 6th European Conference on Product  
and Process Modelling – ECPPM 2006 – eWork and eBusiness in Architecture, Engineering  
and Construction*, 13. – 15. September 2006. Valencia, Spain. Leiden, Netherlands, Taylor &  
Francis / Balkema: str. 165-172.

Plokker, W., Augenbroe, G. 1994. PDI tools in COMBINE project. V: *Proceedings of CIB  
W78 Workshop on Computer Integrated Construction*, 22. – 24. August 1994. Esbo, Finland:  
<http://itc.scix.net/cgi-bin/works/Show?w78-1994-25> (4. 10. 2007).

Reed, K. A. 1988. Product modelling of buildings for data exchange standards: from IGES to PDES/STEP and beyond. V: Christiansson, P., Karlsson, H. (ur.). Proceedings of CIB W74 and CIB W78 Workshop, 25. – 27. October 1988. Lund, Sweden. Lund University and the Swedish Building Centre: str. 157-164.

Rivald, H. 2000. A survey on the impact of information technology in Canadian architecture, engineering and construction industry. ITcon, 5(2000): 37–65.

[http://www.itcon.org/cgi-bin/works/Show?2000\\_3](http://www.itcon.org/cgi-bin/works/Show?2000_3) (23.6. 2007).

Romberg, R., Niggel, A., van Treeck, C., Rank, E. 2004. Structural analysis based on the product model standard IFC. V: Proceedings of International Conference on Computing in Civil and Building Engineering – ICCCB, 2. – 4. July 2004. Weimar, Bauhaus – Universität Weimar.

<http://e-pub.uni-weimar.de/volltexte/2004/251/> (10. 11. 2007).

Ronneblad, A. (2003). Product models for concrete structures: standards, applications and implementations. Licentiate thesis. Lulea University of Technology, Department of Civil and Mining Engineering, Division of Structural Engineering: 128 str.

<http://epubl.ltu.se/1402-1757/2003/22/index.html> (22. 8. 2006).

Ronneblad, A., Olofsson, T. 2003. Application of IFC in design and production of precast concrete constructions. ITcon, 13(2003): 167-180.

<http://www.itcon.org/2003/8> (13. 11. 2006).

Rooney, J. 1987. Principles of computer-aided design. Pitman Publishing and the Open University: 341 str.

Samuelson, O. 2002. IT barometer 2000 – The use of IT in the Nordic construction industry. ITcon, 7(2002): 1-26.

<http://www.itcon.org/2002/1> (13. 11. 2007).

Samuelson, O. 2008. The IT-barometer – a decade's development of IT use in the Swedish construction sector. ITcon, 13(2008): 1-19.

<http://www.itcon.org/2008/1> (5. 4. 2008).

Storer, G., Kazi, S. A., Wilson, I., Zarli, A. 2004. ICCI book – Building a better future. ICCI – European project IST-2001-33022, deliverable D524: 45 str.

<http://cic.vtt.fi/projects/icci/results.html> (23. 3. 2007).

Tolk, A., Muguira, J. A. 2003. The levels of conceptual interoperability model. Simulation Interoperability Workshop, 13. – 15. September 2003. Orlando, Florida.

[www.sei.cmu.edu/isis/pdfs/tolk.pdf](http://www.sei.cmu.edu/isis/pdfs/tolk.pdf) (12. 10. 2007).

Tolman, F. 1999. Product modelling standards for the building and construction industry: past present and future. *Automation in Construction*, 8(1999): 227–235.

Turk, Ž. 1992. Okolje za računalniško integrirano projektiranje gradbenih konstrukcij.

Doktorska disertacija. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo, Oddelek za gradbeništvo, Konstrukcijska smer: 176 str.

Turk, Ž. 1992a. Modeliranje gradbenih produktov. V: Duhovnik J. (ur.). Zbornik 6. seminarja Računalnik v gradbenem inženirstvu. Ljubljana, Univerza v Ljubljani, Fakulteta za arhitekturo, gradbeništvo in geodezijo, Oddelek za gradbeništvo in geodezijo, Inštitut za konstrukcije, potresno inženirstvo in računalništvo FAAG Ljubljana: 9–16.

Turk, Ž. 1999. Constrains of product modelling approach in buildings. V: Lacasse, M. A., Vainer, D. (ur.). *Proceedings of 8th International Conference on Durability of Building Materials and Components*, 30. May – 3. June 1999. Vancouver, Canada: 2776-2787.

<http://www.zturk.com/db/use/works> (5. 5. 2007).

Turk Ž. 2001. E-delo in e-poslovanje v gradbeništvu in arhitekturi. Seminar. Ljubljana, Gradbeni center ZRMK, 15. 11. 2001.

<http://www.zturk.com/pouk/zrmk/edelo1.pdf> (4. 4. 2007).

Weise, M., Katranuschkov, P., Liebich, T. 2002: IAI Project ST4, Volume III – Structural analysis domain. Dresden, Germany, Lehrstuhl für Computeranwendung im Bauwesen, Technische Universität: 59f.

Weise, M., Katranuschkov, P., Liebich, T., Scherer, R. J. 2003. Structural analysis extension of the IFC modeling framework, *ITcon*, 14(2003): 181–200.

[http://www.itcon.org/cgi-bin/works/Show?2003\\_14](http://www.itcon.org/cgi-bin/works/Show?2003_14) (5.6. 2006).

ZGO. 2002. Zakon o graditvi objektov,

<http://www.uradni-list.si/1/objava.jsp?urlid=2002110&stevilka=5387> (2. 5. 2007)

## **PRILOGA A: kratice, okrajšave**

**AAM** – Application Activity Model: Aplikacijski aktivnostni model.

**AEC/FM** – Architecture, Engineering, Construction, Facilities Management: Arhitektura, inženirstvo, gradbeništvo, upravljanje z zgradbami. Termin označuje gradbeništvo v najširšem pomenu besede.

**AIM** – Application Activity Model: Aplikacijski aktivnostni model.

**AISC** – American Institute of Steel Constructions: Ameriški inštitut za jeklene konstrukcije.

**ANSI** – American National Standardization Institute: Ameriški nacionalni inštitut za standardizacijo.

**API** – Application Programming Interface: Programski vmesnik.

**ARM** – Application Reference Model: Aplikacijski referenčni model.

**ATLAS** – Architecture, Methodology and Tools for Computer Integrated Large Scale Engineering: Arhitektura, metodologija in orodja za računalniško integrirano gradnjo obsežnih projektov.

**BCCM** – Building Construction Core Model: Jedrni model za gradbeništvo.

**BIM** – Building Information Modelling: Informacijski model zgradbe oz. digitalna predstavitev AEC/FM objektov.

**BLIS** – Building Lifecycle Interoperable Software: Projekt, namenjen široki implementaciji specifikacije IFC.

**Brep** – Boundary Representaton: Mejna robna predstavitev.

**BSM** – Building System Model: Sistemski model zgradb.

**CAAD** – Computer Aided Architectural Design: Računalniško podprto načrtovanje arhitekture.

**CAD** – Computer Aided Design: Računalniško podprto načrtovanje ali računalniško podprto konstruiranje.



**CAM** – Computer Aided Manufacturing: Računalniško podprta proizvodnja.

**CAM-I** – Computer Aided Manufacturing – International: Mednarodno združenje za računalniško podprto proizvodnjo.

**CAX** – Computer Aided X: Termin označuje računalniško podprto načrtovanje v izbrani panogi, pri čemer »X« označuje izbrano panogo.

**COMBI** – Computer Integrated Object Oriented Model for the Construction Industry: Računalniško integriran objektno orientiran model za gradbeništvo.

**COAST** – COrba Access to Step models: Platforma za povezovanje porazdeljenih CAD aplikacij.

**CORBA** – Common Object Request Broker Architecture: Objektni model in specifikacija za razvoj porazdeljenih aplikacij.

**CIC** – Computer Integrated Construction: Računalniško integrirana gradnja.

**CIS** – cimSteel Integration Standard: cimSteel integracijski standard.

**COMBINE** – Computer Models for the Building Industry in Europe: računalniški modeli za evropsko gradbeno industrijo.

**DES** – Data Exchange System: Sistem za izmenjavo podatkov.

**DOM** – Document Object Model: Aplikacijski programski vmesnik (za HTML in XML dokumente).

**DTD** – Document Type Definition: Definicija vrste podatkov.

**DTF** – Design Tool Function: Stopnja interoperabilnosti izbranega orodja.

**DLL** – Dynamic Link Library: Dinamično povezljiva knjižnica.

**DDL** – Data Definition Language: Definicija podatkovnega jezika.

**DXF** – Drawing eXchange Format: Format za izmenjavo načrtov.

**EPISTLE** – The European Process Industries STEP Tehnical Liaison Executive: Združenje za razvoj specifikacij in standardov za izboljšanje učinkovitosti upravljanja in izmenjave informacij v življenjskem ciklu produkta.

**FU** – Functional unit: Funkcijska enota.

**GARM** – General AEC Reference Model: Splošni AEC referenčni model.

**GUI** – Graphical User Interface: Uporabniški grafični vmesnik.

**GUID** – Globally Unique Identifier: Globalni unikatni identifikator.

**HPS** – Harmonization of Product Data: Harmonizacija podatkov o produktu.

**HTML** – Hypertext Markup Language: Jezik za opis hipertekstnih dokumentov.

**HTTP** – Hypertext Transfer Protocol: Protokol za prenos hipertekstnih dokumentov.

**HVAC** – Heating, Ventilation, Air Condition: Ogrevanje, prezračevanje, klimatizacija.

**IAI** – International Alliance for Interoperability: Mednarodno združenje za inteoperabilnost.

**ICAM** – Integrated Computer Aided Manufacturing: Integrirana računalniško podprta proizvodnja.

**ICT** – Information and Communication Technologies: Informacijske in komunikacijske tehnologije.

**IDM** – Integrated Data Model: Integriran podatkovni model.

**IDL** – Interface Description Language: Jezik za opis vmesnika.

**IGES** – Initial Graphics Exchange Specification: Začetna specifikacija za izmenjavo grafike.

**IIBDS** – Intelligent Integrated Building Design System: Inteligentni integrirani sistem za načrtovanje zgradb.

**IFC** – Industry Foundation Classes: Temeljni razredi za industrijo.

**IPIM** – Integrated Product Information Model: Integriran produktni informacijski model.

**IRMA** – Integration Reference Model Architecture: Arhitektura integriranega referenčnega modela.

**IT** – Information Technology: Termin informacijska tehnologija označuje študije, načrtovanje, razvoj, implementacijo in upravljanje vseh vrst računalniških sistemov še posebej programske ter strojne opreme s poudarkom na varni pretvorbi, shranjevanju, zaščiti, procesiranju, oddaji in sprejemu informacij (ITAA, 2007). V zadnjih letih se precej pogosto uporablja termin ICT (Information and Communication Technologies), ki razširja doseg še na področje elektronskih komunikacij.

**ITC** – Information Technology in Construction: Informacijske tehnologije v gradbeništvu.

**IMS** – IFC Model Server: Strežnik modela zgradb IFC.

**ISO** – International Organization for Standardization: Mednarodna organizacija za standardizacijo.

**ISO 10303** – STandard for the Exchange of Product model data (STEP): Standard za izmenjavo produktnih modelov.

**IIS** – Internet Information Service: Skupina internetnih strežnikov (HTTP, FTP) z nekaterimi dodatnimi zmožnostmi (Windows operacijski sistemi).

**LISI** – Levels of Information System Interoperability: Stopnja interoperabilnosti informacijskih sistemov.

**LSI** – Large Scale Engineering Projects: Obsežni inženirski projekti.

**NIST** – National Institute of Standardization and Technology: Ameriški nacionalni inštitut za standardizacijo in tehnologije.

**OOCAD** – Object Oriented Computer Aided Design: Objektno orientirano računalniško podprto načrtovanje ali računalniško podprto konstruiranje.

**PDDI** – Product Definition Data Interface: Vmesnik za izmenjavo podatkov o produktu.

**PDES** – Product Data Exchange Specification: Specifikacija za izmenjavo podatkov o produktu.

**PDM** – Product Data Model: Produktni podatkovni model.

**PDU** – Product Definition Unit: Produktna definicijska enota.

**SDAI** – Standard Data Access Interface: Standard ISO 10303-22 za dostop do podatkov.

**SOAP** – Simple Object Access Protocol: Standard za spletne storitve, temelječe na XML.

**SQL** – Structured Query Language: Jezik za poizvedovanje po relacijskih zbirkah podatkov.

**ST4** – Structural Analysis Model and Steel Constructions: Razširitev IFC specifikacije na področje računske analize konstrukcij in jeklenih konstrukcij.

**STEP-CDS** – STEP Construction Drawing Subset: STEP – Nabor načrtov za gradbeništvo.

**TS** – Technical Solution: Tehnična rešitev.

**UFO** – Units of Functionality: Enote funkcionalnosti.

**UoD** – Universe of Discourse: Termin opisuje vsa možna stanja objektov realnega sveta vključno z časovno komponento.

**WSDL** – Web Service Description Language: Jezik za opis spletnih storitev.

**W3C** – World Wide Web Consortium: Standardizacijsko telo za spletne standarde.

**XML** – Extensible Markup Language: Razširljiv označevalni jezik.

**XSD** – XML Schema Definition: Definicija XML sheme predstavlja XML dokument, ki predpisuje omejitve pri elementih oz. atributih, razmerja, podatkovne tipe in podobno.

**XBF** – Experimental Boundary File: Poskusni robni model.

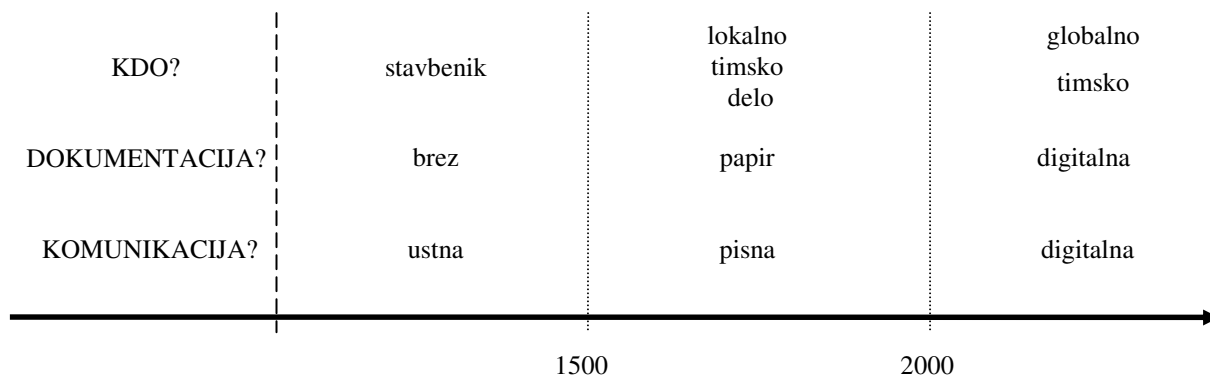


## PRILOGA B: Pregled informacijskega modeliranja

### B.1 Mejniki informacijskega modeliranja

Zgodovina znanosti in tehnike beleži kar nekaj ključnih mejnikov s pomembnejšim vplivom na celotno AEC/FM industrijo. Celovit pregled nad razvojem je podan v ICCI knjigi (Storer et al., 2004), na tem mestu so izpostavljene le poglavitne komunikacijske revolucije s pomembnejšim vplivom na AEC/FM sektor (slika 41).

- Zapis: pred približno 3000 leti se za predstavitev zamisli graditeljev prvič uporabi zapis s simboli. Kot medij zapisa so bili uporabljni kamen, glinene plošče ter papirus.
- Papir: raba papirja ter iznajdba tiska omogočita vzpostavitev informacijskega toka med vedno številčnejšimi udeleženci v življenjskem ciklu zgradbe. Uveljavitev cenenegega ter enostavno prenosljivega medija hkrati predstavlja pomemben element pri uveljavitvi timskega dela.
- Elektronski mediji so z uporabo telefona v poznem 19. stoletju izredno pospešili hitrost komuniciranja. Po stoletju razvoja elektronskih komunikacij je mogoče pravno veljavno prenašati vse dele projektne dokumentacije.
- Digitalizacija z univerzalnostjo dostopa do zapisane vsebine in vključitvijo interdisciplinarnih aspektov prestavlja enakovredno prelomnico predhodno navedenim mejnikom kljub ohranitvi nosilca informacij.



Slika 41: Komunikacijske revolucije in gradbeništvo

Figure 41: Communication revolutions and construction

Pregled komunikacijskih revolucij očitno predstavlja nekoliko drugačen pogled na razvoj družbe, kot je sicer splošno uveljavljen in poznan iz zgodovine. Tam mejnike določajo industrijske revolucije: agrarna, industrijska ter informacijska (Turk, 2001).

## **B.2 Geometrijsko modeliranje**

Vse do 15. stoletja sta bila načrtovanje in gradnja zahtevnejših zgradb v domeni ene osebe – stavbenika. Šele s široko rabo papirja in tiska je bilo omogočeno dokumentiranje načrtovanja ter njegova distribucija. Pričetke geometrijskega modeliranja predstavljajo enostavne ilustracije arhitekture načrtovanih zgradb. Z razvojem opisne geometrije (Gaspard Monge – Opisna Geometrija (1801)) ter standardizacijo posameznih elementov zapisa so se pojavili precej bolj nazorni in verodostojni zapisi, ki poleg arhitekture vključujejo še preostale aspekte načrtovanja (statika, instalacije ipd.).

Danes kot rezultat načrtovanja razumemo v dokumentih zapisane verodostojne podatke/informacije o zgradbi (na papirju ali v digitalni obliki). Dokumenti, vezani na AEC/FM sektor, večinoma vsebujejo grafične podatke in jih zato imenujemo tehnične risbe ali pogovorno uveljavljeno kar načrti.

Ne glede na vrsto geometrijskega modeliranja (arhitekturni načrt, armaturni načrt, načrt strojnih instalacij) so elementi zgradbe vedno predstavljeni le s točkami, črtami, loki, ipd. Pravilna interpretacija rezultatov geometrijskega modeliranja je zato očitno v domeni človeka in je ni moč v celoti avtomatizirati.

Rezultati raziskav o rabi informacijskih tehnologij v gradbeništvu ((Rivald, 2000), (Liebich, Wix, 2002), (Samuelson, 2002), (Pazlar, Turk, 2004), (Goh, 2005), (Samuelson, 2008)) potrjujejo, da načrtovanje v AEC/FM praksi še vedno v veliki meri temelji na (dvodimenzionalnem) geometrijskem modeliranju. Zaradi univerzalnosti najpogosteje uporabljenih orodij za računalniško podprto načrtovanje (CAD – Computer Aided Design) uporaba le-teh ni nujno vezana zgolj na AEC/FM sektor, temveč so široko uporabljeni tudi v ostalih industrijskih panogah.

