Pennsylvania State University
The Graduate School

Department of Architectural Engineering

AN INTEGRATED BUILDING INFORMATION MODELING (BIM) FRAMEWORK FOR MULTI-STORY MODULAR BUILDINGS

A Dissertation in
Architectural Engineering
by
Issa Jafar Ramaji

© 2016 Issa Jafar Ramaji

Submitted in Partial Fulfillment of the Requirements for the Degree of Doctoral of Philosophy

May 2016
The dissertation of Issa Jafar Ramaji was reviewed and approved* by the following:

Ali M. Memari  
Professor and Hankin Chair of Residential Construction,  
Department of Architectural Engineering and Department of Civil and  
Environmental Engineering  
Dissertation Advisor  
Chair of Committee

Robert M. Leicht  
Assistant Professor of Architectural Engineering

John I. Messner  
Charles and Elinor Matts Professor of Architectural Engineering

Kevin M. Parfitt  
Professor of Architectural Engineering

Dinghao Wu  
Assistant Professor of Information Sciences and Technology

Chimay Anumba  
Professor of Architectural Engineering  
Head of the Department of Architectural Engineering

*Signatures are on file in the Graduate School
ABSTRACT

Despite the relative simplicity of design and construction of modular single-family dwellings, the same cannot be stated for multi-story modular buildings, especially in relation to structural modeling and design of these buildings. There is potential for tackling complexity of these projects by leveraging a successful technology and process that is being widely adopted in other sectors of construction industry, and that is Building Information Modeling (BIM). Structural analytical modeling is one of the areas that can benefit from BIM to enhance the design quality; reduce time and cost of the design; and mitigate the complex design activities. Because of lack of interoperability between tools, BIM users cannot take advantage of this BIM use properly, especially in modular building projects that their special needs are not addressed in the currently available information exchange standards.

The primary goal of the proposed research is to develop an information framework and its supporting infrastructure to encourage design and construction of mid- and high-rise buildings using factory-built modular units. Achieving this would provide the industry with an option for economical, energy efficient, sustainable, and more affordable construction. In this research, product-related information that may be generated or used at different stages of projects is studied. In addition, process-related information flow throughout project phases is investigated. Moreover, interpretation of structural design/analysis model of these buildings from their architectural building information model is researched.

This research has been carried out through accomplishing several objectives. First the Product Architecture Model (PAM) for multi-story modular buildings was developed. The PAM is a structured breakdown of building elements and their related attributes and properties. The next objective was information exchange standardization in these projects focusing mainly on structural aspects. To achieve this, an Information Delivery Manual
(IDM) and Model View Definitions (MVDs) were developed based on the characteristic and workflow of modular building projects. To achieve the first two objectives, a comprehensive literature review was carried out, and a series of site visits and interviews with industry experts were conducted. The final objective was to facilitate the structural design of complicated structures of modular buildings by developing a mechanism and a supporting platform to interpret structural design/analysis model of modular buildings directly from the architectural BIM.

Although the developed PAM, IDM and MVD are focused on structural and general architectural aspects of modular buildings, they could be expanded and/or modified to supports other BIM application areas in these projects. The methodology proposed in this research for development of information exchange standards could be used for representation of current information exchange standards to come up with an integrated set of standards for building industry. This would be achieved as a result of using the same PAM for development of the information exchange standards. In addition, the interpretation methodology presented for generation of modular buildings could be implemented in other engineering design/analysis areas in both modular and site-built construction. Implementation of the Interpreted Information Exchange (IIE) concept can significantly facilitate engineering analysis BIM uses by decreasing implementation cost of MVDs in BIM authoring tools and automating the model modifications that are required to make an imported model ready for analysis. Moreover, the platform developed for testing the structural model interpretation process is a general purpose platform, which could be utilized for all-purpose BIM information extractions and implementation of different automated interpreted information exchanges.
## Table of Contents

List of Figures .................................................................................................................. xiv  
List of Tables ................................................................................................................... xxi  
Abbreviation List ........................................................................................................... xxiii  
Acknowledgements ......................................................................................................... xxv  

CHAPTER 1: Introduction ........................................................................................... 1  
1.1. Modular Building Industry ................................................................................... 1  
1.2. Statement of the Problem and Potential Solution ................................................. 5  
1.3. Research Goal and Objectives ............................................................................ 11  
1.4. Scope and Limitations ........................................................................................ 14  
1.5. Organization of the Dissertation ........................................................................ 15  
1.6. References .......................................................................................................... 16  