## **B.3 CAD sistemi**

Prvi prototipi CAD sistemov so se pojavili v začetku šestdesetih let 20. stoletja, teoretične osnove pa so se raziskovale že nekaj desetletij prej. Vodilna vloga v raziskavah je pripadala

avtomobilski (Renault, Citroen, Ford, General Motors) in letalski industriji (Boeing) ter z njimi povezanimi raziskovalnimi institucijami (MIT – Massachusetts Institute of Technology). Prav iz slednje izhaja sistem SKETCHPAD, ki ga je v letu 1963 predstavil Ivan Sutherland. Revolucionarnost omenjenega sistema predstavlja nov način grafične interakcije uporabnika s sistemom: načrtovanje poteka s svetlobnim peresom na sedemcolskem CRT-monitorju. Takšna arhitektura pravzaprav predstavlja temelj grafičnega uporabniškega vmesnika (GUI – Graphical User Interface), ki danes predstavlja nepogrešljiv del vsakega CAD sistema. Prve praktične implementacije sistemov, razvitih v začetku sedemdesetih let, je bilo zaradi visokih cen opreme in razvoja zopet moč najti le v avtomobilski in letalski industriji (DAC-1, Bell GRAPHIC 1, UNISURF). Nadaljnje desetletje razvoja programske in strojne opreme je na trg plasiralo nove tridimenzionalne telesne modelirnice (Unisolid, CATIA, Romulus). Koncepti dvodimenzionalnega načrtovanja ter pripadajoče praktične implementacije so s tem postali dostopnejši širšemu krogu uporabnikom. Revolucionarno rešitev je leta 1982 predstavil John Walker, ustanovitelj podjetja Autodesk. Njihov izdelek AutoCAD (sicer temelječ na programu MicroCAD) je deloval na osebni računalniku in je bil v primerjavi s konkurenti tako rekoč zastoj: stal je manj kot 1000 dolarjev. Navedeni karakteristiki sta poglavito prispevali k njegovi uspešnosti na trgu in še danes sicer povsem spremenjen AutoCAD predstavlja enega najbolj uporabljenih orodij za načrtovanje.

Vzporedno z razvojem 3D modelirnikov so se pojavile ideje in prototipi, pri katerih načrtovanje temelji na semantičnih (parametričnih) modelih.

#### **B.4 Geometrijsko modeliranje in CAD sistemi**

Danes je na trgu na voljo množica splošnih in ozko specializiranih orodij za načrtovanje geometrije produktov. Orodja se med seboj razlikujejo po predstavitvenem prostoru (2D, 3D) ter po tipu objektov, ki jih je moč modelirati (žični, površinski ali prostorninski model). Relacije med sodobnimi CAD sistemi ter geometrijskim načrtovanjem lahko še podrobneje definiramo (Liebich, Wix, 2002):

- CAD sistemi predstavljajo geometrijsko orientirano načrtovalsko okolje. Celoten načrt zgradbe je običajno sestavljen iz več risb, ki predstavljajo načrtovano zgradbo v posameznem pogledu (nadstropja, prerezi, fasada). Znotraj CAD sistema naj bi se vedno načrtovalo v »naravnem« merilu, vendar se stopnja natančnosti opisa



posameznih konstrukcijskih sklopov lahko bistveno razlikuje (npr. detajl fasade bo izrisan natančneje kot ovoj celotne zgradbe);

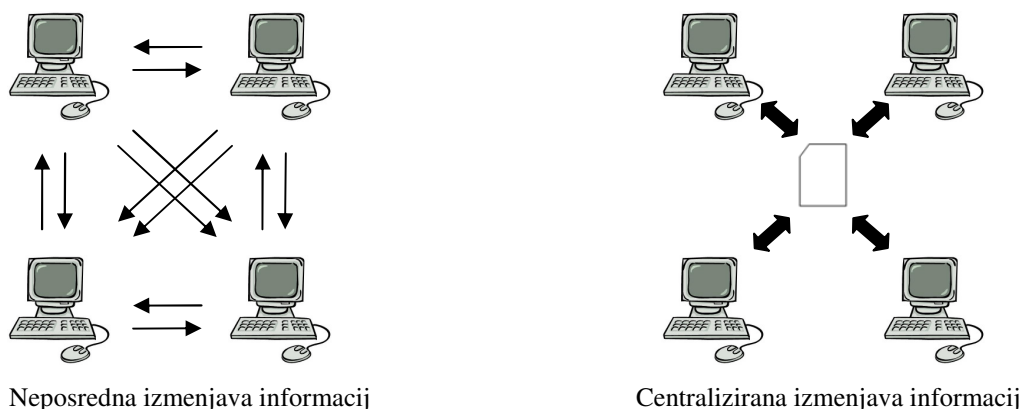
- fizični gradniki zgradbe so na načrtih predstavljeni kot geometrijski elementi (in ne kot primitivi zgradb), katerim se lahko z uporabo risalnih ravnin, blokov, zunanjih referenc in atributov pripiše dodatna sporočilnost;
- termin tehnična risba oz. načrt v kontekstu CAD praviloma predstavlja končni rezultat 2D načrtovanja. Kljub sodobnejšim načinom opisa produktov so v praksi klasični 2D načrti (arhitektura, instalacije ipd.) še vedno najbolj pogosti.

Pri AEC-FM orodjih, ki so namenjeni načrtovanju arhitekture, je običajno predstavitev geometrije zgradbe bistvena funkcionalnost programske opreme, medtem ko pri preostalih orodjih (npr. programi za statično analizo) predstavljajo geometrijski modelirniki le osnovo za domensko specifično delo.

Vsak CAD sistem ima glede na izbrano abstrakcijo geometrije svoj lasten zapis primitivov. Poleg že predstavljene nujne človeške interpretacije rezultatov geometrijskega modeliranja predstavlja nekompatibilnost zapisa poglavitno oviro učinkovitega načrtovanja. Zato ne preseneča dejstvo, da so se prvi poskusi integracije CAD sistemov oziroma standardizacije strukturirane forme vhodno-izhodnih podatkov pojavili kar vzporedno z razvojem CAD sistemov.

Glede na širok spekter rabe CAD programov ter veliko število ponudnikov opreme je pri načrtovanju produktov nemogoče pričakovati rabo enotne (oz. kompatibilne) programske opreme. Nasprotno pa težnjo po integraciji predstavljajo globalni trg, skupno nastopanje na trgu, odvisnost od dobaviteljev ter uporaba večnamenskih CAD orodij.

Opredelimo lahko dva načina izmenjave geometrije med različnimi CAD orodji: neposredna izmenjava (vsak komunicira z vsakim) ter centralizirana (nevtralni format za izmenjavo). Kljub očitnim prednostim drugega načina izmenjave je v praksi najbolj pogost pojav decentraliziranih sistemov, ki predstavljajo kombinacijo neposredne ter centralizirane izmenjave (v odvisnosti od vrste in pomembnosti nabora podatkov za izmenjavo, trenutne stopnje razvoja IT ter interesa uporabnikov) (Goldstein et al., 1998). Predstavljeni načini izmenjave so relevantni tudi za izmenjavo vseh ostalih (negeometrijskih) podatkov o produktu.



Slika 42: Neposredna in centralizirana izmenjava informacij

Figure 42: Direct and centralized information exchange

## B.5 Centralizirana izmenjava – podatkovni nivo

Prvi poskusi izmenjave informacij so bili omejeni na geometrijske primitive (Goldstein et al., 1998). Podobno kot pri razvoju CAD sistemov so se prve specifikacije izmenjave pojavile v letalski industriji. Leta 1977 je evropsko združenje letalske industrije AECMA predstavilo format za izmenjavo geometrije površin. Predstavljeni format zapisa je bil nekajkrat praktično uporabljen, vendar pa se je zaradi neustreznega koncepta ter nezmožnosti opisa kompleksnih ukrivljenih površin sčasoma prenehal uporabljati.

### B.5.1 IGES

Prva različica specifikacije IGES (Initial Graphics Exchange Specification – začetna specifikacija za izmenjavo grafike) je bila objavljena leta 1980 kot rezultat sodelovanja med Ameriškim nacionalnim inštitutom za standardizacijo (ANSI), ministrstvom za obrambo ter industrijo (Boeing, General Electric, Xerox, ComputerVision, Aplicon). Predstavljena specifikacija, ki velja za prvi ameriški standard za CAD izmenjavo podatkov poleg strukture zapisa predpisuje tudi jezik za predstavitev geometrije, topologije ter negeometrijskih informacij o produktu. Slednje predstavlja že sama oznaka: I (interim – vmesni), G (graphics – grafika in ne le geometrija), E (exchange – izmenjava) ter S (specification – specifikacija). IGES specifikacija definira produkt kot nabor entitet, zapisanih v 80-znakovnem ASCII vrstičnem zapisu (luknjane kartice). Vrstice so razvrščene v odseke:

- S (Start): vrstice vsebujejo človeku berljiv prolog k vsebini. Vsaj ena, četudi prazna vrstica je obvezna.
- G (Global): splošni podatki z opisom preprocesorja ter podatki, potrebnimi za postprocesiranje (npr: ime datoteke, različica IGES zapisa, datum in ura generacije zapisa, ime avtorja ter organizacije ipd.).
- D (Directory Entry): opis entitet (posamezni entiteti sta namenjeni dve vrstici). Entitete so razdeljene v pet skupin (krivulje in površine, konstruktivna volumska predstavitev, robna volumska predstavitev, notacije – komentarji ter strukturne entitete). Namesto imen entitete označujejo številke (npr. oznaka 110 predstavlja daljico, 116 točko, 100 krožni lok). Vsako entiteto sicer lahko opisujemo v svojem tridimenzionalnem evklidskem prostoru, vendar je potrebno dodatno podati ustrezno transformacijo v modelni prostor produkta. Vsaka entiteta, pripadajoča polja (atributi) ter tudi parametrični podatki so v specifikaciji standarda (IGES, 2006) natančno opredeljeni. V razdelku D se za posamezno entiteto podajata indeks pripadajoče vrstice v razdelku P ter opisni atributi entitete (transformacijska matrika, risalna ravnina ipd.).
- P (Parameter Data): v tem razdelku se nahajajo specifični podatki o entitetah, navedenih v sekciji D (npr. za daljico se podaja začetno in končno točko, za lok odmik od ravnine XY, center loka, začetno ter končno točko ipd.).
- T (Terminate): zaključek datoteke. Odsek sestavlja deset polj, od katerih pa jih je le pet v rabi. V njih so zapisani podatki o dolžini zapisov preostalih sekcij (S, G, D in P).

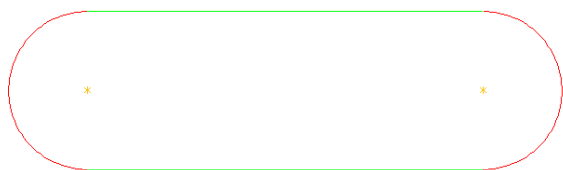
Oznaka sekcije se nahaja v 73. stolpcu. Znotraj sekcije so posamezne vrstice še dodatno inkrementno oštevilčene (desno poravnane številke na koncu vrstice).

Z IGES specifikacijo je možno izmenjati naslednje primitive: žične modele, površine ter tudi telesa. Entitete so v splošnem razdeljene na geometrijske (točke, krivulje, površine, telesa) ter na negeometrijske, s katerimi geometrijo dodatno utemeljimo (kote, opombe, tekst ipd.).

Specifikacija je bila takoj po objavi predana standardizacijskem telesu (ANSI Y14.26 komisiji). Po dveh revizijah je specifikacija postala ANSI standard Y14.26M-1981. Standard je bil prevzet tudi v nekaterih drugih državah (Japonska, Avstralija in Velika Britanija).

Specifikacija je dosegla precejšen uspeh predvsem v ZDA, saj je posamezne različice implementiralo precejšnje število pomembnejših razvijalcev programske opreme.

```
1H, , 1H, , 4H$LOT, 37H$1$DUA2: [IGESLIB.BDRAFT.B2I] SLOT.IGS; , S 1
17HBravo3 BravoDRAFT, 31HBravo3->IGES V3.002 (02-Oct-87), 32, 38, 6, 38, 15, G 1
4H$LOT, 1., 1, 4HINCH, 8, 0.08, 13H871006.192927, 1.E-06, 6., G 2
31HD. A. Harrod, Tel. 313/995-6333, 24HAPPLICON - Ann Arbor, MI, 4, 0; G 3
116 1 0 1 0 0 0 0 0 1D 4
116 1 5 1 0 0 0 0 0 OD 1
116 2 0 1 0 0 0 0 0 1D 2
116 1 5 1 0 0 0 0 0 OD 3
100 3 0 1 0 0 0 0 0 1D 4
100 1 2 1 0 0 0 0 0 OD 5
100 4 0 1 0 0 0 0 0 1D 6
100 1 2 1 0 0 0 0 0 OD 7
110 5 0 1 0 0 0 0 0 1D 8
110 1 3 1 0 0 0 0 0 OD 9
110 6 0 1 0 0 0 0 0 1D 10
110 1 3 1 0 0 0 0 0 OD 11
116, 0., 0., 0., 0., 0, 0, 0; 1P 1
116, 5., 0., 0., 0., 0, 0, 0; 3P 2
100, 0., 0., 0., 0., 1., 0., -1., 0, 0; 5P 3
100, 0., 5., 0., 5., -1., 5., 1., 0, 0; 7P 4
110, 0., -1., 0., 5., -1., 0., 0, 0; 9P 5
110, 0., 1., 0., 5., 1., 0., 0, 0; 11P 6
S 1G 4D 12P 6 T 1
```



Privzeta slika 17: IGES tekstovna datoteka – opis dveh točk, daljic in krožnih lokov (IGES5, 2007)

Adopted figure 17: IGES text file – description of two points, lines and arches (IGES5, 2007)

Zadnja različica standarda IGES (5.3) je bila objavljena leta 1996 in se v ZDA še vedno uporablja, predvsem za izmenjavo rezultatov 2D načrtovanja. V pripravi naj bi bila tudi različica 6.0, vendar pa je zanimanje za nadaljnji razvoj tega standarda po letu 1996 upadlo. Vse bolj se je namreč pričel uveljavljati mednarodni standard za izmenjavo podatkov ISO 10303, ki v tehničnem smislu deloma temelji na IGES specifikaciji.

### B.5.2 VDA-FS in SET

Podobno kot v letalski je bilo tudi v avtomobilski industriji moč zaslediti potrebo po izmenjavi geometrije ter s tem izboljšanjem procesa načrtovanja in uporabnosti CAD/CAM sistemov. Obstoječi format IGES je bil zaradi za avtomobilsko industrijo specifične

geometrije (prostorske krivulje, ukrivljene ploskve) potrebno dopolniti. Tako je bil v Nemčiji v letu 1982 predstavljen standard VDA-FS (Verbandes der Deutschen Automobilindustrie - Flachenschnittstelle).

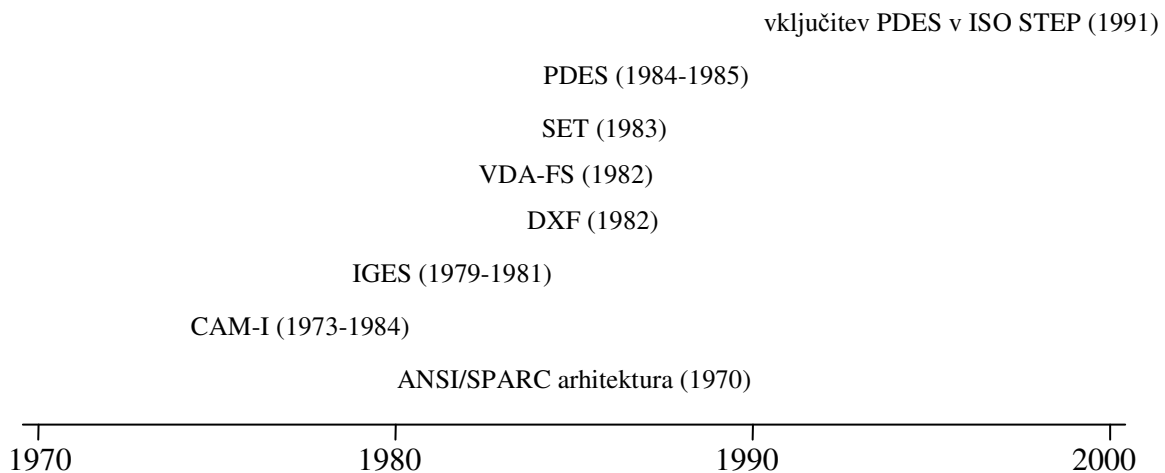
Neustreznost IGES-a za potrebe avtomobilske ter letalske industrije so na nacionalnem nivoju podobno kot Nemci reševali tudi Francozi: leta 1983 je bil predstavljen standard SET (Standard d'Echange et de Transfert).

Za razvoj standarda SET je skrbelo združenje GOSET, v katerem so sodelovali predstavniki vladnih organizacij in industrije. Prva različica specifikacije (SET 1.1) je obsegala (Goldstein, 1998):

- natančno specifikacijo entitet za opis geometrij v strojništvu,
- dodatne informacije o strukturah podatkov ter o uporabljenih konceptih,
- pravila in priporočila za zagotavljanje koherence pri nadaljnjem razvoju specifikacije.

Predstavniki obeh združenj so sodelovali tudi pri pripravi standarda ISO 10303.

### B.5.3 CAM-I



Slika 43: Pregled geometrijskega modeliranja

Figure 43: Geometric modelling – historical overview

Pomembnejši prispevek k formalnemu opisu geometrije predstavlja specifikacija, izdelana v združenju Computer-Aided Manufacturing – International (CAM-I). V združenje so bili vključeni predvsem industrijski partnerji s področja elektrotehnike in računalništva. Poskusni mejni model (XBF – Experimental Boundary File) obsega matematični zapis geometrije ter topologije s poudarkom na predstavitvi geometrije teles z mejnimi ploskvami (BREP – Boundary Representation). Možnosti opisa, ki jih je ponujala CAM-I specifikacija so bile v letu njene predstavitve (1982) precej nad zmožnostmi, ki jih je ponujala takratna strojna in programska oprema (Goldstein, 1998).

Poudariti je potrebno, da CAM-I specifikacija ne vsebuje mehanizmov za izmenjavo, temveč le temeljni opis podatkov, ki jih je moč izmenjevati.

#### **B.5.4 DXF**

Specifikacija izvornih zapisov lastniških aplikacij praviloma ni javno objavljena, vendar pa zahteve končnih uporabnikov silijo tudi razvijalce programske opreme v razvoj alternativnih rešitev, to je zapisov, katerih specifikacija je prosto dostopna.

Podjetje Autodesk je tako že v letu 1982 predstavilo format DXF (Drawing Interchange Format oz. Drawing eXchange Format), in sicer kot alternativo izvorni binarni DWG specifikaciji. Podobno kot pri ostalih interoperabilnosti namenjenim zapisih gre pri DXF specifikaciji za tekstovni zapis podatkov v ASCII datoteko (na voljo je sicer tudi binarni zapis). Vsaki različici DWG zapisa pripada svoja različica DXF specifikacije – osnovna zgradba zapisa je sicer identična, razlike nastopajo na nivoju entitet oz. elementov.

V DXF formatu se objekti razlikujejo od entitet – elementi geometrije so predstavljeni kot entitete, medtem ko objekti nimajo grafične predstavitve. Tekstovno datoteko predstavlja sedem razdelkov (preglednica 14). Posamezen zapis v določeni sekciji obsega dve vrstici: v prvi se nahaja koda gruče (0–9 niz znakov, 999 komentar in drugo) v drugi vrstici pa številčne vrednosti.

V splošnem je uporabnost DXF specifikacije precej vprašljiva, saj le-ta ne obsega (dovolj podrobnega) opisa primitivov, ki jih je moč zapisati v formatu DWG (npr. ACIS volumskih teles, dinamičnih blokov in še nekaterih kompleksnejših gradnikov). Omeniti je potrebno še težave z velikostjo datotek ter z numerično natančnostjo. V DXF formatu je uporabljena natančnost šestih decimalnih mest, v DWG formatu pa štirinajst (Opendesign, 2006).

## Preglednica 14: Struktura DXF datoteke

Table 14: DXF file structure

glava	splošne spremenljivke in njihove vrednosti (različica DXF specifikacije, datum in ura, natančnost, enote, velikost risbe, vidnost, tolerance, nastavitve kotiranja ipd.)
razredi in tabele	priprava razredov, katerih instance se bodo pojavile v nadaljevanju (npr. konstante, koordinatni sistemi, risalne ravnine, slogi – tekst ipd.)
bloki	skupine entitet, ki tvorijo zaključeno enoto. Bloki obstajajo tudi, če s strani končnega uporabnika niso bili definirani (npr. šrafure predstavljajo t. i. anonimni blok)
entitete	opis geometrijskih elementov modela (točke, daljice, ploskve ipd.)
objekti	podatki o negrafičnih objektih (potrebna AutoLISP oz. ObjectARX podpora)
predogled	slika predogleda DXF datoteke
konec	zaključek datoteke

**B.5.5 OpenDWG in OpenDNG**

Nezadovoljiva dokumentacija ter opisane težave pri vsakdanji rabi DXF zapisa za izmenjavo podatkov je botrovala nastanku združenja Open Design (sprva je združenje imenovalo OpenDWG), ki ponuja alternativo zapisu, uporabljenem v Autodeskovih programih. Poleg objavljene specifikacije združenje ponuja tudi programske knjižnice za pomoč pri pisanju/branju OpenDWG datotek. Podobne rešitve in knjižnice združenje ponuja tudi za DNG format (Bentley programska oprema), vendar na podlagi poznane specifikacije (navedeno podjetje je namreč član združenja).

Že pred ustanovitvijo združenja so nekatera podjetja skušala izdelati programske vmesnike, sposobne branja/zapisovanja v sicer neobjavljeno DWG specifikacijo. Eno izmed uspešnejših podjetij – Marcomp – je podobno kot ostali raziskovalci uporabilo princip črne škatle: z opazovanjem spreminjana binarnih AutoCAD-ovih datotek pri manjših spremembah testnih risb so uspeli razvozlati zapis in izdelati vmesnike za branje/pisanje DWG zapisa, in sicer za različice AutoCAD 2.5 do 14. Njihov program AUTODIRECT je postal standard za delo z DWG datotekami izven programskih orodij podjetja Autodesk.

Podjetje Visio je leta 1998 prevzelo Marcomp ter nato ustanovilo združenje OpenDWG, ki nadaljuje (oz. je nadaljevalo) z izdelavo knjižnic oz. vmesnikov do različice 2004 (OpenDesign, 2006).

Ne glede na izvor posameznih specifikacij (mednarodna združenja, alternativa) se le-te niso najboljše uveljavile v praksi. Kljub predstavljenim pomanjkljivostim izvornih zapisov pa raziskave trenutnega stanja v gradbeni informatiki ((Pazlar, Turk, 2004), (OpenDesign, 2006)) nakazujejo, da le-ti še vedno predstavljajo najpogostejši način hranjenja rezultatov načrtovanja v CAD programih. Končni uporabniki niso dovolj osveščeni o prednostih predstavljenih specifikacij oz. le-te niso zadostne, da bi prepričale končne uporabnike k uporabi navedenih zapisov.

## **B.6 Informacijsko modeliranje**

Vse do sedaj predstavljene specifikacije so namenjene predvsem modeliranju geometrije. Z dodatnimi entitetami, atributi in relacijami je mogoče opisati tudi »negeometrijske« gradnike. Zato je ostro ločnico med geometrijskim ter informacijskim modeliranjem težko določiti. Najpogosteje se za pričetke informacijskega modeliranja označuje strukturiran pristop vhodno-izhodnih podatkov iz sedemdesetih let prejšnjega stoletja (Turk, 1992). Nekateri koncepti informacijskega modeliranja so bili pred pojavom pravih informacijskih modelov zgradb prisotni in vezani zgolj na CAD sisteme. Tipičen primer predstavlja Archicad (Graphisoft), prvi CAD program, v katerem je bilo moč načrtovati dvo- in hkrati tridimenzionalno. Program, katerega začetki segajo v leto 1982, je bil v osnovi sicer namenjen modeliranju geometrije, vendar pa za razliko od ostalih orodij (npr. Autocad) modeliranje temelji na semantičnih parametričnih objektih. Graphisoft označuje takšen model z zaščitenim imenom »virtualna zgradba«.

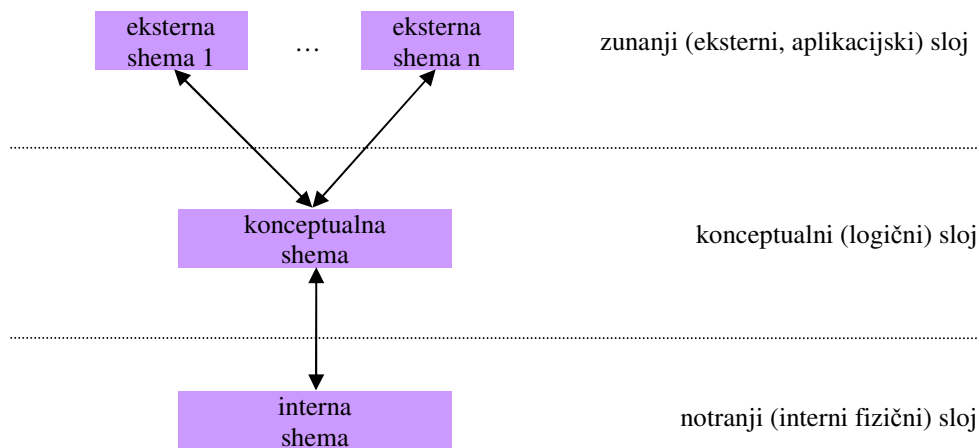
Pomemben prispevek k informacijskemu modeliranju poleg že podrobneje predstavljenih (mednarodnih) iniciativ za izmenjavo podatkov o geometriji predstavljajo tudi nekateri danes večinoma nepoznani CAD sistemi: BDS/GDS, RUCAPS/SONATA (Velika Britanija), GLIDE in CAEADS (ZDA). Njihov prispevek v razvoju predstavlja inovativna in še danes uporabljena implementacija relacij med objekti in geometrijo: geometrija je lastnost objekta in ne obratno.



RUCAPS je na primer omogočal t. i. 2.5D načrtovanje. Primitivi zgradbe so pri takšnem načrtovanju v prostor umeščeni tridimenzionalno, vendar pa je vsak izmed njih modeliran z 2D pogledi. Pri RUCAPS-u velja omeniti še pojav enega izmed prvih CAD večuporabniških vmesnikov, ki omogočajo sočasno modeliranje.

### B.6.1 ANSI/SPARC arhitektura

Podobno kot v Evropi so se tudi v ZDA ukvarjali s standardizacijo jezika za izmenjavo informacij o produktu. Leta 1975 je bila pod okriljem Ameriškega nacionalnega inštituta za standardizacijo (ANSI-SPARC – American National Standards Institute / Standards Planning and Requirements Committee) predstavljena t. i. troslojna arhitektura sistema, ki ločuje opis podatkov in njihovo rabo oz. predstavitev (slika 44). Poskusi, da bi vse podatke za integrirani CAD sistem opisali le na podatkovnem nivoju, so se namreč izkazali za neuresničljive (Reed, 1998).



Slika 44: Troslojna ANSI-SPARC arhitektura

Figure 44: ANSI-SPARC three layer architecture

ANSI-SPARC arhitektura predstavlja temelje večini sodobnih konceptualnih modelov (Turk, 1992). Troslojno arhitekturo, izvirajočo in tudi bolj poznano iz teorije podatkovnih baz sestavljajo:

- zunanji (eksterni, aplikacijski) sloj opredeljuje različne poglede uporabnikov na podatke oz. model. Pri tem velja, da je v splošnem za vsakega uporabnika na voljo prilagojen pogled, ki ga opisuje pripadajoča zunanja shema. Posledično za različne uporabnike obstajajo različne sheme za predstavitev istih podatkov;
- konceptualni (logični) sloj obsega že predstavljeni konceptualni model. Predstavljena plast ne vsebuje nobenih podatkovnih struktur ter tudi ne načina dostopa do podatkov;
- notranji (interni, fizični) sloj opisuje predstavitev, manipulacijo in shranjevanje podatkov (podatkovne strukture, datoteke, indeksi ipd.).

### **B.6.2 ICAM (1973–1984)**

Sredi sedemdesetih let so se z uveljavitvijo računalniške tehnologije pojavile zamisli, kako bi s sistemsko implementacijo le-te povečali produktivnost proizvodnje. Ameriška vojska je s programom ICAM (Integrated Computer Aided Manufacturing) kot ključno zahtevo identificirala izboljšavo načinov komuniciranja med posameznimi sistemi. Nekompatibilnost je namreč predstavljala poglobitno oviro vseh do tedaj uveljavljenih metod (sekvenčne, hierarhične, mrežne) shranjevanja podatkov.

Rezultat projekta predstavlja zbirka različnih tehnik modeliranja, poznanih pod imenom IDEF (ICAM Definition):

- IDEF0 (Integration DEFinition Language 0) jezik za opis »funkcijskega« modela. Funkcijski model predstavlja strukturirano predstavitev funkcij aktivnosti in procesov znotraj UoD. IDEF0 obsega grafični jezik za modeliranje ter tudi opis metodologije za razvoj modelov. Rezultat modeliranja z IDEF0 predstavlja hierarhični nabor diagramov, teksta ter pripadajočih referenc. Poglavitne komponente modeliranja predstavljajo funkcije (grafično predstavljene s pravokotniki) ter podatki oz. objekti, ki povezujejo funkcije (grafično predstavljeni s puščicami).
- IDEF1 za opis »informacijskega« modela. Informacijski model v tem kontekstu predstavlja strukturo in semantiko sistema oz. področja, ki ga modeliramo.
- IDEF2 za opis »dinamičnega« modela. Dinamični model obsega opis karakteristik UoD v odvisnosti od časovne komponente.

Do vključno leta 1995 je bilo razvitih štirinajst IDEF metod: IDEF3 – zajem opisov procesov, IDEF4 – objektno orientirano načrtovanje, IDEF5 – ontologije, IDEF6 – temeljni principi načrtovanja, IDEF7 – informacijski sistemi, IDEF8 – modeliranje uporabniških vmesnikov, IDEF9 – načrtovanje informacijskih sistemov, IDEF10 – arhitektura implementacij, IDEF11 – modeliranje primitivov, IDEF12 – modeliranje sestavov, IDEF13 – preslikave drevesnih shem, IDEF14 – načrtovanje mrež.

### **B.6.3 PDDI**

Namen PDDI projekta (Product Definition Data Interface – vmesnik za izmenjavo podatkov o produktu) (1982–1987) je bil razvoj mehanizma za neposredno deljenje oz. izmenjavo podatkov o produktu med računalniškimi aplikacijami brez posredovanja človeka.

Inovativnost projekta, vodenega s strani ameriške vojske, se kaže v ločenem opisu podatkov in mehanizmov izmenjave podatkov. V okviru PDDI projekta je bil razvit nov jezik za modeliranje informacij, ki je pomembneje vplival na nastanek in na razvoj danes najbolj uporabljanega jezika za produktno modeliranje, to je jezika EXPRESS.

Poleg izdelave vmesnika za izmenjavo podatkov o produktu je projekt PDDI kritično ocenil IGES zapis ter njegove implementacije (Goldstein, 1998):

- različni načini opisa istih informacij (interpretacija odvisna od vmesnika);
- čas, potreben za procesiranje oz. velikost datoteke, je bil za takratno strojno opremo nesprejemljivo (pre)dolg;
- izguba informacij med izmenjavo (še posebno med različno zmogljivimi CAD sistemi);
- preveč posplošena in zato neprimerna arhitektura;
- povezljivost vnaprej ni zagotovljena (starejše različice IGES zapisov ni možno brati z novejšo programsko opremo, saj pripadajoči vmesniki niso bili razviti);
- IGES predstavlja le način izmenjave načrtov in ne izmenjave kompleksnejših informacij o produktu;
- delna implementacija IGES specifikacije onemogoča izmenjavo podatkov (razen v primerih predhodnega dogovora);

- mehanizmi za testiranje vmesnikov ter reševanje napak niso bili definirani.

Kljub raziskovalni naravnosti PDDI projekt predstavlja nepogrešljiv prispevek pri razumevanju mehanizmov izmenjave podatkov.

#### **B.6.4 PDES**

Že tri leta po uradni izdaji prve različice standarda IGES se je pokazalo, da je specifikacija zadovoljiva le za obstoječe CAD programe, ne pa tudi za prihajajoče CAD sisteme. Zato se je že v letu 1984 pristopilo k razvoju specifikacije za izmenjavo podatkov o produktih (PDES – Product Data Exchange Specification – specifikacija za izmenjavo podatkov o produktu), ki bi omogočila nedvoumne računalniško razpoznavne definicije fizičnih funkcionalnih ter programskih značilnosti vsake enote dela produkta skozi njegovo življenjsko dobo (Turk, 1992). PDES (1984–1985) bi lahko označili kot novo, bolj splošno generacijo IGES specifikacije, ki pa se v praksi ni nikoli uveljavila. Podobno kot pri projektu PDDI je bil poglobitni cilj predstavljene specifikacije predvsem spoznavanje standardizacije kompleksnih produktov ter raziskovanje možnosti avtomatizacije (Goldstein, 1998).

PDES iniciativa se je v letu 1985 pridružila evropski pobudi STEP (ISO 10303).

#### **B.6.5 HPS**

Namesto uvajanja novih specifikacij pobuda HPS (Harmonization of Product Data – harmonizacija podatkov o produktu) s konca osemdesetih let temelji na dialogu med posameznimi standardizacijskimi iniciativami, ki so bile podobno kot XBF specifikacija razvite v domenah elektrotehnike oz. elektronike. V letu 1989 je vodenje HPS prevzel NIST. Pod novim vodstvom so bili ustanovljeni trije koncili: poslovne potrebe in načrtovanje, razvoj standardov ter koordinacija in orodja ter tehnologije. Poglobitni dosežek iniciative predstavlja predlagana metodologija za harmonizacijo ANSI standardov ter objava prve različice »koordinacijskega informacijskega modela ANSI/HPS 100«.

#### **B.6.6 STEP – ISO 10303**

STEP (STandard for the Exchange of Product model data – standard za izmenjavo informacij o produktu) predstavlja trenutno najbolj uveljavljen standard na področju splošnega produktnega modeliranja. Standard vsebuje mehanizme za nevtralni opis produkta skozi

njegov življenjski cikel, neodvisno od sistema. S tem je omogočena ne le nevtralna datotečna izmenjava, temveč je zagotovljena tudi osnova za delo s produktnimi podatkovnimi bazami ter arhiviranje.

STEP se uporablja za izmenjavo podatkov med CAD, CAM, CAE, PDM/EDM in drugimi CAx sistemi v strojništvu, gradbeništvu, elektrotehniki ter na številnih drugih področjih. Standard je razdeljen na posamezne dele (»Parts«) z oznakami 1–1xxx. Domensko vezani deli standarda so poimenovani aplikacijski protokoli (»Application protocols«).

Standard STEP je v pregledu razvoja informacijskih modelov zgradb večkrat omenjen, saj predstavlja podlago za več v nadaljevanju predstavljenih specifikacij (med drugim na njem temelji tudi IFC specifikacija). Podrobnejši pregled standarda ISO 10303 je podan v tretjem poglavju.

## **B.7 Razvoj informacijskih modelov zgradb**

V podpoglavju »informacijsko modeliranje« so bile predstavljene pomembnejše iniciative v generaliziranem produktnem modeliranju. Istočasno so se na AEC-FM področju razvijale ožje usmerjene specifikacije, ki so skušale z različno stopnjo natančnosti opisati izdelke industrije, ki oblikuje grajeno okolje.

Obstaja več pogledov na razvoj informacijskih modelov zgradb. Le-te lahko v osnovi razdelimo na ogrodne (izbrani UoD opisujejo v celoti) ter aspektne modele (UoD se opisuje z delnimi do izbranega nivoja kompatibilnimi modeli) ((Eastman, 1999), (Björk, 1995)). Zadnji izmed navedenih avtorjev utemeljuje delitev ob kronološki razlagi informacijskih modelov zgradb:

- predhodne raziskave – definicija problema ter zgodnji prototipi modelov (1970–1985). Navedeno obdobje zaznamujejo preveč napredni prototipi, ki zaradi premalo zmogljive strojne opreme niso bili nikoli implementirani v praksi;
- sistemi za računalniško načrtovanje ter ekspertni sistemi (1980–1984);
- objektno orientirani in konceptualni modeli (1985–1990). V sredini osemdesetih let so strojna oprema, programski jeziki ter teorija podatkovnih baz zagotavljali minimalen potreben nivo za digitalni opis zgradb;
- slojeviti produktni podatkovni modeli (od 1990 dalje).

Kompleksnost sodobnih informacijskih modelov zgradb je takšna, da jih v celoti ni mogoče več opisati v enem koraku. Zato je arhitektura večslojna in razširljiva, modele pa se ponavadi deli v manjše še obvladljive kose oz. poddomene. Takšne podmodele označujejo različni kvalifikatorji: lokalni produktni modeli (»local product models«), aktualni modeli (»topical models«), delni modeli (»partial models«), ogledni modeli (»view type models«) ter aspektni modeli (»aspect models«). Takšni modeli ne morejo biti definirani ločeno, temveč morajo biti kompatibilni vsaj v tistih podatkovnih strukturah, ki so skupne posameznim delnim modelom. Navedene podatkovne strukture običajno označuje termin jedro (»kernel«, »core«). Podobno se lahko tudi preostale podatkovne strukture združuje v sloje. Poglavitno prednost večslojne arhitekture predstavlja možnost inkrementne definicije podatkovnih struktur ter s tem tvorbo relativno neodvisnih podmodelov.

Sorodna strategija z uporabo slojev predstavlja pravilo tudi pri tvorbi ogrodij celotnih modelov. Število slojev, ki jih vsebuje sodobni informacijski model, je praviloma karakteristika vsakega modela. Björk (1995) navaja, da je izbira števila slojev sicer poljubna, vendar pa naj bi bilo produkte industrije, ki oblikuje grajeno okolje, mogoče učinkovito razgraditi že s petimi sloji (konkretni primeri bodo podrobneje predstavljeni v nadaljevanju):

- **Jeziki za modeliranje informacij.** Z izbranim jezikom za modeliranje je potrebno opisati informacijske strukture, kot so objekti, atributi, razmerja, generalizacije – specializacije, pravila, metode, sporočila ipd. V predstavljeno skupino uvrščamo relacijske, podatkovne, modele entiteta-razmerje ter splošne objektne modele.
- **Generični opis produkta.** Ta sloj praviloma vsebuje informacijske strukture, ki opisujejo primitive, ki jih je ustvaril človek: obliko in umestitev objektov, razmerja med objekti, podatke o različicah obravnavanega modela ter ponovno rabo ponavljajočih se komponent. Primer takšnega opisa predstavljajo model GARM, STEP (del, ki opisuje vire), objektno usmerjeni CAD sistemi (OOCAD).
- **Jedro modela** opisuje informacijske strukture, specifične za modeliranje zgradb, in sicer le tiste, ki se pojavljajo v več disciplinah in v več fazah življenjskega cikla zgradb. Med drugim mora jedro zadostiti vsaj naslednjim zahtevam:
  - zmožnost modeliranja poljubnih informacij v poljubni fazi življenjskega cikla zgradbe;

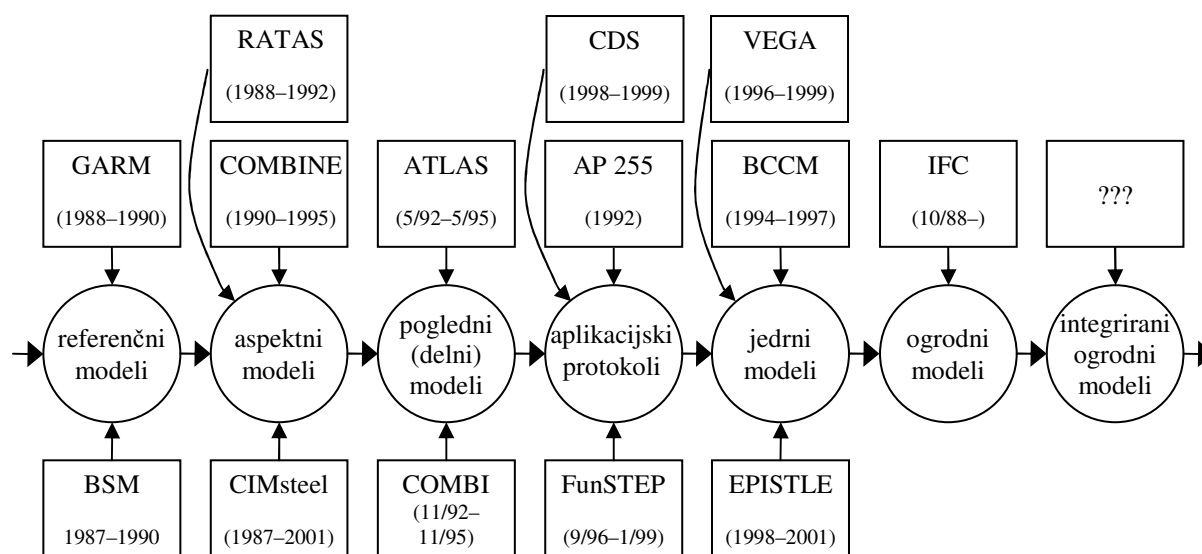
- uporabnost modela za vse udeležence v življenjskem ciklu zgradbe;
- natančna specifikacija opisa brez podvajanj;
- alternativni izpis podatkov v različnih formatih;
- možnost implementacije v širokem spektru programske opreme.