CHAPTER 2: Literature Review ................................................................................ 20  
2.1. Modular Buildings ............................................................................................... 20  
2.1.1. Types of Modular Buildings ........................................................................... 21  
2.1.1.1. 3D System .................................................................................................. 21  
2.1.1.2. The Open House System .......................................................................... 23  
2.1.1.3. Hybrid Cored-Modular Systems (Cluster) ................................................... 25
3.2. Objectives and Tasks Outline ................................................................. 72

3.3. Further Explanation of the Tasks ............................................................... 74
   3.3.1. Tasks for Objective 1 ........................................................................ 74
   3.3.2. Tasks for Objective 2 ........................................................................ 76
   3.3.3. Tasks for Objective 3 ........................................................................ 78

3.4. Research Process Summary .................................................................... 81

3.5. References ................................................................................................. 84

CHAPTER 4: Product Architecture Model (PAM) .......................................... 85

   Product Architecture Model for Multi-Story Modular Buildings ................. 85
   4.1. Introduction .......................................................................................... 86
   4.2. Anatomy of Modular Buildings ............................................................ 92
   4.3. Development of the PAM .................................................................... 95
      4.3.1. Development of Product Breakdown Hierarchy ......................... 97
      4.3.2. Assigning Attributes to Assemblies and Components ................ 104
      4.3.3. Categorizing Objects Based on Level of Detail ......................... 106
      4.3.4. Validation .................................................................................... 110
   4.4. Overview of the PAM .......................................................................... 115
   4.5. Applications of the PAM .................................................................... 118
   4.6. Summary and Concluding Remarks .................................................... 122
CHAPTER 5: Information Delivery Manual (IDM) ........................................... 135

A Product-Oriented Information Exchange Schema for Multi-Story Modular Building Projects ........................................................................................................................ 135

5.1. Introduction ............................................................................................... 136

5.2. BIM in Modular Building Projects ............................................................ 142

5.3. Development of the Information Framework .......................................... 148

5.4. Process Investigation ................................................................................ 153

5.4.1. Structure of the Process Model ............................................................. 154

5.4.2. Process Overview .................................................................................. 157

5.4.3. Exchange Models and Task Unit Specification .................................... 159

5.5. Product Investigation ................................................................................ 165

5.6. Exchange Requirement Specification ...................................................... 169

5.7. Uses of the Information Framework .......................................................... 172

5.8. Summary and Concluding Remarks ........................................................ 173

5.9. Acknowledgements ................................................................................... 175

5.10. References .............................................................................................. 175

CHAPTER 6: Model View Definition (MVDs) ................................................... 180
Extending the Current Model View Standards to Support Multi-Story Modular Building Construction ........................................................................................................................................... 180

6.1.  Introduction ...................................................................................................... 181

6.2.  Information Exchange Standardization ............................................................ 186

   6.2.1.  Standardization in Construction Industry ................................................. 188

   6.2.2.  Information Delivery Manual/Model View Definitions ......................... 190

   6.2.3.  Industry Foundation Classes ............................................................... 191

   6.2.4.  International Framework Dictionary ................................................... 192

6.3.  Investigation of Information Units ................................................................... 194

6.4.  Development of the MVDs .............................................................................. 197

6.5.  Extension of the Structural MVDs ................................................................. 207

6.6.  Summary and Concluding Remarks ............................................................... 217

6.7.  Acknowledgements ........................................................................................ 219

6.8.  References ....................................................................................................... 219

CHAPTER 7:  Interpreted Information Exchanges (IIEs) ........................................ 223

Interpreted Information Exchange: A Systematic Approach for BIM to Engineering Analysis Information Transformations ........................................................................................................... 223

7.1.  Introduction ...................................................................................................... 224

7.2.  Interpreted Information Exchange ................................................................. 229
7.3. A Methodology for Design and Standardization of IIEs ............................... 235

7.3.1. Concept Layer ........................................................................................... 237

7.3.2. Generic Layer ............................................................................................ 238

7.3.3. IFC Layer .................................................................................................. 241

7.4. Case Study and Validation ............................................................................... 246

7.5. Implementation of the IIE Standards ............................................................... 258

7.6. Potentials and Benefits of Implementing IIEs .................................................. 266

7.7. Summary and Concluding Remarks ................................................................. 273

7.8. References ........................................................................................................ 275

CHAPTER 8: Platform Developed for Implementation of IIEs ............................... 279

Implementation of the Interpreted Information Exchange from A Software Development Point of View ......................................................................................................................... 279

8.1. Introduction ...................................................................................................... 280

8.2. Industry Foundation Classes ............................................................................ 284

8.3. Interpreted Information Exchange ................................................................... 287

8.3.1. Interpretation Workflow ........................................................................... 291

8.4. Importing the IFC File ...................................................................................... 293

8.4.1. Reading IFC File Instances ....................................................................... 294

8.4.2. Reading IFC Schema Entities ................................................................... 296
CHAPTER 9: Interpretation of Structural Analysis Model from Coordination Model