Modela BSM in RATAS predstavljata primera modelov, osredotočenih predvsem na opis jedra.

- **Aspektni modeli** praviloma vsebujejo informacije, ki opisujejo specifične poddiscipline oz. faze v življenjskem ciklu zgradb. Praktično ti modeli vsebujejo precej informacijskih struktur iz prvih treh slojev (z upoštevanjem dedovanja). Hipotetičen primer: osnovni opis armiranobetonske stene bo uvrščen v jedro modela, medtem ko bodo detajli armature predstavljeni v aspektnem modelu, specifičnem za gradbenike konstruktorje. Podobno velja tudi za požarne cone, instalacije ipd. Enake zahteve, kot so podane za jedro modela, veljajo tudi za aspektne modele. V predstavljeni kategoriji se uvrščajo Integrirani podatkovni model (IDM) projekta COMBI, ter modela COMBINE in RATAS.
- **Aplikacijski modeli.** Predstavljeni sloj vsebuje implicitno ali eksplicitno konceptualno shemo, implementirano v specifično računalniško aplikacijo in je očitno v domeni razvijalcev aplikacij. Sloj je naveden le ilustrativno, saj v nasprotju s predhodno navedenimi sloji zanj ni potrebno razvijati dobro dokumentiranih in večkratno uporabnih specifikacij. Razmerje med shemami aplikacijskih modelov ter višje ležečimi shemami je lahko neposredno ali posredno. Pri neposrednem so aplikacijske sheme izpeljane iz standardiziranega jedra in aspektnih modelov z upoštevanjem dedovanja ter z dodajanjem za posamezen aplikacijski model specifičnih struktur. Takšen pristop lahko bistveno pripeva k učinkovitosti izmenjave ter h kvaliteti vmesnika. Nasprotno pa posredno razmerje označuje prirejanje med obstoječim zapisom posamezne aplikacije ter aplikacijskim modelom, ki ga želimo implementirati. Očitno je takšen pristop najbolj pogost pri izdelavi vmesnikov za obstoječo programsko opremo.

Ne glede na število izbranih slojev se podrobnost opisa le-teh običajno precej razlikuje. Prvi poskusi standardizacije informacijskih modelov zgradb so bili omejeni na generične opise produktov, medtem ko večina najnovjših poskusov vsebuje tudi komponente aplikacijskega sloja.

V podrobnejši pregled informacijskega modeliranja zgradb je vključena večina pomembnejših mednarodno priznanih prispevkov z relevantnim vplivom na trenutno najbolj poznane in uporabljene modele v AEC-FM sektorju (slika 45). Razvrstitev predstavljenih modelov, delno povzeta (po Liebich, Wix, 2002) očitno ni strogo kronološka, temveč so modeli kategorizirani glede na svojo strukturo. V literaturi ((Björk, 1995), (Eastman, 1999)) je sicer moč najti tudi bolj grobe razvrstitve. Vse navedene razvrstitve med drugim temeljijo na relativno subjektivnih ocenah pomembnosti posameznih modelov pri evoluciji informacijskih modelov zgradb. Posledično se posamezne razvrstitve projektov oz. raziskovalnih modelov lahko precej razlikujejo.



Slika 45: Informacijski modeli v AEC-FM sektorju

Figure 45: Information models in AEC-FM industry



### B.7.1 Predhodne raziskave

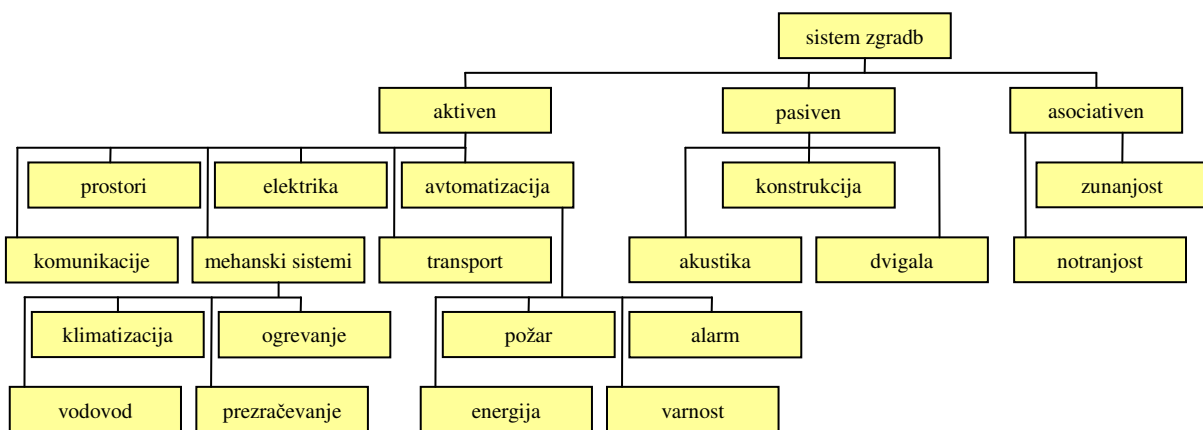
V predhodne raziskave produktnega modeliranja na področju gradbeništva se lahko uvrstijo spoznanja na področju podatkovnih baz, v predhodnem podpoglavju predstavljeni geometrijski modeli, CAD sistemi ter deloma že predstavljeni generalizirani produktni podatkovni modeli.

### B.7.2 Referenčni modeli

Poimenovanje referenčnih modelov izhaja iz namere, da bi le-ti opisali celotno področje gradbeništva v najširšem pomenu besede. Opis modelov je povsem posplošen, konkretne implementacije pa bi zahtevale dodatne raziskave in razvoj. Ključna predstavnik referenčnega modeliranja sta GARM in BSM.

#### B.7.2.1 BSM – Building System Model

BSM velja za prvi informacijski model, razvit posebej za gradbeno stroko, avtor James Turner pa ga je prvič predstavil leta 1988, in sicer v okviru že predstavljenega PDES projekta. Model temelji na navidez preprostem sistemskem pogledu na UoD: karkoli povezano s samo zgradbo predstavlja sistem oz. del sistema, ki je lahko aktiven (HVAC) ali pasiven (nosilna konstrukcija) ali asociativen (notranjost) (privzeta slika 18).



Privzeta slika 18: Taksonomija BSM (Liebich, Wix, 2002)

Adopted figure 18: BSM taxonomy (Liebich, Wix, 2002)

Nekateri viri (Liebich, Wix, 2002) označujejo predstavljeni model kot prvi poskus izdelave informacijskega modela v gradbeništvu z natančno opredelitvijo projektov, ki jih je moč modelirati (bivalni in industrijski objekti, ladje, prostorski habitati) ter projektnih faz (konceptualno načrtovanje, preliminarno načrtovanje, načrtovanje, načrtovanje detajlov, projektna dokumentacija, načrtovanje gradnje oz. terminski plani, gradnja, uporaba in vzdrževanje, ponovno načrtovanje oz. preprojektiranje ali odstranitev). Posamezen sistem enolično opredeljujeta lokacija ter modelirana zgradba (z lastnostmi, kot so tip zgradbe, primarna in sekundarna aktivnost ipd.). Sistem opredeljujejo systemske komponente (in lastnosti). Z razčlenitvijo se lahko nadaljuje, saj systemske komponente zopet predstavljajo svoj sistem s pripadajočimi komponentami.

Osnovni koncepti BSM predstavljajo komponente s poljubnim številom lastnosti. Večina objektov gre v življenjski dobi skozi tri faze:

- generična faza (atributi niso določeni),
- specifična faza (večina atributov je poznanih),
- navzočnost (umestitev specifičnega objekta je natančno določena).

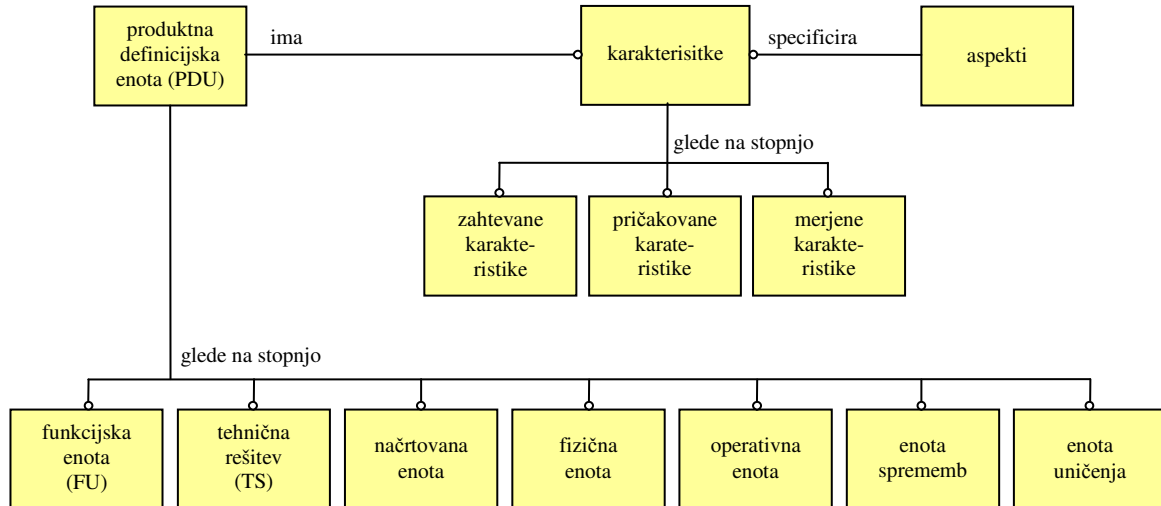
Posebnost modela predstavlja t. i. neposredna izmenjava, kjer se informacije izmenjujejo s pomočjo generičnih objektov ter atributov in ne z za industrijo specifičnimi objekti (npr. vrata imajo atribut mere, material, barvo ipd.). Navedena posebnost pri CAx izmenjavah predstavlja vir napačnih interpretacij – npr. kar v enem sistemu predstavlja dolžino, v drugem predstavlja širino (in obratno). Za odpravo takšnih nejasnosti bi bilo poleg modela potrebno dodatno definirati še podatkovni model z domensko specifičnimi objekti ter pripadajočimi atributi.

#### **B.7.2.2 GARM – General AEC Reference Model**

Specifikacija GARM je bila izdelana kot meta model, ki določa splošne principe modeliranja, ki jih je moč uporabiti pri izdelavi konkretnih modelnih specifikacij. Neposredna implementacija modela GARM posledično ni mogoča.

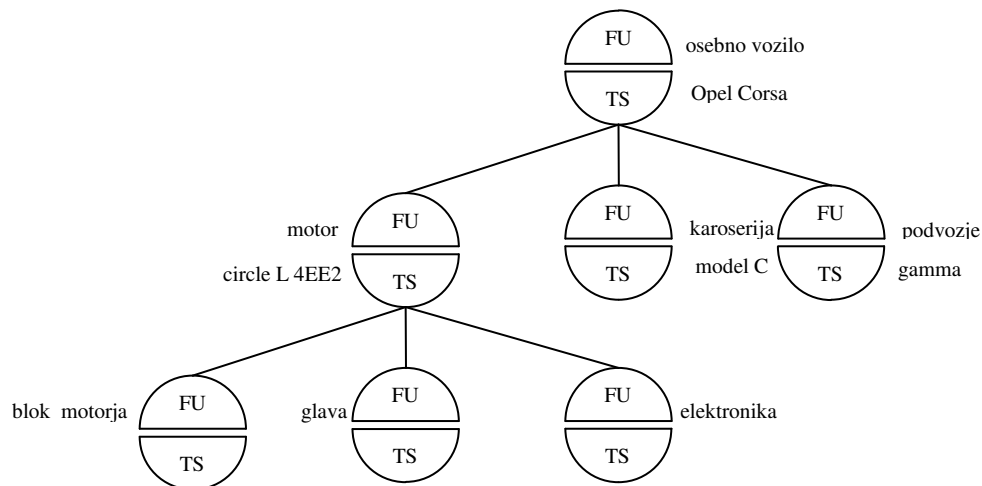
Izhodišče specifikacije predstavlja produktna definicijska enota (PDU – Product Definition Unit). Le-ta predstavlja vsakdanje gradbene objekte (stavbe, tovarne, mostove ipd. – torej zgradbe v najširšem pomenu besede) ali pa samo posamezne dele zgradb. Produktni definicijski enoti pripadajo karakteristike ter aspekti (oboje predstavlja lastnosti PDU). Pri

tem aspekti oz. vidiki označujejo splošne lastnosti (cena, toleranca ipd.), medtem ko karakteristike označujejo specifično vrednost določene lastnosti (privzeta slika 19).



Privzeta slika 19: PDU in podtipi modela GARM (Ghang, 2004)

Adopted figure 19: GARM PDU units and subtypes (Ghang, 2004)



Privzeta slika 20: GARM – hamburger diagram (Ekholm, 1994)

Adopted figure 20: GARM – hamburger diagram (Ekholm, 1994)

Jedro modela opisujeta termina funkcionalna enota (FU – Functional Unit) ter tehnična rešitev (TS – Technical Solution), ki omogočata razvoj modelov v smislu »deli in vladaj«: potrebe produktnih definicijskih enot izražajo funkcionalne enote, zadošča pa se jih s tehničnimi rešitvami. Najbolj priljubljen prikaz takšnih relacij predstavljajo »hamburger diagrami« (privzeta slika 20).

Omeniti je potrebno še pojav generične pojavnosti paradigme, s katero je olajšan razvoj posameznih komponent specifikacije, in sicer od splošnih idej do povsem konkretnih zapisov, ki jih je moč večkrat uporabiti. Konkretno predstavljena paradigma uveljavlja koncept entitet/instanc, ki so predstavljene in shranjene enkrat, vendar pa so preko referenc vedno dostopne.

Podrobnejša analiza GARM specifikacije (ugotavljanje njene primernosti za ISO standard) je sprožila kar nekaj vprašanj glede terminologije in konceptualne jasnosti modela. Ekholm (1994) je izpostavil kar nekaj pomankljivosti:

- nekatere produktne definicijske enote na nivoju produktnih tipov nimajo nadtipov;
- zamisel o generični specifični in pojavnosti razvrstitvi razredov ni zadosti obrazložena in zato lahko pri implementacijah privede do pojava razredov namesto konkretnih entitet;
- FU-TS sheme oz. »hamburger diagrami« (privzeta slika 20) so precej nejasni, saj skušajo na eni shemi predstaviti dva različna koncepta: stopnjo dekompozicije ter proces načrtovanja. Privzeta slika 20 prikazuje bolj kot dekompozicijo samega vozila specifičen in ne edini možen proces načrtovanja (od zgoraj navzdol).

### **B.7.3 Aspektni modeli**

V devetdesetih letih prejšnjega stoletja je bil razvoj informacijskih modelov zgradb usmerjen v domensko specifične modele. Predstavljeni modeli ne skušajo opisati UoD v celoti, temveč predstavljajo le ožje področje posamezne domene.

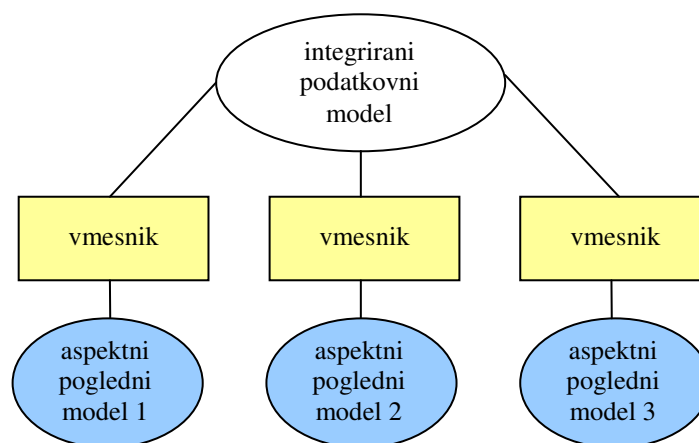
#### **B.7.3.1 COMBINE**

Projekt COMBINE (Computer Models for the Building Industry in Europe) je bil vključen v evropski program JOULE – THERMIE. V okviru dvofaznega projekta (I – 1990–1992, II – 1992–1995), ki je združeval enajst partnerjev iz sedmih držav so bili predstavljeni prvi koraki

v razvoju inteligentnega (integriranega) sistema za načrtovanje gradnje (IIBDS – Intelligent Integrated Building Design System).

V okviru projekta COMBINE so bila razvita orodja za t. i. heterogeno deljenje/izmenjavo informacij. Za razliko od homogenih izmenjav, ki se odvijajo med sorodnimi sistemi (npr. CAD – CAD) pri heterogenih izmenjavah nastopajo različni sistemi (npr. CAD – orodja za analizo toplotnega odziva zgradb). Gre za povsem naravno in tudi v praksi uveljavljeno delitev. Geometrija modela namreč predstavlja osnovo večine programskih orodij, uporabljenih v življenjskem ciklu zgradbe, izmenjavo pa zaplete vnos dodatnih negeometrijskih informacij.

Prva faza projekta je bila osredotočena na integracijo centralnega podatkovnega repozitorija, do katerega različni akterji dostopajo preko svojega uporabniškega vmesnika. Poglavitni rezultat projekta predstavlja obsežen podatkovni model (več kot 400 entitet), imenovan integriran podatkovni model (IDM – Integrated Data Model). Le-ta je bil definiran na podlagi zmožnosti orodij, ki se pri vsakodnevnem delu uporabljajo v gradbeništvu oz. na posameznih področjih, ki so bila vključena v projekt: arhitektura (geometrijsko modeliranje, strojne ter elektro instalacije (HVAC – Heating, Ventilation, Air Conditioning), toplotni odziv zgradb v povezavi z ekonomiko, načrtovanje notranjih prostorov ter načrtovanje zunanjih gradbenih elementov).



Privzeta slika 21: Aspektni modeli v projektu COMBINE (Liebich, Wix, 2002)

Adopted figure 21: Aspect models in COMBINE project (Liebich, Wix, 2002)

Za predstavitev geometrije je uporabljena t. i. RM predstavitev (RM – Relational Model), ki temelji na za gradbeništvo prilagojeni predstavitvi z mejnimi ploskvami (Brep). Izmenjava temelji na STEP fizičnih datotekah, kar predstavlja eno prvih uporab navedene specifikacije v gradbeništvu.

V okviru prve faze so bili izdelani prototipi vmesnikov, ki omogočajo izmenjavo podatkov med šestimi orodji za načrtovanje (Plokker, Augenbroe, 1994).

Cilj druge faze projekta predstavlja vzpostavitev delujočega integriranega sistema za načrtovanje zgradb (IBDS – Integrated Building Design Systems). Le-ta temelji na razširjenem ter dodatno razvitem integriranem podatkovnem modelu, razvitem v prvi fazi. Pri tem se razširitev nanaša na področje ekonomike – stroškov, energijske učinkovitosti ter novih CAD sistemov, na novo pa so bili vključeni osvetlitev, gradbena regulativa ter evropski standardi.

Najpomembnejšo razširitev predstavlja centralni sistem za izmenjavo podatkov (DES – Data Exchange System), ki med drugim omogoča dostop in izmenjavo informacij preko predstavljenega formata zapisa. Objektno usmerjene zbirke podatkov v COMBINE II nadomeščajo ASCII datoteke, hkrati pa omogočajo hranjenje na strežnikih in izmenjavo preko spleta.

Projekt COMBINE predstavlja pomemben mejnik v zgodovini modeliranja, zlasti pri razvoju pogledih modelov. Podrobneje velja predstaviti v projektu specificirano večslojno taksonomijo integriranih sistemov (Augenbroe, 1995):

- prvi sloj: skupen jezik. V predstavljenem sloju je v formalni obliki zajeto znanje o UoD. Obstaja več primernih načinov predstavitve vsebine, pri čemer je izbira odvisna predvsem od domene ter vrste komunikacije (drugi sloj). Izpostaviti je potrebno še dva različna pristopa pri zagotavljanju podpore akterjem:
  - eksplicitna predstavitev pogledov posameznih akterjev na integrirani sistem s sledenjem medsebojnih relacij med pogledi,
  - brez eksplicitne predstavitve pogledov. Vsi akterji prispevajo k definiciji skupnega jezika. Za rezultirajoči skupni model se v tem primeru smatra, da je popoln v smislu preslikav specifičnih pogledov (v oziroma iz modela).

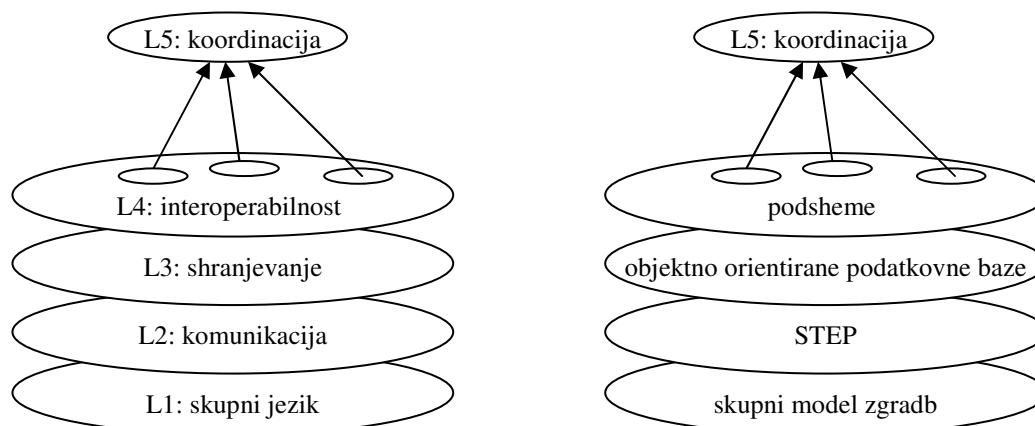
V modelu COMBINE predstavlja prvi sloj integracija že predstavljenih področij gradbeništva. Za (statični) opis modela je uporabljena enostavna predstavitev entitete – relacije (entity – relationship);

- drugi sloj: komunikacija. V ta sloj uvrščamo fizične načine povezav (preko mreže ali preko prenosljivih medijev) ter komunikacijske protokole (povezani/nepovezani OLE, CORBA ipd.).

V modelu COMBINE se pri nepovezani izmenjavi instance STEP specifikacije preslikajo v lokalne entitete (in obratno). Gre torej za statični opis oz. izmenjavo podatkov s centralnim repozitorijem. Prednost povezane izmenjave predstavlja neposredna interakcija s centralnim repozitorijem, ki omogoča izmenjavo katerekoli entitete ter izvajanje raznih operacij na celotnem modelu;

- tretji sloj: shranjevanje. Eksplicitno ter pregledno shranjevanje predstavlja pomemben dejavnik uspešnega načrtovanja. Funkcionalnost shranjevanja se praviloma opredeljuje z naslednjimi kriteriji: obvladovanje različic modela v povezavi z nadzorno sledjo (»audit trail«), dostopnost do posameznih entitet, privatne/javne različice ter shranjevanje individualnih pogledov. Vsi navedeni kriterij so upoštevani tudi pri COMBINE modelu;
- četrti sloj: interoperabilnost. V iteracijski proces načrtovanja s povratnimi zankami ter naraščajočo natančnostjo je poleg CAD sistemov vključena še programska oprema za popis del, oceno stroškov, terminske plane ipd. S predstavljenim slojem je dejansko omogočena izmenjava informacij ter evolucijski proces načrtovanja. COMBINE model zagotavlja interoperabilnost z naborom podshem. Le-te vsebujejo nabor entitet, ki so predmet izmenjave in so zato specifične vsakemu orodju za načrtovanje. Stopnja interoperabilnosti je posledično odvisna od izbranega orodja za načrtovanje, označuje pa jo kratica DTF (Design Tool Function);
- peti sloj: koordinacija. Izmenjava podatkov brez interakcijskega scenarija in ustreznega nadzora vodi v informacijski kaos. Zato je v predstavljeni integrirani sistem uveden koordinacijski sloj, ki usmerja proces načrtovanja proti zastavljenim ciljem oz. poenostavljeno: kdo dela kaj, kdaj in s kakšnim namenom. Torej je potrebno nadzorovati aktivnosti posameznih akterjev, časovno opredeliti posamezne

dogodke ter zagotavljati skladnost s predvidenim delotokom. Elementarni primer koordinacije predstavlja sledenje vsem spremembam modela, ki jih povzroča iteracijsko načrtovanje.



Privzeta slika 22: Sloji v večslojnih integriranih sistemih (Augenbroe, 1995)

Adopted figure 22: Layers in integrated multilayer systems (Augenbroe, 1995)

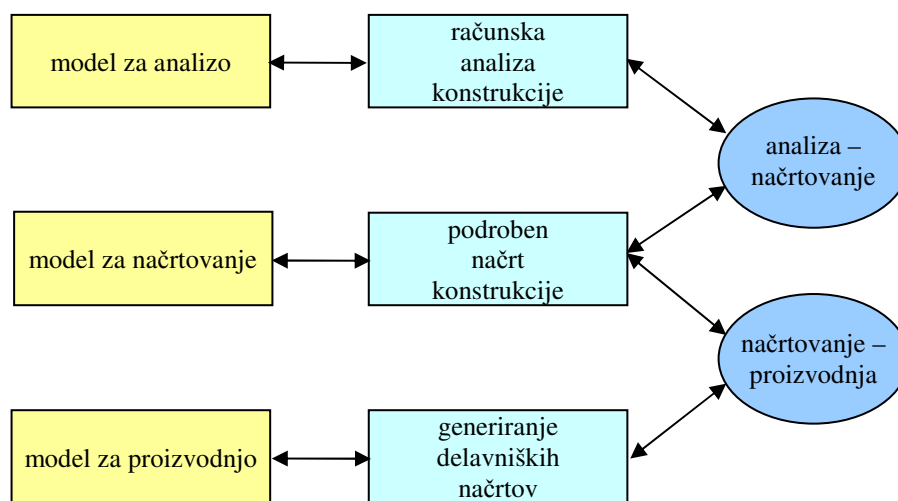
### B.7.3.2 CIS2 (cimSTEEL Integration Standard)

cimSTEEL integracijski standard (cimSTEEL Integration Standard – CIS) predpisuje specifikacijo informacijskega modela za jeklene konstrukcije. Predstavljeni model ima korenine v Eureka projektu EU 130: Računalniško integrirana proizvodnja jeklenih konstrukcij (Computer Integrated Manufacture of Constructional Steelwork – cimSTEEL). Projekt je trajal dobro desetletje (1987–1998) in je vključeval preko štirideset partnerjev iz osmih evropskih držav. Poleg izmenjave podatkov pri načrtovanju jeklenih konstrukcij projekt obravnava še nekatere druge vidike načrtovanja v predstavljeni domeni.

Prva zelo omejena različica specifikacije je bila objavljena leta 1995 (CIS1). Zaradi omejenosti modela CIS1 na prototipno rešitev omembe vrednih praktičnih implementacij v programsko opremo ni bilo. Projekt EU 130 se je sicer uradno zaključil leta 1998, vendar pa se je začeto delo nadaljevalo. Razvijalcem specifikacije se je pridružil še ameriški inštitut za jeklene konstrukcije (American Institute of Steel Construction – AISC). V letu 2000 je bila objavljena druga različica modela (CIS2) s popolnejšim opisom geometrije (731 entitet) ter izboljšanim upravljanjem in izmenjavo podatkov.



Tehnično osnovo modela cimSTEEL predstavlja standard STEP: koncepti modela so predstavljeni s pomočjo jezika EXPRESS (STEP – del 11), zapis v fizično datoteko temelji na STEP – del 21, geometrijski opis pa je povzet po STEP – del 42 (koordinatni sistemi, točke, krivulje). Bolj kot sam zapis modelov v fizične datoteke je zato zanimiva uporabljena arhitektura. cimSTEEL integracijski standard namreč vsebuje tri podmodele oz. domenske modele: za načrtovanje (»Design«), za analizo (»Analysis«) ter za proizvodnjo oz. načrtovanje detajlov (»Fabrication«). Med modeli obstajajo logične relacije (npr. nosilec, ki ga je bilo v računskem modelu zaradi vozlišča potrebno razdeliti, je v detajlnem modelu predstavljen kot en konstrukcijski element). Poenostavljeni procesni model (privzeta slika 23) prikazuje tri domenske modele, ki jih specifikacija označuje kot »poglede« (»View«). Nadalje je pri poimenovanju sestavnih delov posameznih domenskih modelov uporabljen izvorni dogovor: sestavine podmodela za analizo označuje termin elementi (»Elements«), podmodela za načrtovanje termin »člani« (»Members«) ter podmodela za proizvodnjo termin »deli« (»Parts«).

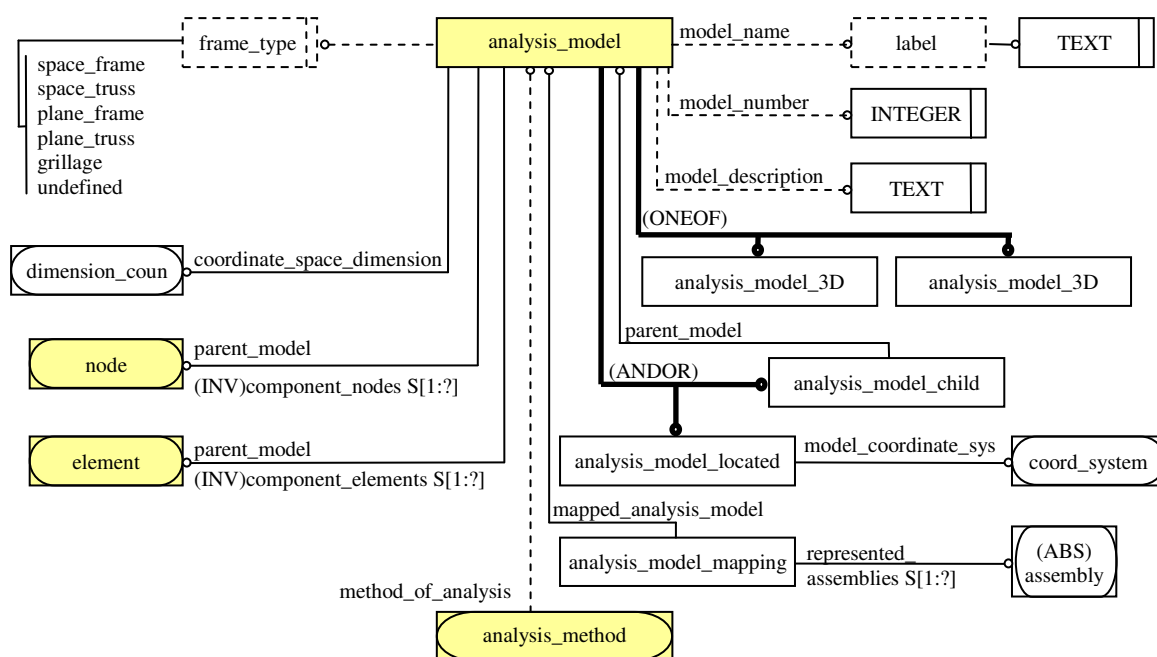


Privzeta slika 23: CIS2 in pod modeli (Eastman, 2000)

Adopted figure 23: CIS2 and submodels (Eastman, 2000)

Atributi vrhnje entitete modela za analizo (»analysis\_model«) opisujejo splošne lastnosti modelirane konstrukcije (privzeta slika 24). Med drugim je moč opisati, podati ime, oznako in opis modela, prostorsko dimenzijo ter nanjo navezujočo se vrsto konstrukcije (prostorski

okvir, prostorsko paličje, ravninski okvir, ravninsko paličje, brana, nedefinirano) in vrsto analize (statična, dinamična, psevdodinamična). Znotraj modela za analizo je možno definirati več hierarhično urejenih podmodelov (»analysis\_model\_child«). Le-temu lahko pripada svoj koordinatni sistem in samostojna lega v prostoru. Otipljivi deli modela za analizo so predstavljeni z elementi (volumski, površinski, krivuljni, točkovni) ter vozlišči, ki jim lahko določimo ime, oznako, ustrezen sklic ter robne pogoje (logični – prosto ali togo podprto, linearno oz. elastično ter nelinearno). Dodatno se lahko definira tudi sprostitve (številka, opis ter oznaka sprostitve). Relacije med vozlišči oz. elementi in modelom so dvosmerne (model se lahko sklicuje na elemente in vozlišča, preko inverzne relacije pa je možno tudi obratno). S sklicevanjem (»analysis\_model\_mapping«) je možno model za analizo povezati z ostalimi pod modeli.

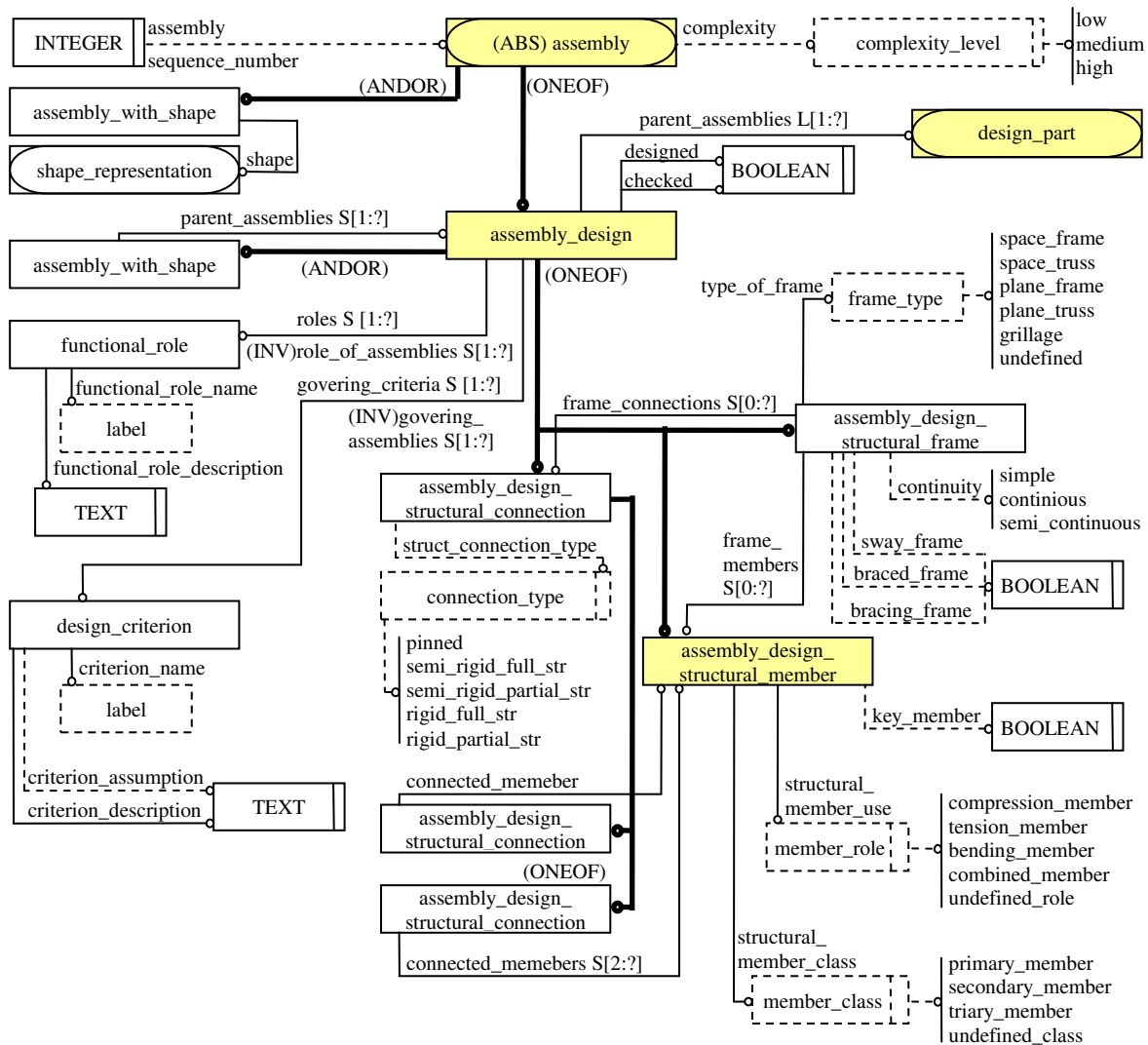


Privzeta slika 24: CIS2 model za analizo (EXPRESS-G) (Eastman, 2000)

Adopted figure 24: CIS2 analysis model (EXPRESS-G) (Eastman, 2000)

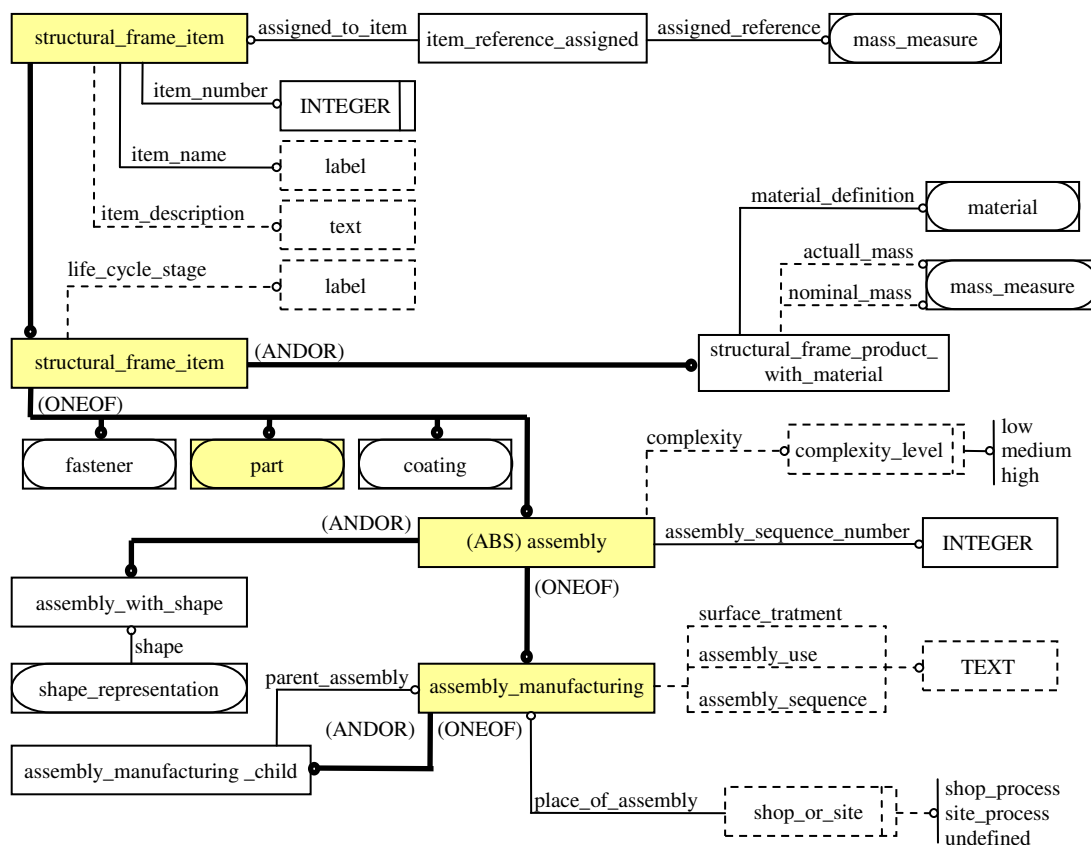
Vrhno entiteto modela za načrtovanje (privzeta slika 25), ki jo je moč instancirati, predstavlja entiteta »assembly\_design«, ki lahko podeduje attribute abstraktne nadentitete »assembly«. »Design\_part« predstavlja pomembnejšo referenco k entitetam »assembly\_design«, s

pomočjo katere združujemo gradnike v model. Podobno kot pri modelu za analizo je tudi pri modelu za načrtovanje možno definirati hierarhično urejene strukture, pri čemer je potrebno vsak podmodel označiti kot načrtovan ali preverjen, določiti njegovo funkcionalno vlogo ter nabor kriterijev za načrtovanje. »Assembly\_design« predstavlja visoko nivojsko entiteto, katere karakteristike se dedujejo v nižje ležeče entitete: v okvirje, elemente oz. po predpisani notaciji v člane ali v vozlišča (okvirji so definirani kot skupek slednjih dveh). Elemente oz. člane nadalje lahko razdelimo v linearne (stebri, prečke, elementi paličja, elementi zavetrovanja, kabli, vezi ipd.), ravninske (plošče, stene, poševnine, stopniščni elementi) ter prostorninske (nadstropja, stopnišče, jedra, lupine ipd.).