Interpretation of Structural Analytical Model from BIM in Coordination View

9.1. Introduction

9.2. Coordination View versus Structural Analysis View

9.3. Interpreted Information Exchange

9.4. Interpretation Process

9.5. Representation Transformation and Material Information Enrichment

9.5.1. Acquiring Information about Linear Elements

9.5.2. Representation Transformation of Linear Elements

9.5.3. Representation of the Transformed Elements
9.6. Connectivity Adjustment .................................................................................. 350

9.6.1. Connectivity adjustment of Connecting Points ........................................ 351

9.6.2. Connectivity adjustment of Point-to-Line Connections .......................... 354

9.7. Validation ......................................................................................................... 357

9.8. Summary and Concluding Remarks ................................................................. 360

9.9. References ........................................................................................................ 362

CHAPTER 10: Design of an IIE for Structural Design of Modular Buildings........... 366

10.1. Introduction .................................................................................................. 366

10.2. Structural Modeling and Design Complexities ............................................ 367

10.3. Design Workflow Using IIE ......................................................................... 371

10.4. Required Interpretation ................................................................................. 374

10.5. Summary ....................................................................................................... 390

CHAPTER 11: Summary and Conclusions ................................................................. 391

11.1. Outcomes ...................................................................................................... 391

11.1.1. Product Architecture Model ................................................................. 392

11.1.2. Information Delivery Manual and Model View Definitions for Modular
Buildings 394

11.1.3. A Tool and Approach for Interpretation of Structural Analysis Model
from Building Information Model ........................................................................... 397
11.1.4. A Mechanism for Interpretation of Structural Analysis Models from Building Information Models ................................................................. 398

11.2. Research Contributions ............................................................................. 399

11.2.1. Proposing a New Information Exchange Mechanism for Facilitate Engineering Analysis BIM Uses ................................................................. 400

11.2.2. Proposing a New Methodology for Standardization of BIM Information Exchanges 401

11.3. Broader Impacts ............................................................................................ 403

11.4. Potential Future Researches ........................................................................ 406

APPENDIX A: Task Unit and Exchange Model Table Sets ...................................... 408

A.1. Task Unit Table Set .......................................................................................... 409

A.2. Exchange Model Table Set .............................................................................. 417

APPENDIX B: Exchange Requirements Specification ............................................. 425

APPENDIX C: Model View Diagrams and Concept Definitions .............................. 433
List of Figures


Figure 1-2. $4.9 billion Atlantic Yards Project (SHoP Architects) ................................................. 3

Figure 1-3. Shares of different sectors in modular building industry (McGraw Hill Construction 2011)......................................................................................................................................... 5

Figure 1-4. Prefabrication decision impact at different project stages (Gibb 1999)............ 7

Figure 1-5. McLeamy Curve (AEC Magazine 2013)................................................................. 7

Figure 1-6. BIM in prefabrication (McGraw Hill 2012)............................................................ 9

Figure 1-7. Occurrence rate distribution comparison of modular buildings and site-built construction........................................................................................................................................... 12

Figure 2-1. Modular building construction vs. site-built construction timeline .......... 21

Figure 2-2. Main components of the Open House system.................................................... 24

Figure 2-3. Schematic illustration of the hybrid cored-modular system (Left) and Hybrid podium-modular system (Right) .............................................................................................................. 26

Figure 2-4. Schematic illustration of construction of a building with Framed Unit system .................................................................................................................................................... 27
Figure 2-5. Comparing modularizations in different scales (Courtesy of National Audit Office (NAO)) .................................................................................................................................................. 28

Figure 2-6. Cost break down comparison of modular projects vs. site-built projects (Lawson et al. 2014) .......................................................................................................................................................... 30

Figure 2-7. Drivers and challenges of prefabrication in United States (McGraw Hill Construction 2011).................................................................................................................................................. 33

Figure 2-8. An example of ER information modeling method ........................................................................................................ 36

Figure 2-9. An example of ORM information modeling method ........................................................................................................ 36

Figure 2-10. An example of OO information modeling method ........................................................................................................ 37

Figure 2-11. Information extraction from data stored in IFC file........................................................................................................ 39

Figure 2-12. An example for definition of an object in IFC schema ........................................................................................................ 41

Figure 2-13. An example for definition of a type of relationship in IFC schema ........................................................................................................ 42

Figure 2-14. Defining IfcProperty entity in IFC and ifcXML schemas (Nisbet and Liebich 2007) .................................................................................................................................................. 43

Figure 2-15. Holistic Diagram of the efforts for information exchange standardizations in National BIM Standard – United States .................................................................................................................................................. 46