Privzeta slika 25: CIS2 model za načrtovanje (EXPRESS-G) (Eastman, 2000)

Adopted figure 25: CIS2 design model (EXPRESS-G) (Eastman, 2000)



Privzeta slika 26: CIS2 model za proizvodnjo (EXPRESS-G) (Eastman, 2000)

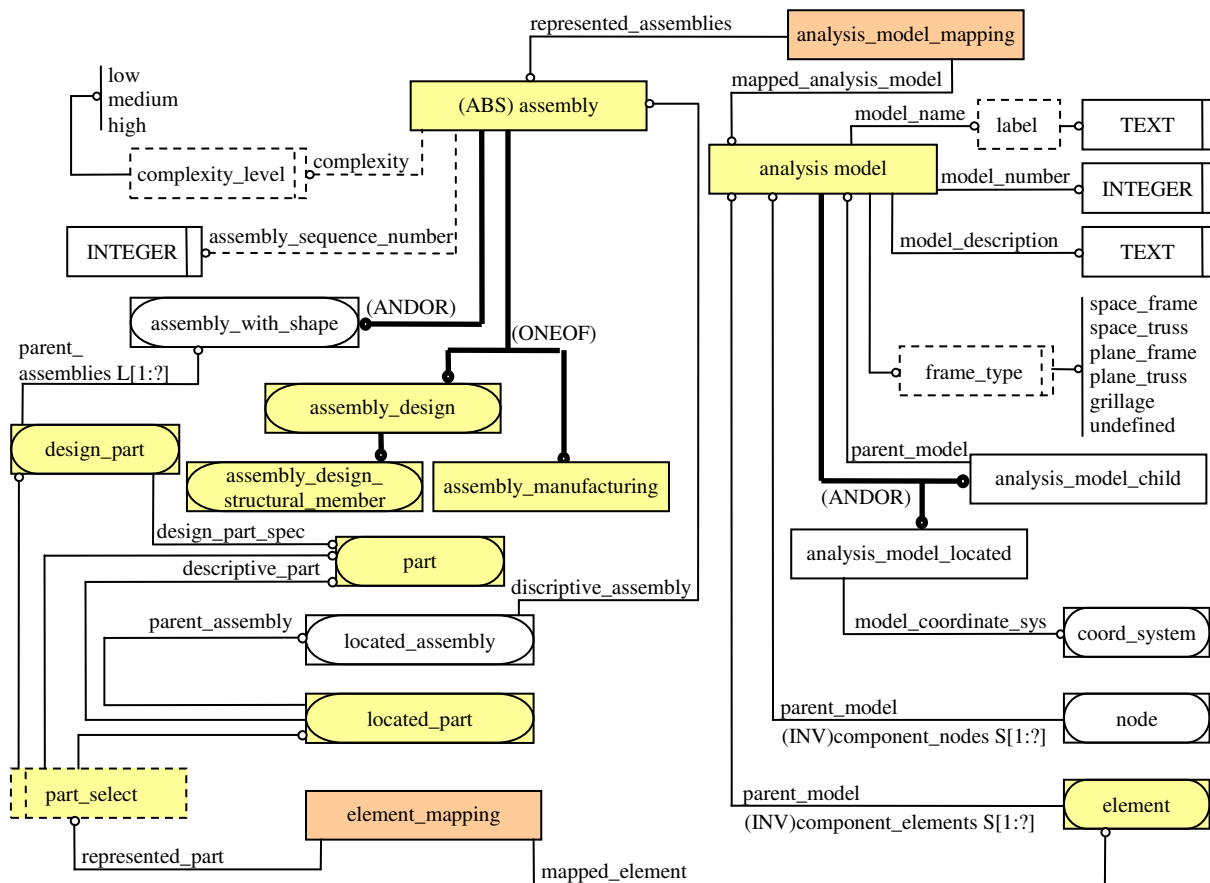
Adopted figure 26: CIS2 fabrication model (EXPRESS-G) (Eastman, 2000)

Model za proizvodnjo (privzeta slika 26) predstavlja verodostojen zapis informacij o produktu, s pomočjo katerih je možno slednjega tudi izdelati. Praktični model za proizvodnjo predstavlja izpopolnjeni model za načrtovanje, saj slednji ne omogoča opisa vseh karakteristik produkta, potrebnih za njegovo izdelavo. Vrhno entiteto, ki jo je moč instancirati v modelu za proizvodnjo, predstavlja entiteta »assembly\_manufacturing«. Podobno kot »assembly\_design« v modelu za načrtovanje predstavljena entiteta lahko deduje attribute sledečih entitet: »assembly«, »structural\_frame\_product«, »structural\_frame\_item«. »Assembly\_manufacturing« je lahko definirana hierarhično, njene reference pa so lahko enake ali različne od uporabljenih v modelu za načrtovanje (velja za entitete »parts«). Entiteta »part« je sicer definirana že v modelu za načrtovanje, vendar pa se v modelu za proizvodnjo za njeno referenciranje namesto »design\_parts« uporablja entiteta »located\_parts«, ki

dejansko predstavlja individualni element (oz. del) jeklene konstrukcije. Ta entiteta med drugim enolično določa lego z entiteto »part« definiranih splošnih elementov (oz. delov), dodatni atributi pa omogočajo še podrobnejši opis.

Čeprav vsi predstavljeni modeli opisujejo isto zgradbo, se žarišče ter stopnja podrobnosti opisa predstavljenih modelov lahko razlikuje. Zato ni nujno, da so komponente posameznih modelov medsebojno skladne, tudi če opisujejo iste primitive. Zato je bilo potrebno na podlagi primerjav entitet vseh treh modelov izdelati ustrezne preslikave (EXPRESS in EXPRESS-G shemi) .

V nadaljevanju bodo predstavljene konceptualne preslikave med modelom za analizo ter preostalima modeloma (privzeta slika 27).



Privzeta slika 27: CIS2 – preslikave med pod modeli (EXPRESS-G) (Eastman, 2000)

Adopted figure 27: CIS2 – mapping between submodels (EXPRESS-G) (Eastman, 2000)

Vrhno entiteto modela za analizo predstavlja »analysis\_model«, medtem ko primerljivi entiteti modela za načrtovanje oz. za proizvodnjo predstavljata »assembly\_design« ter »assembly\_manufacturing«. Povezava (»analysis\_model\_mapping«) na nivoju sestavov (»assembly«) identificira skladnost posameznih modelov, medtem ko se na nivoju osnovnih gradnikov modela skladnost ugotavlja z »element\_mapping«. Dejansko se ugotavlja relacije med elementi (»elements«) ter med entitetami »part«, »design\_part«, »located\_part«. Posamezen element je lahko povezan le z eno izmed navedenih entitet, medtem ko je več elementov lahko povezanih z eno izmed navedenih treh entitet (npr. nosilec podpira več prečk).

Ne glede na namen modeliranja in posledično izbrani model CIS2 specifikacije pri vseh treh prikazanih modelih kot osnovni gradniki nastopajo jekleni profili: med drugim so opisani tudi s prečnim prerezom, dolžino, spremembami na konceh profila, točkami pritrjevanja oz. vozlišči, odprtini v profilu, ojačitvami ipd. Prečni prerez se lahko definira na dva načina: parametrično (npr.: I profil se lahko podaja z višino, širino ter debelino pasnice in stojine) ali preko knjižnične oz. kataloške reference.

Specifikacija CIS2 je bila načeloma razvita predvsem za opis jeklenih okvirjev, vendar pa jo je do določene mere vseeno mogoče aplicirati na ostale jeklene konstrukcije.

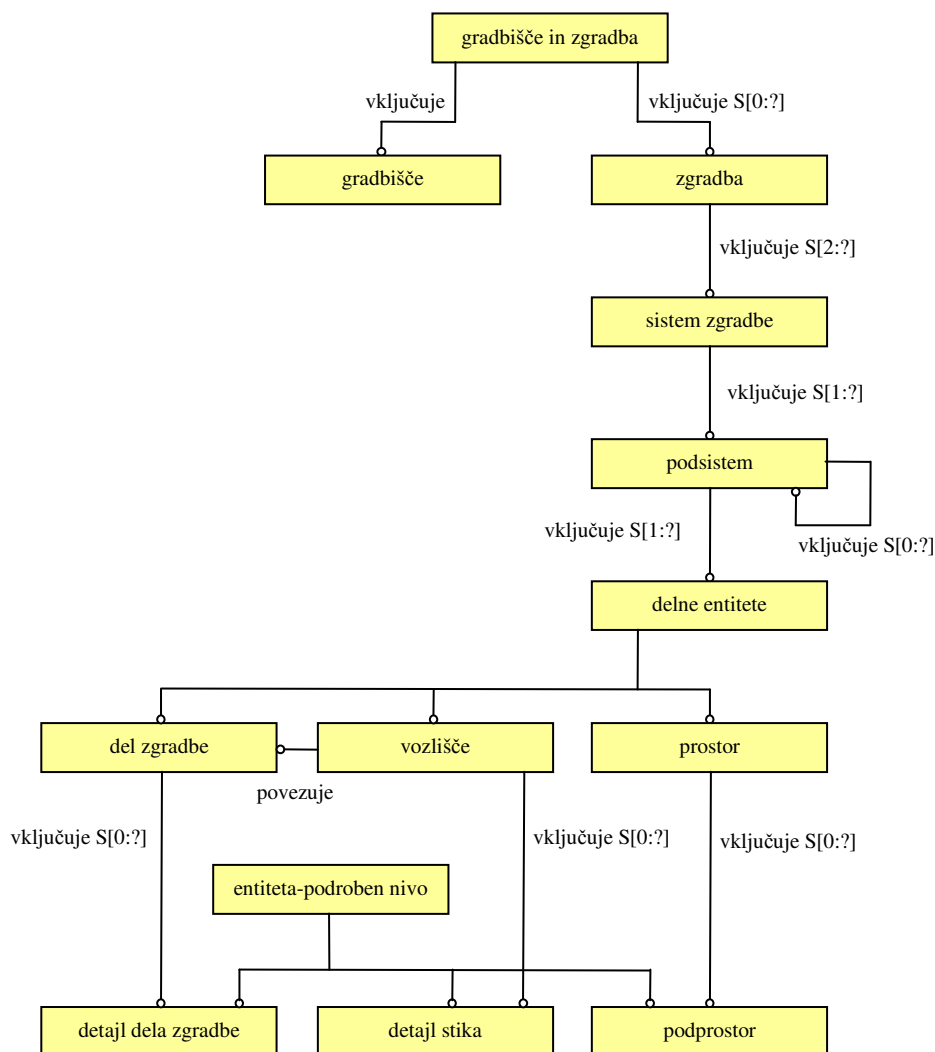
Izbira modelov za implementacijo je odvisna od narave posamezne aplikacije. Redke so aplikacije, ki pokrivajo celotno CIS2 specifikacijo, običajno so implementirani le posamezni domenski modeli.

Uporabo CIS2 specifikacije za elektronsko upravljanje in deljenje informacij je po obsežni študiji predpisal tudi ameriški inštitut za jeklene konstrukcije AISC. Uspešnost modela CIS2, ki jo potrjujejo številne implementacije, raziskovalci pripisujejo omejitvi na jeklene konstrukcije. Implementacija domensko usmerjenih informacijskih modelov je namreč enostavnejša in zato tudi uspešnejša.

Model CIS2 naj bi predstavljal osnovo za STEP aplikacijski protokol 230, vendar je bila namera zaradi pomanjkanja sredstev opuščena.

### B.7.3.3 RATAS

S projektom RATAS je finski raziskovalni inštitut VTT skušal definirati temelje za računalniško integrirano gradbeništvo (CIC – Computer Integrated Construction), in sicer na podlagi že predstavljene slojevite strukture. V modelu so primitivi opisani z entitetami, atributi (numerične vrednosti, tekst ali slike) ter ustreznimi razmerji med entitetami. Praktično so tudi razmerja predstavljena z entitetami.

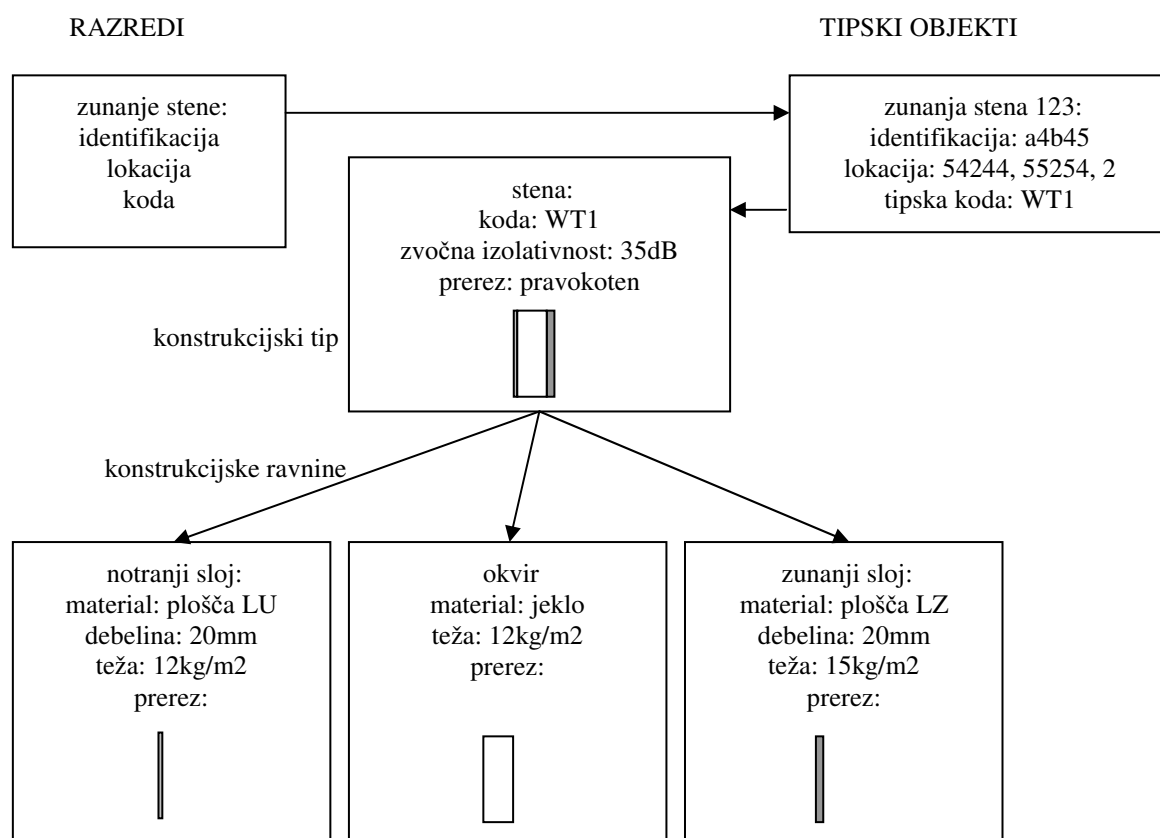


Privzeta slika 28: Ogrodje modela RATAS (EXPRESS-G) (Björk, 1995)

Adopted figure 28: RATAS framework (EXPRESS-G) (Björk, 1995)

Specifično lastnost modela RATAS predstavlja njegova generična abstraktna hierarhija (privzeta slika 28). Objekti v višjih slojih so namenjeni generičnemu opisu zgradbe v začetnih fazah projektiranja, primitivi, ki se nahajajo v nižjih slojih, pa so bolj »fizične« narave in vsebujejo več atributov. Hierarhijo sestavlja pet slojev (privzeta slika 29):

- zgradba in lokacija (v posameznem projektu je lahko zgrajenih več zgradb);
- zgradba (opis zgradbe kot celote – npr. s podatki o vrsti zgradbe, gabaritih, stroških gradnje ipd.);
- sistem zgradbe (opis funkcij zgradbe, ki jih tvorijo skupine primitivov: npr. AB skelet tvori nosilno konstrukcijo zgradbe, prostori v zgradbi tvorijo sistem prostorov, vodovodne, elektro ter ostale instalacije tvorijo sistem instalacij ipd.);
- podsistem (opisuje delitev posameznega sistema zgradbe na njegove dele – npr. prefabricirana AB plošča).



Privzeta slika 29: Tipski objekti v informacijskem modelu RATAS (Björk, 1995)

Adopted figure 29: Type object in RATAS information model (Björk, 1995)



Detalji predstavljajo najštevilčnejšo skupino objektov, saj predstavljajo večino primitivov zgradb (npr. posamezen element AB plošče) s pripadajočimi atributi (tipičen primer atributov sta oblika ter lokacija). Pri tem je potrebno izpostaviti, da je posamezen podsistem lahko del nekega drugega podsistema (rekurzivna dekompozicija).

V modelu RATAS je bila za področje informacijskega modeliranja zgradb prvič jasno opredeljena informacijska struktura »tip objekta« (»type object«). Termin izvira iz gradbene prakse, kjer se za rešitev konkretnih problemov uporablja »tipske« rešitve (npr. specifične tehnične rešitve za izdelavo stene). Predstavljena informacijska struktura se uvršča v sloj »generičnega opisa produkta«, z njo pa lahko opisujemo generične sestavine informacijskega modela zgradb (npr. okna, pri katerem so določene informacije poznane (npr. dimenzije), nekatere pa ne (npr. natančna lokacija)). Tip objekta ni enakovreden razredu, saj prvi termin predstavlja informacijsko strukturo v informacijski bazi (in ne v konceptualni shemi). Izbrani tip objekta tako predstavlja instanco pripadajočega razreda objektov, definiranih v konceptualni shemi in je s tega vidika enak pojavi instance specifičnega razreda, to je posamezni entiteti. Predstavljeni mehanizmi so v določeni meri prisotni tudi v večini aplikacij za CAD načrtovanje, in sicer v obliki knjižnic. Predpripravljeni primitivi se lahko pri načrtovanju pojavijo večkrat, in sicer na različnih lokacijah, njihova uporaba pa zmanjšuje obseg modela ter pospeši samo načrtovanje.

Model RATAS z izjemo prototipov ni bil nikoli implementiran v praksi. V manjšem obsegu (do 50 razredov oz. 2000 objektov) so bili na štirih prototipih z različnimi načini shranjevanja podatkov (relacijske podatkovne baze, hipertekst ter CAD sistem v povezavi s podatkovnimi bazami) raziskovane prednosti/slabosti posameznih pristopnih rešitev.

#### **B.7.4 Pogledni (delni) modeli**

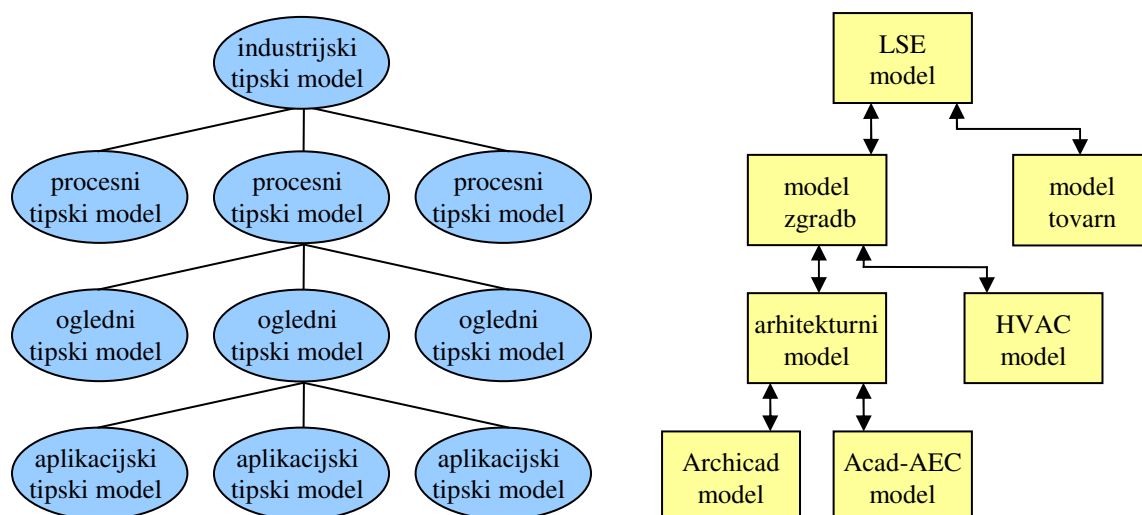
Ogledni modeli posplošujejo ter razširjajo doseg aspektnih modelov. Namesto ločenega opisa posameznih disciplin so se z oglednimi modeli raziskave osredotočile na prepletajoče se modele, ki skušajo pokriti celotne zahteve posamezne industrijske panoge.

##### **B.7.4.1 ATLAS**

V okviru projekta ATLAS (Architecture, Methodology and Tools for Computer Integrated Large Scale Engineering) so bile izvedene raziskave, kako strukturirano razčleniti informacije

o produktih znotraj AEC-FM sektorja. Izhodišče projekta predstavlja sicer referenčno usmerjeni IRMA model (Integration Reference Model Architecture).

Definiran je bil koncept tipskih modelov (»type models«), kjer se informacije progresivno specializirajo glede na izbrano stopnjo opisa. Tipski modeli so nadalje glede na izbrano stopnjo podrobnosti opisa razdeljeni v posamezne hierarhično razvrščene plasti oz. ravnine: povsem splošni koncepti, zbrani v vrhu hierarhičnega drevesa, prehajajo v aplikacijsko specifične koncepte, zbrane v aplikacijskih modelih na dnu strukture (privzeta slika 30).



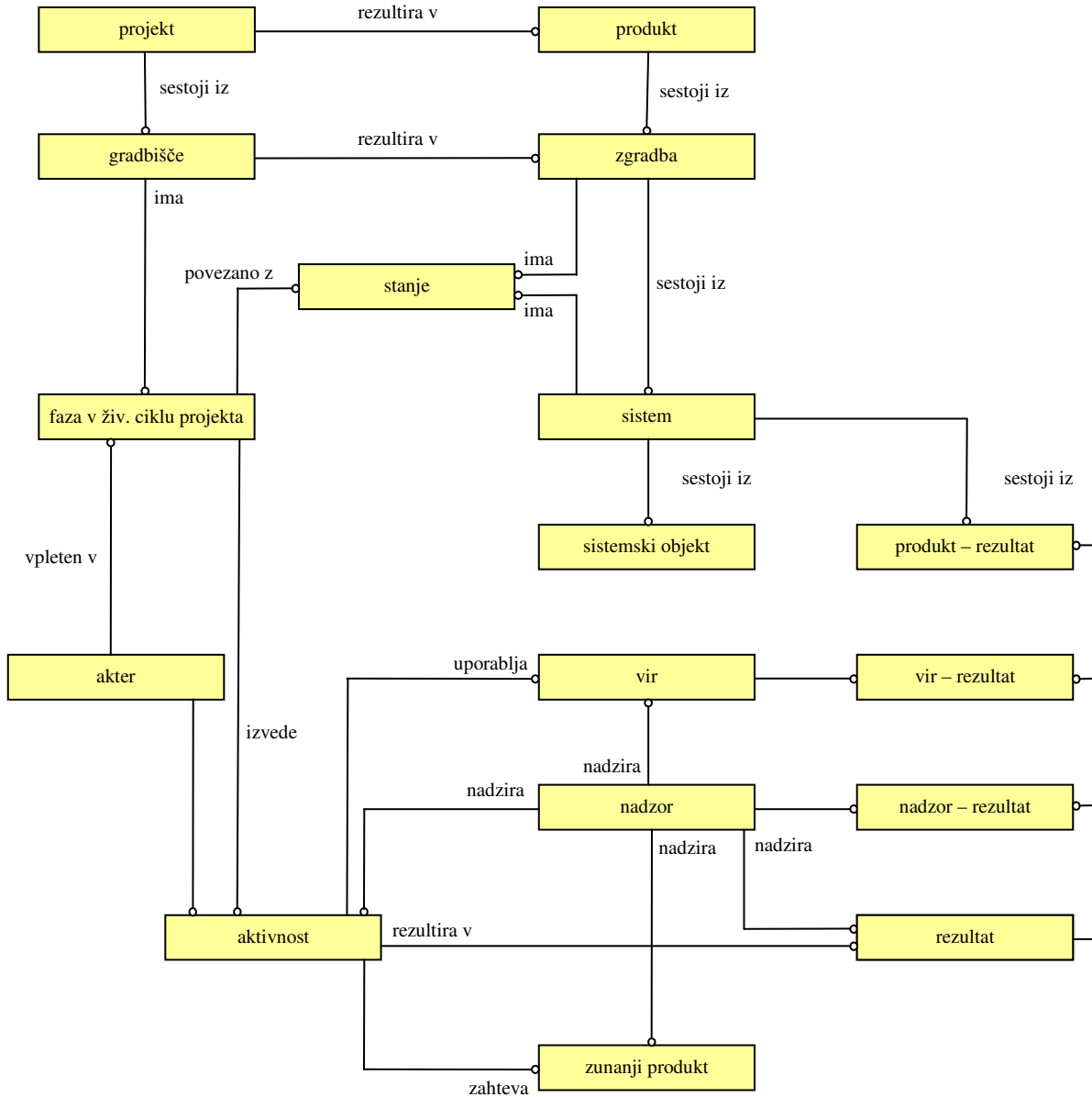
Privzeta slika 30: Tipski objekti v informacijskem modelu ATLAS (Liebich, Wix, 2002)

Adopted figure 30: Type object in ATLAS information model (Liebich, Wix, 2002)

V projektu ATLAS je bil prvič podrobneje opredeljen pomen ter hkrati razlika med največkrat opredeljenimi termini pri informacijskem modeliranju ((Tolman, 1999), (Froese, 1996)).

- Entiteta »produkt« (»product«) predstavlja korenisko entiteto tradicionalnega modeliranja v produktno usmerjenih aplikacijah (npr. za načrtovanje geometrije).
- Entiteta »proces« (»process«) označuje korenisko entiteto modeliranja, kjer so v ospredju aktivnosti (npr. v aplikacijah za terminsko načrtovanje).
- Entiteta »vir« (»resource«) predstavlja korenisko entiteto pri rabi programske opreme za načrtovanje virov (npr. načrtovanje gradbišča).

- Entiteta »nadzor« (»control«) označuje korenisko entiteto na področju projektnega upravljanja.



Privzeta slika 31: Del arhitekture ATLAS demonstratorja (Tolman, 1999)

Adopted figure 31: ATLAS model architecture – demonstrator (Tolman, 1999)

V projektu ATLAS je bil izdelan tudi demonstrator aplikacije naprednih integriranih informacijskih rešitev za vodenje obsežnih projektov (LSE – Large Scale Engineering

Projects). V takšne projekte je namreč vključeno veliko število sodelujočih strok, ki pri svojem delu uporabljajo raznovrstno programsko opremo. Slednjo se je v demonstratorju povezal z odprto platformo, ki omogoča semantično izmenjavo informacij z uporabo obstoječe programske opreme.

Privzeta slika 31 prikazuje arhitekturo modela ter uporabo slednje na primeru demonstratorja: pravokotniki označujejo posamezne modele oz. sheme, linije pa preslikave. Načeloma v višje ležečih slojih najdemo abstraktne entitete (npr. objekt ločitve namesto zunanje stene (»SeperationObject – OutsideWall«)). V vrhnjem sloju se nahaja obsežni LSE model, ki predstavlja integracijski mehanizem med posameznimi sektorji (ni nujno AEC-FM sektor). V sloju nižje najdemo integracijske mehanizme med posameznimi disciplinami izbranega sektorja (Building Model, Proces Plant Model). Nadalje je z nižje ležečimi sloji zagotovljena integracija med različnimi aplikacijami posameznih disciplin.

Na privzeti sliki 31 je prikazan le del arhitekture prototipa – vključeni so bili 2D in 3D načrtovanje in modeliranje, statična analiza, ocena stroškov, terminsko planiranje, vključitev stroškov ipd.

#### **B.7.4.2. COMBI**

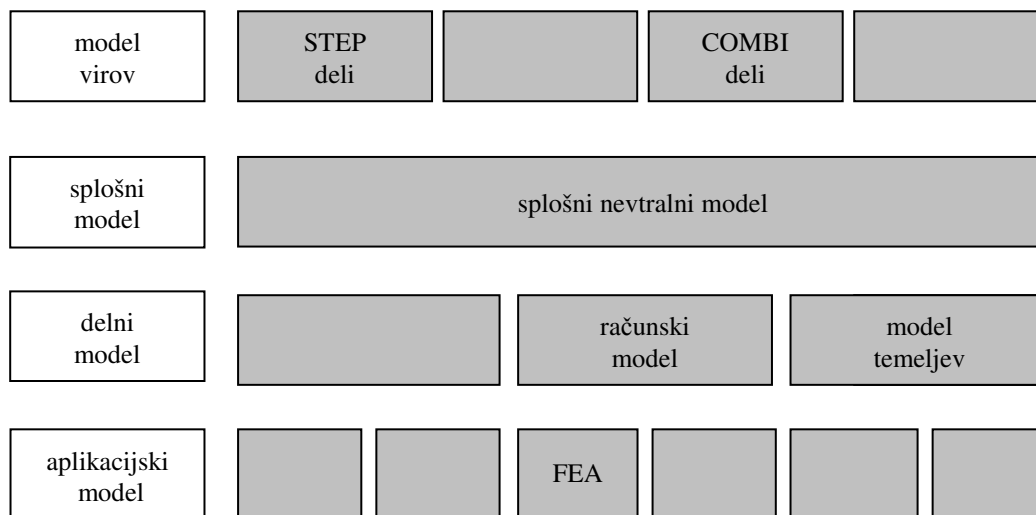
Cilj projekta COMBI (Computer Integrated Object Oriented Model for the Building Industry) je predstavljal razvoj informacijskega sistema za integracijo arhitekturnega modeliranja ter načrtovanja v domeni gradbenikov konstruktorjev s poudarkom na armiranobetonskih konstrukcijah. Podrobneje so obravnavani naslednji vidiki (Katranuschkov, Scherer, 1996):

- preslikave med različnimi aspekti istih fizičnih objektov;
- transformacije med različnimi predstavitvami objektov;
- dinamična evolucija objektov in klasifikacije.

Predstavljeni projekt temelji na ISO STEP metodologiji: konceptualni produktni model je zapisan v jeziku EXPRESS, izmenjava konkretnih podatkov pa poteka na nivoju ISO 10303-21 nevtralne datotečne specifikacije.

V projektu COMBI je bil uporabljen drugačen pristop kot pri projektu ATLAS: načrtovani modeli temeljijo na naravi informacij in ne na njihovi rabi. Ogrodje modela sestavljajo sloj virov, splošni modelni sloj, delni modelni sloj ter aplikacijski modeli (privzeta slika 32).

Aplikacijsko specifični modeli so uvrščeni v najnižji sloj in predstavljajo modele, ki so načeloma lahko neodvisni in praviloma specifični za izbrano nalogo (npr. MKE analiza). Na drugem nivoju se nahajajo delni (aspektni) modeli, ki opisujejo funkcionalnost ter karakteristike posamezne domene (npr. arhitekture). Ti modeli so razviti povsem generično ter opisujejo tudi integracijske zahteve oz. potrebe domensko specifičnih orodij za načrtovanje. Posebnost modela predstavlja minimalno jedro modela (splošni nevtralni model), ki tvori osnovo za nadaljnji razvoj. Skupaj z delnim modelnim slojem tvorita skupno jedro integracijskega ogrodja. Višje ležeči sloj virov obsega izbrane dele standarda STEP ter v okviru projekta dodatno definirane vire. Takšna razvrstitev informacij predstavlja večslojno arhitekturo modela z uporabo centralnega jedra ter delnih modelov.



Privzeta slika 32: Ogrodje modela COMBI (Katranuschkov, Scherer, 1996)

Adopted figure 32: COMBI framework (Katranuschkov, Scherer, 1996)

V sklopu demonstratorjev projekta so bila razvita oz. ustrezno dopolnjena štiri programska orodja za statično analizo, ki jih pri svojem delu uporabljajo gradbeniki konstruktorji: mehanika tal, temeljenje, statična in dinamična analiza ter načrtovanje armature. Navedena programska orodja temeljijo na produktnem modelu zgradb, s čimer je omogočena njihova integracija ter povezava z eksternim CAD sistemom za načrtovanje.

## **B.7.5 Aplikacijski protokoli**

Prvotni namen ISO STEP metodologije (danes poznane kot družina ISO 10303 standardov) je bila uporaba trislojne ANSI/SPARC arhitekture za generacijo industrijsko specifičnih modelov. Poskus, da bi navedeno dosegli z leta 1989 objavljenim standardom za izmenjavo podatkov IPIM (Integrated Product Information Model), v praksi ni bil uspešen. Poglavitna vzroka za neuspeh predstavljata kompleksnost modela ter nezadostna podpora v programski opremi.

Zato je bil v drugem poskusu izbran drugačen pristop implementacije. Uporabljen je bil koncept aplikacijskih protokolov (AP), poznan iz IGES specifikacije. Z aplikacijskimi protokoli se natančno opiše podatke, ki se izmenjujejo v specifičnem primeru izmenjave, pri čemer se pri opisu uporabi v okviru standarda STEP pripravljene vire oz. sredstva. Posledično je praktična izmenjava podatkov implementirana na nivoju aplikacijskih protokolov. ISO STEP metodologija je sicer podrobneje predstavljena v nadaljevanju.

### **B.7.5.1 ISO 10303 Part 225**

Analiza nemške AEC-FM industrije konec osemdesetih let je med drugim pokazala potrebo po izmenjavi kompleksne geometrije gradbenih produktov. Pod okriljem ministrstva za gradbeništvo ter s sodelovanjem nemškega združenja avtomobilske industrije VDA (delovne skupine za načrtovanje tovarn) je leta 1993 nastal aplikacijski protokol z imenom »elementi zgradb z eksplicitnim prikazom oblik«. Nekoliko spremenjeni protokol je bil kasneje standardiziran, in sicer kot ISO 10303-del 225.

Del 225 ne opisuje le posameznih elementov gradbenih produktov, temveč tudi umestitev produkta (okoliški teren). Elemente opisa je možno združevati v zbirnike (npr. strehe stopnišča ipd.), dodatno pa se lahko pripiše klasifikacijske ter negrafične attribute.

### **B.7.5.2 STEP CDS**

Raziskave o implementaciji informacijskih tehnologij v AEC-FM sektorju potrjujejo prevladujočo rabo 2D orodij za načrtovanje. Nezadovoljstvo nad kvaliteto orodij za 2D CAD načrtovanje ter pripadajočih vmesnikov za izmenjavo podatkov je konec devetdesetih let botrovalo nastanku STEP CDS (Construction Drawing Subset) iniciative. V združenju, ki so ga ustanovili gradbeni oddelki velikih avtomobilskih proizvajalcev ter izdelovalci CAD

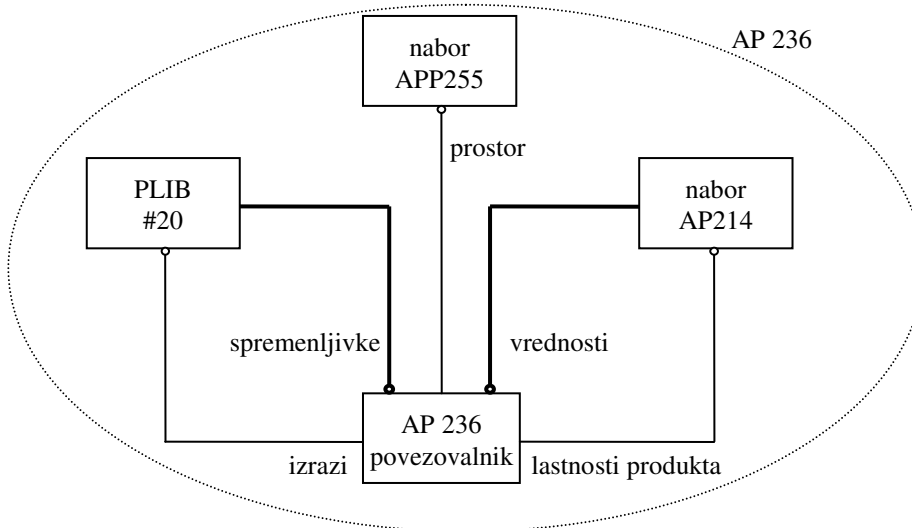
programske opreme, ni bila sicer razvita nova specifikacija, temveč se je preko STEP CDS interesnega združenja poskrbelo za kvalitetne implementacije specifikacije ISO 10303-del 214.

Podobne pobude je moč zaslediti tudi v Koreji (KOSDIC) ter na Japonskem (SCADEC). Ker se navedene iniciative nanašajo predvsem na izmenjavo risb oz. načrtov, ne bodo podrobneje predstavljene.