Figure 2-16. IFC data structure .................................................................................................................................................. 49

Figure 2-17. IFD application in BIM models (Bell et al. 2008) .................................................................................................................................................. 51

Figure 3-1. Primary tasks for Research Objective 1 .................................................................................................................................................. 72

Figure 3-2. Primary tasks for Research Objective 2 .................................................................................................................................................. 73

Figure 3-3. Primary tasks for Research Objective 3 .................................................................................................................................................. 73

Figure 3-4. Visual Representation of the Objectives .................................................................................................................................................. 83
Figure 4-1. Limitations of production and construction industries to address the needs of modular building industry and the way that the PAM can fill the gap................................. 92

Figure 4-2. Integration Layer of the PAM developed for multi-story modular buildings 99

Figure 4-3. Abstract objects defined in the Abstraction Layer of the PAM ................. 100

Figure 4-4. UML notation of the four types of relationships used in the PAM .......... 101

Figure 4-5. A prototype of the object boxes representing different building components in the PAM .......................................................................................................................... 104

Figure 4-6. Workflow of using the LoDs defined in the PAM for defining exchange requirements in modular building projects .......................................................................................................................... 109

Figure 4-7. Information model for the Student Village building developed in the case study for validation of the PAM .......................................................................................................................... 114

Figure 4-8. Breakdown of a volumetric module prototype into its main sub-assemblies ......................................................................................................................................... 116

Figure 4-9. Leveraging the PAM for defining exchange models required for structural design of modular buildings .......................................................................................................................... 119

Figure 5-1. Anatomy of the IDM .................................................................................... 149

Figure 5-2. Design process for a modular building project ............................................. 156

Figure 5-3. Structural design sub-process map for modular buildings ............................. 157

Figure 5-4. Sample Task Unit Description ..................................................................... 160

Figure 5-5. Sample Exchange Model .............................................................................. 162

Figure 5-6. Partial view of the PAM for multi-story modular buildings .......................... 167

Figure 5-7. Sample Exchange Requirement (ER) specification ..................................... 170
Figure 6-1. Direct Translators vs. Open Interoperability Standard ........................................ 187

Figure 6-2. Holistic Diagram of the efforts for information exchange standardizations in National BIM Standard – United States.............................................................. 189

Figure 6-3. Structure of the MVDs developed for conventional constructions............. 200

Figure 6-4. Structure of the MVDs extended for modular buildings.......................... 200

Figure 6-5. MVD Concept Documentation ................................................................. 204

Figure 6-6. Model views of VBL-001, VBL-02, and PSU-001 along with their connections .......................................................... 213

Figure 7-1. Information transformation from Architectural Model to Structural Models ........................................................................................................ 232

Figure 7-2. An example for the IIE........................................................................... 235

Figure 7-3. Relationship among the layers in the presented methodology for design and standardization of IIEs .................................................................................. 237

Figure 7-4. Declaration of an iU for extraction of the live load from space occupancy in the Concept Layer .................................................................................................. 238

Figure 7-5. Generic Layer representation of the iU-01................................................ 241

Figure 7-6. Binding Space Occupancy concept to the IFC schema............................ 243

Figure 7-7. Binding slab live load to the IFC schema ................................................ 244

Figure 7-8. Process of interpreting live load from occupancy information............... 245