### B.7.5.3 FunSTEP

Cilj raziskovalnega projekta FunSTEP, delno financiranega s strani Evropske komisije (program ESPRIT), je predstavljala implementacija standarda STEP v pohištvno industrijo. Potrebe po učinkoviti izmenjavi informacij v navedeni panogi so identične potrebam v celotni AEC-FM industriji: izmenjava grafičnih ter negrafičnih informacij o produktu oz. splošna izmenjava kataloških informacij.

V okviru projekta je bil razvit aplikacijski protokol 236 (Produktni in projektni podatki za pohištvno industrijo), v katerem je definiran opis izdelkov pohištvne industrije ter izmenjava podatkov o le-teh med načrtovalci notranje opreme, proizvajalci, dobavitelji, prodajalci ter končnimi uporabniki. Opis lahko vključuje tudi kataloge oz. knjižnice produktov.



Privzeta slika 33: Globalna arhitektura aplikacijskega protokola 236 (Liebich, Wix, 2002)

Adopted figure 33: Application protocol 236 global architecture (Liebich, Wix, 2002)

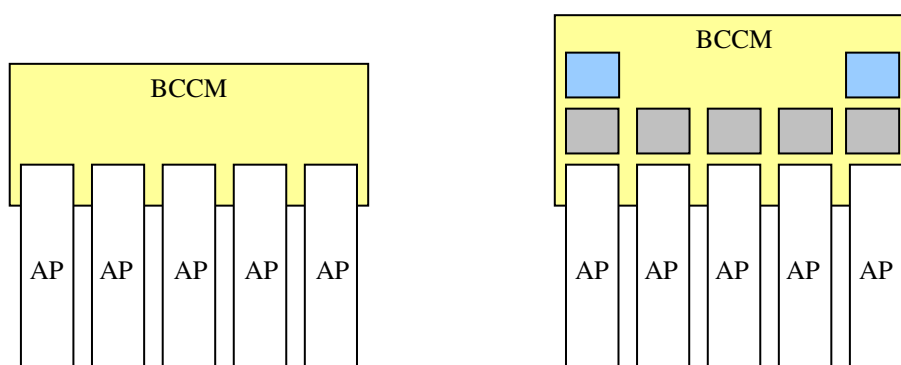
Arhitektura AP temelji na ponovni rabi že obstoječih sorodnih aplikacijskih protokolov: AP 214 (Bistveni podatki procesa strojnega načrtovanja v avtomobilski industriji), AP 225 (Elementi zgradb z eksplisnim prikazom oblik) ter knjižnice Plib#20. Aplikacijski protokol 236 (Produktni in projektni podatki notranje opreme), razvit v projektu FunSTEP povezuje vse omenjene AP (privzeta slika 33). Arhitektura AP236 je bila podedovana iz AP 214, protokol 225 pa je bil dodatno razširjen z možnostjo definicije prostora (npr. soba ipd.).

### B.7.6 Jedrni modeli

Tipski modeli projekta ATLAS, kjer se informacije progresivno specializirajo glede na izbrano stopnjo opisa ter strukturo delnih modelov projekta COMBI, so pokazali potrebo po generalizaciji določenih aspektov AEC-FM sektorja. S tem bi bila poenostavljena komunikacija med posameznimi disciplinami. Praktično to pomeni delitev osrednjega dela modelirane domene, ki jo običajno označuje termin jedro modela.

#### B.7.6.1 BCCM

Razvoj modela zgradb BCCM (Building Construction Core Model) se je pričel leta 1994, in sicer z vzpostavitvijo delovne skupine za področje gradbeništva v okviru ISO STEP pobude. Model BCCM je bil prvotno načrtovan kot del standarda STEP (del 106). Za razliko od ostalih aplikacijskih protokolov, ki izbrano področje opisujejo zelo natančno, je bila BCCM pobuda precej širše zasnovana, saj je skušala zadovoljiti potrebe po interdisciplinarni izmenjavi informacij.

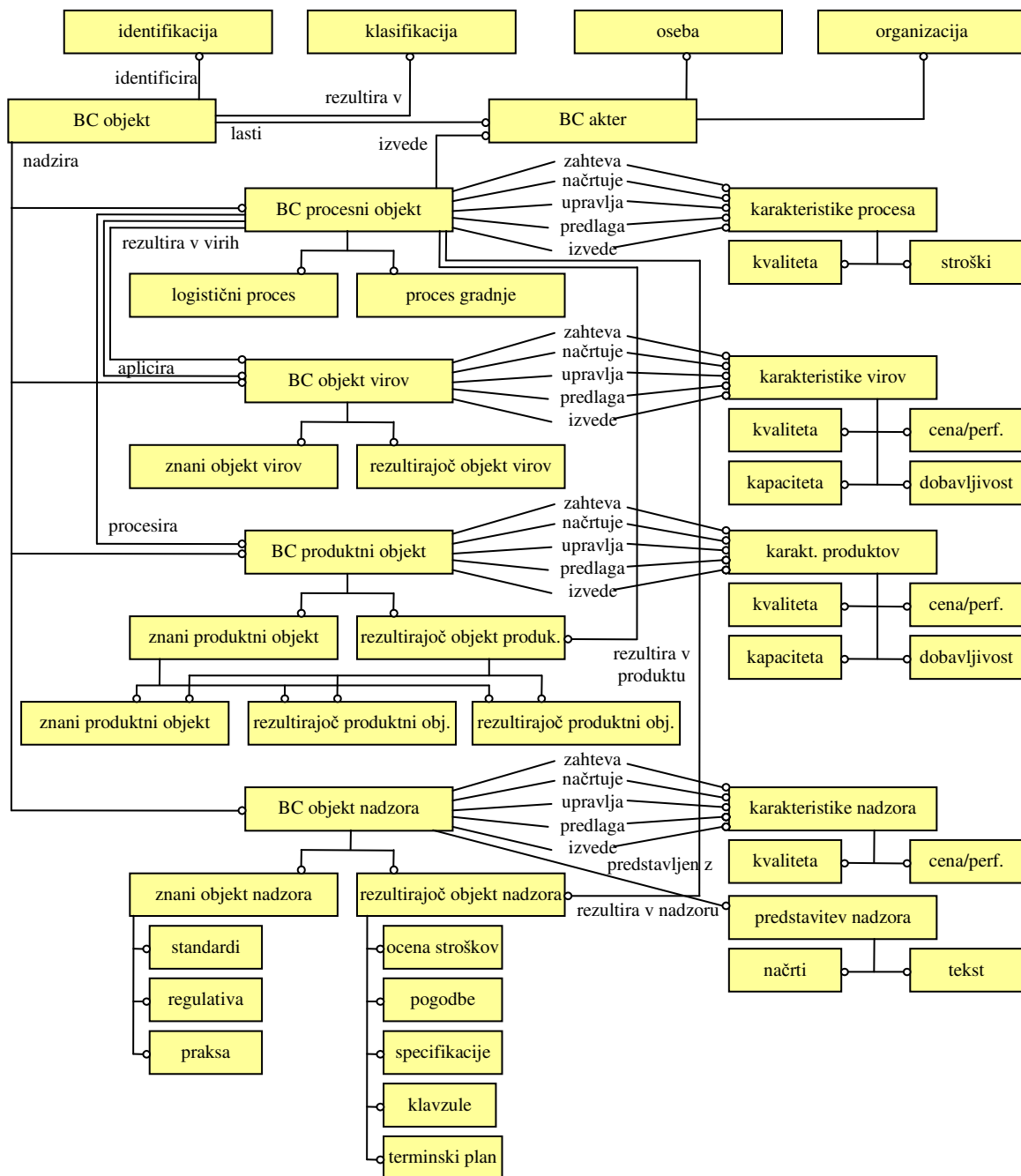


Privzeta slika 34: Arhitektura BCCM – predlog in implementacija (Liebich, Wix, 2002)

Adopted figure 34: BCCM model architecture – proposal and implementation (Liebich, Wix, 2002)



S tehničnega vidika je BCCM progresivno združil koncepte modelov ATLAS ter COMBI (privzeta slika 34). V začetni fazi razvoja je model temeljil na jedrni zasnovi (ekvivalentno modelu ATLAS), vendar pa je jedro modela v kasnejših fazah projekta ločeno v sloj virov ter osrednji jedrni sloj (ekvivalentno modelu COMBI) (privzeta slika 35).



Privzeta slika 35: Ogradje modela BCCM (Froese, 1996)

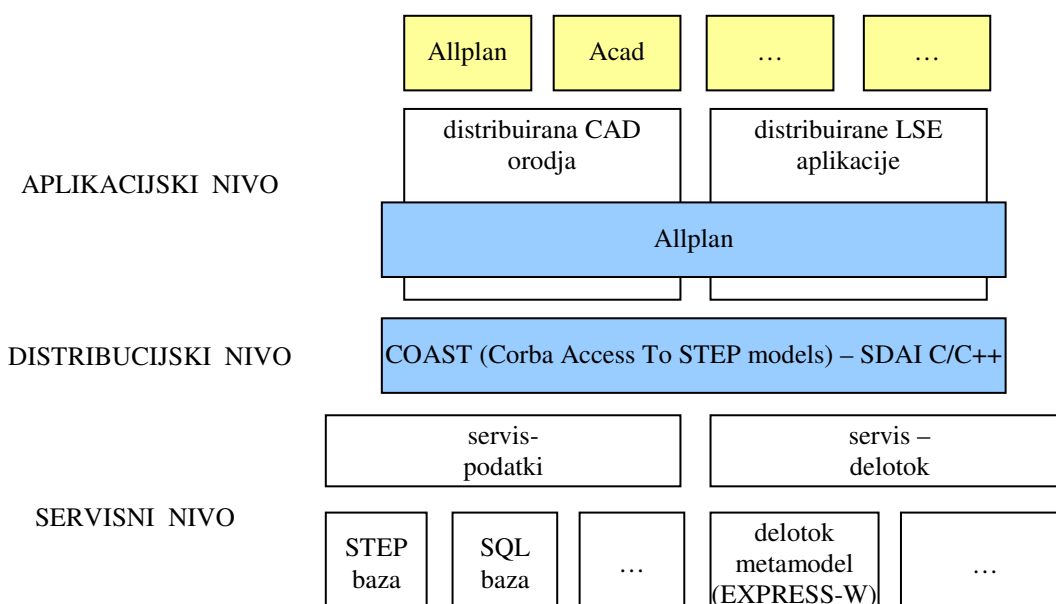
Adopted figure 35: BCCM framework (Froese, 1996)

V modelu BCCM je uporabljen nekoliko drugačen koncept razvrščanja primitivov v sloje: vsi objekti, ki jih uporablja več kot en sistem, sodijo v jedro modela. Takšen pristop rezultira v model z obsežnim jedrom ter »sateliti«, v katerih bi bili zbrani domensko specifični primitivi, njegova uporabnost pa je zaradi obsežnosti jedra ter splošnega upravljanja z modelom vprašljiva.

Avtorji BCCM modela so predvidevali združitev sorodnih pobud (del 228 – HVAC, del 230 – jeblene konstrukcije), vendar do združitve ni prišlo. Žal se je po petih letih razvoja izkazalo, da so posamezni deli modela preveč kompleksni, da bi jih bilo moč implementirati v praksi. Tudi zato se je v letu 1997 razvoj modela zgradb BCCM ustavil, ugotovitve raziskave oz. prednosti modela pa so bile v večji meri vključene v novonastajajoči model zgradb IFC.

#### B.7.6.2 VEGA

V projektu VEGA (Virtual Enterprise using Groupware tools and distributed Architectures) so se raziskovalci ukvarjali z arhitekturo konceptualnih produktnih modelov v distribuiranem okolju, s tehnologijami, ki omogočajo komunikacijo med distribuiranimi okolji (COAST), z distribuiranimi informacijskimi sistemi ter z upravljanjem delotokov ter tokov informacij.



Privzeta slika 36: Osnovni gradniki platforme VEGA (Debras et al., 1997)

Adopted figure 36: Basic elements of VEGA framework (Debras et al, 1997)

Temelje projekta predstavljajo obstoječi standardi STEP, CORBA, HTML ter VRML. Ogradnje modela je sicer povzeto po modelu COMBI, vendar je ustrezno prirejeno principom jedrnih modelov. VEGA model implementira tudi danes splošno uporabljeni koncept: posamezne ponavljajoče entitete s sheme modela se ne kopira, temveč referencira.

Pri modelu VEGA je potrebno izpostaviti koncepte objektno orientiranih tehnologij (CORBA – Common Object Request Broker Architecture, IDL – Interface Description Language), s katerim se poenostavlja dostop do vsebin v heterogenem okolju. Podobno kot vsi novejši informacijski modeli zgradb tudi geometrijski del modela VEGA temelji na STEP konceptualni predstavitvi (dejansko poglavitni del VEGA arhitekture predstavlja distribucijska tehnologija imenovana COAST – CORBA Access to STEP models) (Debras et al., 1997).

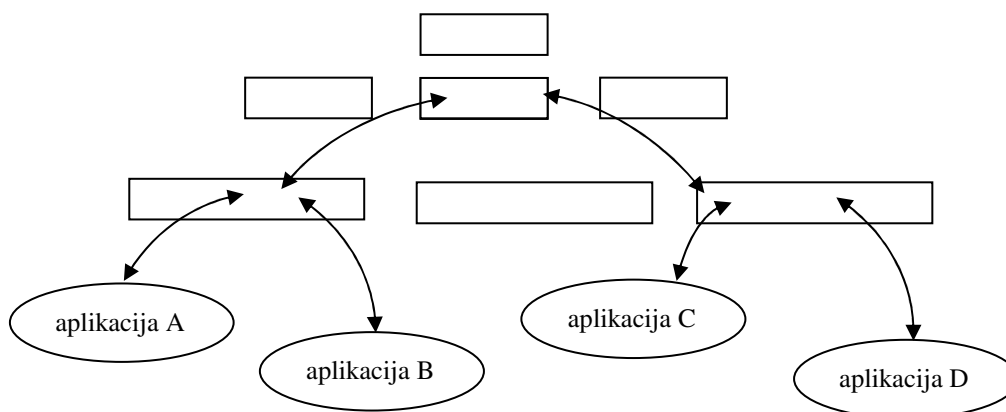
V prvem sloju so združene aplikacije končnih uporabnikov, ki jih je moč povezati s tehnologijo COAST (npr. CAD/CAM aplikacije, STEP aplikacije, PDMS aplikacije (Product Data Management Systems, ipd.). Kompatibilnost navedenih aplikacij s standardom STEP (oz. ustreznim aplikacijskim protokolom – najpogosteje z AP 214, AP 255 ter AP 228) zagotavljata IDL vmesnik (Interface Description Language) ali neposredni COAST vmesnik.

Najpomembnejši gradniki VEGA platforme so vsebovani v distribucijskem sloju. COAST tehnologija poleg že poznane izmenjave STEP fizičnih datotek ter sočasnega dostopa do baz podatkov (preko standardiziranega API) uvaja nov način dostopa do podatkov. Le-ta je prilagojen fragmentiranim ter oddaljenim informacijskim sistemom. Funkcionalnost obstoječih SDAI vmesnikov (Standard Data Access Interface) je z novim pristopom razširjena z:

- bolj transparentno distribucijo informacij – uporabniku se na primer ni potrebno več ukvarjati s povezavami oz. z naslavljanjem strežnikov, na katerih se nahajajo informacije;
- implementacijo pravilno delujočega sočasnega dostopa.

Tretji sloj se imenuje tudi sloj strežniških aplikacij in vsebuje nabor servisov, ki opravljajo specifične naloge. Predstavljeni sloj vsebuje tudi elemente za trajno shranjevanje podatkov.

Kot pomemben prispevek projekta VEGA k razvoju informacijskega modela zgradb več avtorjev ((Debras et al., 1997), (Junge et al., 1997)) navaja princip t. i. »skupnih konceptov«, ki ga je najlažje razložiti na primeru najenostavnejših primitivov – recimo stene. Definicija le-te je v večini informacijskih modelov zgradb vezana na zahteve posameznih domen. Posledično lahko v enem modelu obstaja več različnih definicij stene (npr. stena kot element arhitekture in stena kot element računskega modela). Različne predstavitve stene sicer povezuje skupni (lahko abstraktni) nadtip, vendar pa v informacijskih modelih zgradb običajno nastopajo tudi horizontalne preslikave med različnimi predstavitvami istega elementa (v našem primeru stene). Z uvedbo »skupnih konceptov« se skuša eliminirati horizontalne preslikave ter jih nadomestiti z vertikalnimi povezavami oz. s skupnimi koncepti posameznih elementov (sten). Praktičen primer (slika 46): aplikaciji A in B (ter C in D) pripadata isti domeni in komunikacija poteka preko pripadajočega domenskega modela. V primeru komunikacije med aplikacijama A in C pa poteka le-ta na višjem nivoju (skupni koncepti).



Slika 46: Komunikacije v modelu VEGA

Figure 46: VEGA model – communications

### B.7.6.3 EPISTLE

Cilj evropskega neprofitnega konzorcija EPISTLE (European Process Industries STEP Technical Liaison Executive) predstavlja razvoj specifikacij ter standardov za izboljšanje učinkovitosti upravljanja in izmenjave informacij v življenjskem ciklu produkta. V okviru konzorcija so bili razviti EPISTLE jedrni model (generični produktni model), referenčna

knjižnica, predloge (podlaga za izmenjavo informacij med aplikacijami), ustrezni priročniki ter pojmovnik. Predstavljena iniciativa je standardizirana – ISO 15926-2.

EPISTLE model se osredotoča na življenjski cikel procesnih obratov (oz. tovarn). Raziskave konzorcija EPISTLE so bile med drugim tudi podlaga za STEP aplikacijski protokol AP 221: funkcionalni podatki ter shematski prikaz procesnih obratov.

### **B.7.7 Ogrodni modeli**

Ogrodni modeli sicer temeljijo na jedrnih modelih ter ISO STEP specifikaciji, vendar pa so osredotočeni na določene industrijske panoge (npr. IFC specifikacija za področje gradbeništva oz. visokogradnje). Model IFC kot tipični predstavnik ogrodnih modelov je podrobneje predstavljen v zadnjem delu tretjega poglavja.

#### **B.7.7.1 Integrirani ogrodni modeli**

Z implementacijo ter širšo uporabo izmenjave oz. deljenja informacij na osnovi informacijskih modelov zgradb so bile ugotovljene nekatere pomanjkljivosti modelov ter uporabljenih tehnologij.

Tako je bila ne glede na obsežno in semantično bogato specifikacijo modelov vedno izkazana potreba po prisotnosti mehanizma za opis primitivov, ki niso bili opisani v specifikaciji modela. Takšna razširitev modela izkazuje potrebo po skupnem semantičnem slovarju, v katerem bi bili uporabljeni termini ustrezno obrazloženi. Takšna obrazložitev lahko skupaj s specifikacijami atributov (lastnosti), seznamom merskih enot, sinonimov in podobnih specifikacij bistveno eliminira nekonsistentnost modela.

Obseg sodobnih informacijskih modelov presega mejo učinkovite datotečne izmenjave. Slednja je primerna le še za delno izmenjavo informacij oz. za enostavne modele, medtem ko je za kompleksnejše modele primernejši pristop shranjevanja v podatkovnih bazah.

Možne rešitve navedenih pomanjkljivosti predstavljajo smernice za razvoj integriranih ogrodnih modelov, katerih osnovna karakteristika predstavlja konsistentno celovitost informacij. V takšne modele bodo poleg informacijskega modela lahko vključeni tudi referenčni procesni model ter ostali že obrazloženi eliminatorji nekonsistentnosti.

Najverjetneje pa bodo v nadaljnjih fazah razvoja integriranih ogrodnih modelov vanj vključene tudi ustrezne taksonomije/ontologije.