Figure 7-9. Building information model of the test structure (top) and structural analytical model of the building generated directly from the IFC file without interpretation (bottom) ........................................................................................................... 254
Figure 7-10. Structural analytical model of the test building interpreted from its building information model ........................................................................................................... 255
Figure 7-11. Process of automating of the IIE ................................................................ 259
Figure 7-12. Execution of the multi-stage interpretations .............................................. 263
Figure 7-13. Multi-stage execution of the eUs to avoid multiple representation transformations ............................................................................................................... 263
Figure 7-14. Current MVD implementation framework (left) vs. the proposed framework (right) .............................................................................................................................. 270
Figure 7-15. The connections between BIM authoring and engineering D/A tools in the proposed framework .................................................................................................................. 271
Figure 8-1. Workflow of the interpretation platform ...................................................... 292
Figure 8-2. An example for the parses IFC object .......................................................... 293
Figure 8-3. Anatomy of IFC instances ............................................................................ 294
Figure 8-4. Class definition for modeling IFC instances ................................................ 296
Figure 8-5. Anatomy of IFC entity blocks ...................................................................... 297
Figure 8-6. Class definition for modeling IFC schema entities ...................................... 298
Figure 8-7. Relative and essential dependents of an instance ........................................ 304
Figure 8-8. Process breakdown of transforming an information unit into IFC instances311
Figure 8-9. Process breakdown of transforming an information unit into IFC instances313
Figure 9-1. Two different representations of an identical model .................................... 327
Figure 9-2. Two different representation of a connection point ........................................ 329
Figure 9-3. Units defined in the coordination model-to-structural D/A model IIE ....... 333
Figure 9-4. Overall workflow of IIE ................................................................. 336
Figure 9-5. Swept solid representation of linear structural element in Structural View. 340
Figure 9-6. Topological representation of linear structural element in Structural View 344
Figure 9-7. Topological representation of planar structural element in e Structural View ......................................................................................................................................... 345
Figure 9-8. The view used for material and section properties in output files observing
IFC 2x3 ........................................................................................................................................................................................................................................ 347
Figure 9-9. The view used for material and section properties in output files observing
IFC 4 ........................................................................................................................................................................................................................................ 349
Figure 9-10. Definition of Merge Point class ...................................................... 351
Figure 9-11. Connectivity adjustment of connecting points .................................. 354
Figure 9-12. Connectivity adjustment at point-to-line connections .................... 356
Figure 9-13. Test Model in Coordination View (Top) and related structural model
generated without interpretation (Bottom).......................................................... 359
Figure 9-14. Structural model interpreted from building information model ....... 360
Figure 10-1. An example of connection of the floor finite element model for an upper
module to the ceiling level of the lower module .................................................. 369
Figure 10-2. Vertical Shear transfer between stacked units: left side) no vertical shear
transfer or partial transfer due to flexibility of connections; right side) coupled at vertical
joints though rigid connections ........................................................................... 370
Figure 10-3. Structural D/A model generation of a module ............................... 372
Figure 10-4. Process map of the Modular Buildings Structural D/A Model Preparation using the proposed iEs ................................................................. 374
Figure 10-5. The iU-01 in Generic Layer ................................................................. 377
Figure 10-6. The iU-02 in Generic Layer ................................................................. 378
Figure 10-7. The iU-03 in Generic Layer ................................................................. 378
Figure 10-8. The iU-04 in Generic Layer ................................................................. 379
Figure 10-9. The iU-05 in Generic Layer ................................................................. 380
Figure 10-10. The iU-06 in Generic Layer ............................................................... 380
Figure 10-11. The iU-07 in Generic Layer ............................................................... 381
Figure 10-12. The iU-08 in Generic Layer ............................................................... 381
Figure 10-13. The iU-09 in Generic Layer ............................................................... 382
Figure 10-14. The iU-10 in Generic Layer .............................................................. 383
Figure 10-15. The iU-11 in Generic Layer .............................................................. 384
Figure 10-16. The iU-12 in Generic Layer .............................................................. 385
Figure 10-17. The iU-13 in Generic Layer .............................................................. 387
Figure 10-18. Building information model (top) and interpreted structural model (bottom) of a module of the building ................................................................. 388
Figure 10-19. Building information model (top) and interpreted structural model (bottom) of the modular building ................................................................. 389
Figure 11-1. Information exchange standards: using the proposed methodology (top) and using the current methodology (down) ......................................................... 402
List of Tables

Table 2-1. Truck transportation limitations in different states of U.S. (Garrison and Tweedie 2008) ................................................................. 22
Table 2-2. Advantages of offsite construction in U.K. (Goodier and Gibb 2005) .......... 32
Table 2-3. Disadvantages of offsite construction in U.K. (Goodier and Gibb 2005) ...... 32
Table 2-4. Drivers and drawbacks of using modular technology in building constructions .................................................................................. 34
Table 2-5. IFC Solution Factory MVDs Projects .................................................. 52
Table 5-1. The modular buildings IDM scope .......................................................... 152
Table 5-2. The code name used for the EMs and TUs tables .................................... 161
Table 5-3. Task Unit (TU) descriptions ................................................................. 163
Table 5-4. Exchange Model (EM) descriptions ...................................................... 164
Table 6-1. List of the IFC concepts developed to support general aspects of modular buildings ........................................................................ 202
Table 6-2. List of the IFC concepts developed to support general aspects of modular buildings ........................................................................ 211
Table 7-1. Connectivity information of a corner connection point in the case study .... 256
Table 7-2. Connectivity information of a middle connection point in the case study .... 257
Table 10-1: Defined Unis for Multi-Story modular buildings in Concept level ......... 376
Table A-1: Task Unit table set ................................................................................ 409
Table A-2: Exchange Model table set............................................................................. 417
Abbreviation List

AEC = Architecture/Engineering/Construction
bSa = buildingSMART alliance
bSI = buildingSMART International
BIM = Building Information Modeling
BPMN = Business Process Map and Notation
CAD = Computer Aided Design
CPM = Core Product Model
DAIM = Design-Analysis Integration Model
DIE = Direct Information Exchange
EM = Exchange Model
ER = Exchange Requirement
FE = Finite Element
FO = Fact-Oriented
GC = General Contractor
HVAC = Heating, Ventilation, and Air Conditioning
IAI = International Alliance for Interoperability
IDM = Information Delivery Manual
IFC = Industry Foundation Class
IFD = International Framework for Dictionary
IIE = Interpreted Information Exchange
ISO = International Standard Organization
LOD = Level of Development
LoD = Levels of Detail
LOT = Linear Object Type
MVD = Model View Definition
MEP = Mechanical/Electrical/Plumbing
NBIMS–US = National Building Information Modeling Standard – United States
OAM = Open Assembly Model
OO = Object Oriented
OOP = Object Oriented Programming
OOT = Opening Object Type
ORM = Object-Role Modeling
PAM = Product Architecture Model
PFEM = Product Family Evolution Model
PLM = Product Lifecycle Management
ROI = Return of Investment
RU = Receiver Unit
SEM = Semantic Exchange Models
SIP = Structurally Insulated Panel
SU = Sender Unit
ToBIM = Tetralogy of BIM
TU = Task Unit
UML = Unified Modeling Language
UoD = Universe of Discourse
XML = Extensible Markup Language
XSD = XML Schema Definition
Acknowledgements

Throughout the journey of my education, I encountered many wonderful individuals who encouraged and supported me in this path. I am greatly indebted to my advisors, Dr. Ali Memari, who offered me new perspectives and gave me the opportunity to find my ideal career path. I appreciate my committee members – Dr. John Messner, Prof. Kevin Parfitt, Dr. Robert Leicht, and Dr. Dinghao Wu – for their contribution, guidance, and support that have greatly improved the quality of this study.

I wish to thank Tom O’Hara from Capsys Corps, John Erb from Deluxe Building System, Don Engel from NRB Inc., John Morrison from Mark Line Industries, and Cliff Cort from Triumph Modular for their review, interviews, arranging site visits, and devoted time during this research. I would like extend my appreciation to Richard See from Digital Alchemy who was of great help by sharing his knowledge and experience in information exchange standardization.

My sincere gratitude to AE faculty members and staff, in particular, AE Department Head Dr. Chimay Anumba. I also want to sincerely thank the Pennsylvania Housing Research Center (PHRC) staff for their support over the past four years. I am also grateful to my peers in the Department of Architectural Engineering and Department of Civil Engineering who have helped me by providing an inspiring and enjoyable atmosphere.
Finally, I thank my family. I would like to thank my dear wife, Rosita, who encouraged me to pursue my PhD studies and stood supportively beside me in this path. Without her invaluable sacrifices and supports, I could not afford the time and resources I needed to complete my PhD study. We are blessed to be expecting our first child, Nikki, to be born in a few months, bringing ultimate happiness to our lives. I would also like to thank my parents, Firouzeh and Ghorban, who dedicated the best years of their lives to support me in my educational journey and persistently encouraging me to continue this path. I appreciate my brother, my best friend, Majid, who was always a source of energy and inspiration throughout my life. I would also like to thank father-in-law, Alireza, my mother-in law, Simin, and my brother-in-law, Sasan, for their kind support and love. I am deeply indebted to my family for my accomplishments.
Dedication

Dedicated to:

My lovely wife, Rosita

My sweet girl, Nikki

My amazing and kind parents, Firouzeh and Ghorban

My supportive brother, Majid
CHAPTER 1: Introduction

1.1. Modular Building Industry

Improvement of technology for multi-story modular construction can change the methods of design, manufacturing, and construction of buildings (Giles and Lara 2006). Popularity of the use of panelized components in commercial and industrial building construction is an evidence for various benefits that exist in shifting from all site-built construction to more panelized and modular construction of residential buildings (MBI 2012, 2013). The term modular is used in the industry for buildings that are constructed using several volumetric (3-Dimensional) modules. A modular building could also have several site-built parts such as shear walls, podiums, or foundations. In Chapter 2, different types of modular buildings are discussed in more details. Initially, application areas of modular buildings were limited to single- or two-story dwellings based on wood-frame construction. Later, availability of alternative construction materials such as light-gage steel for structural load bearing systems in modules made it possible to use this system for low-rise multi-story buildings. During the past decade, by combination of modular systems with site-built supporting structures like shear walls and steel frames, it
became possible to use this technology in mid-rise and high-rise buildings; Figure 1-1, shows some examples of these buildings.


The manufacturing, design, and construction issues become more complicated with increase in number of stories and prevent modular buildings to achieve their real potentials as alternatives for some conventional site-built construction (McGraw Hill Construction 2011). There is a lack of well-developed methods, guidelines, frameworks, and software to facilitate these issues in such multi-story construction. Nonetheless,
interest exists in construction of multi-story low-rise, mid-rise and high-rise modular buildings, in particular in dense urban and metropolitan areas. However, because this technology is relatively new, most applications of multi-story modular construction currently focus on low-rise buildings or in areas with minor natural hazard potential. As an evidence of this new interest, United Kingdom has made significant progress in developing and advancing this technology in construction of multi-story modular buildings (Jellen and Memari 2013). An example for high-rise modular projects in the U.S. is Atlantic Yard B2 Building that is a 32-stories building depicted in Figure 1-2. It will hold 363 market-rate and affordable-rate rental units and 4,000 square feet of retail space (Nonko 2013).

![Figure 1-2. $4.9 billion Atlantic Yards Project (SHoP Architects)](image)

a) Architecture rendering  b) Structural lateral system  c) Modular unit, or block”
The following three attributes are stated by Lawson and Ogden (2008) as major advantages of modular construction of buildings:

a. Lower construction cost—as the number of plant fabrication increases, the constant cost of the plant will be shared by more fabrication, thus reducing the total cost.

b. Shorter construction time—because of reduced jobsite activities;

c. Higher quality production—because of better plant fabrication and inspection.

Modular Building Institute LEED™ report (MBI 2009) mentions the following attributes as major benefits of modular buildings in LEED™ era: safer construction, less material waste, less site disturbance, less material exposure to inclement weather, flexibility, adaptability, and built to code with shorter build times.

A 2011 AEC market report states better meeting of client demands, higher quality, higher productivity, greater Return of Investment (ROI), and greener project as major current drivers for using prefabrication/modularization in construction projects. It has been shown that modularization in construction projects can decrease the overall cost by up to 20 percent and shortens the project time schedule by up to 66 percent (McGraw Hill Construction 2011). Modular building technology is mostly used in four major areas including: offices, education buildings, medical centers, and multi-family and student housing. Figure 1-3 depicts the share of each sector in a pie-diagram (MBI 2011).
1.2. Statement of the Problem and Potential Solution

In prefabrication industry in general, the importance of early stages of the project is more important compared to site-built projects. Figure 1-4, depicts the impacts of implementing prefabrication in different phases of the project (Gibb 1999). The horizontal and vertical axes show the project phases and impacts, respectively. As shown in this figure, the later the project incorporates prefabrication, the higher benefits or positive impacts may be lost from implementation of this system. In fact, after a certain point, it may even have negative impact on the projects. This is also in harmony with the McLeamy Curve depicted in Figure 1-5, which illustrates that making decisions in early stages of the project maximizes positive incomes and minimizes change costs especially
changes in design (AEC Magazine 2013; Smith et al. 2011). Making proper decisions in early stages of projects needs a high level of integration between different project disciplines from owner to architect, design engineers, manufacturer, and contractor. As an integration tool, Building Information Modeling (BIM) could help modular building industry to achieve a more integrated design from early stages of the project to maximize the benefits of using modular building technology. BIM is defined by National BIM Standard – United States® (NBIMS-US™) as “a computable representation of the physical and functional characteristics of a facility and its related project/lifecycle information using open industry standards to inform decision making for realizing better value” (NIBS 2012). Improvement of technology for multi-story modular construction can change the methods of design, manufacturing, and construction of buildings. BIM has significantly affected the construction industry during the past decade. Adoption of BIM within the U.S. construction industry is highlighted by an increase from 28 percent in 2007 to 71 percent in 2012 (McGraw Hill Construction 2012).
Figure 1-4. Prefabrication decision impact at different project stages (Gibb 1999)

Figure 1-5. McLeamy Curve (AEC Magazine 2013)
It is evident that multi-story modular constructions, as a sector, can significantly benefit from and grow upon implementation of BIM. As Nawari (2012) believes, BIM could allow modularization technology to be used in more complex projects. As an evidence, based on the McGraw Hill Construction report (2012), as indicated in Figure 1-6 (a), looking towards BIM usage in North America’s construction projects, steel fabricators/erectors and curtain wall fabricators/installers are among the four most skillful BIM-user contractors. In addition, as Figure 1-6 (b) shows, model driven prefabrication is the second best area for leveraging BIM in contractors’ opinion. Therefore, we can conclude that BIM advantages are in harmony with the nature and characteristics of the prefabricated constructions and can bring about significant benefits if it is implemented in these projects. Incorporation of BIM for design, manufacturing, logistic management, and construction of multi-story modular building projects will significantly increase the information integrity level of the projects and would offer a powerful platform for lifecycle management of buildings.
BIM is for lifecycle information management of buildings. The lifecycle of building projects could be broken down in many phases such as analysis, design, development, test, execution, evaluation, maintenance and disposal (Xiao et al. 2010). For modular buildings we can replace the disposal phase with deconstruction or reuse (Guy et al. 2006). Having an information framework for leveraging BIM in these projects
makes it possible to use the building information model in deconstruction/reuse stage of modular buildings lifecycle. (Lazarus 2005) states that reclaiming, reusing, or recycling materials can save up to 95% of their embodied energy. A potential barrier is labor and equipment costs that can be expensive depending on the complexity and location of the project (Saleh 2009). An advantage that the BIM environment can bring to counteract this barrier is the ability to design multi-story modular buildings for this deconstruction at early stages of the design process. This will improve sustainability of the project, as materials will no longer be taken to a landfill; rather, the dismantled modular units would be taken for storage and future reuse.

Due to advances in BIM, the extent of generated and exchanged data has increased, and this has encouraged firms to take advantage of the value of this tool. An information modeling platform is required to support the interchange and information exchange among different phases to make implementation of Product Lifecycle Management (PLM) concept possible (Sudarsan et al. 2005; Xiao et al. 2010). A standard for organization of this information (taking into the account the amount and its variety) or a common language is required to make the information exchange, analysis and translation possible (OCCS 2013). Fortunately, information technology within the building industry has developed significantly during the past decade through the use of BIM. By extending and customizing BIM tools based on the architecture of modular buildings and using them in this project, we will have a more integrated project, with the advantage of generated information in different project phases, and an increase in the collaboration between different units will result.
1.3. Research Goal and Objectives

The primary goal of this research was to develop an information framework to encourage design and construction of mid- and high-rise buildings using factory-built modular units in order to provide the industry with an option for economical, energy efficient, sustainable, and more affordable construction. As shown in Figure 1-7, the costs of site-built and modular constructions vary in different ranges. Because of the abundance of research, experience, tools, and frameworks available for site-built constructions, the average of such projects is closer to the lower bound. On the other hand, because of the young age and current inefficiencies of the industry, as the figure shows, the lower bound of the project costs in modular building projects is potentially lower than that of the site-built constructions, and the occurrence rate of project costs is higher closer to the upper bound costs. As a result, in spite of numerous benefits of modular constructions, on average, it is slightly more (around 10% – 20 %) expensive in today’s projects compared to site-built projects. With more research leading to further acceptability of these systems, it is expected that the cost occurrence rate of modular buildings will shift closer to the lower bound.
Figure 1-7. Occurrence rate distribution comparison of modular buildings and site-built construction

The rationale for the proposed study is supported by the literature (e.g., Alwisy et al. 2012; Lu and Korman 2010; Nawari 2012) arguing that BIM has the potential to have significant impacts on modular and prefabricated construction compared to site-built projects. To be able to leverage BIM as an information management tool in modular projects, first, the required information about product and project processes needs to be identified as well as the way this information flows among different disciplines along different stages of the project, from early stages of design to maintenance and deconstruction phases of the building life-cycle. Having an information framework for modular building projects along with its supporting infrastructure would promote more use of modular systems for projects, as this leads to improved design, fabrication, and construction processes.
Modular building construction is a market niche that is part of both production and construction industries (Ballard and Howell 1998). It is not as project-dependent as construction, and not as industrialized as production. Therefore, none of the present production and construction techniques and tools can address this industry sector’s needs properly. Currently, interoperable BIM frameworks and platforms are developed to support implementation of BIM in different areas of building construction industry address a significant portion of the requirements at a wide range of building lifecycle. However, there are still important areas that are not supported completely and need special attention. An example effort to address different business sectors’ needs is that by American Institute of Steel Construction (AISC) that is conducting research to address structural steel prefabrication needs (AISC 2015). Modular building sector is another part of these areas. Because of the overlaps between the needs of modular buildings and site-built construction, one portion of this sector’s need as it relates to site-built part is already addressed., The research presented here is to recognize the rest of these requirements and develop a BIM platform to support implementation of BIM in these projects.

A major focus of this research was to work on two of the most fundamental tools that are required to make multi-story modular building more acceptable and easy to design. One main component of the work was to structure an information modeling framework for analysis, design, fabrication, and construction of the structure and major architectural components of modular buildings. The other major thrust of the work was to develop required infrastructure for interoperable information exchanges through different phases of the product’s lifecycle and design of a mechanism for structural design/analysis
modeling of modular buildings. In the following chapters these items are discussed in
detail.

1.4. **Scope and Limitations**

Defining scope and limitation is critical for proper implementation of a
framework. The scope and limitations of this research are specified as follows:

- The focus of this research is more on structural elements of modular buildings and
general architectural elements like sheathings and insulations.

- This research only addresses popular mid-rise and high-rise modular building
systems, although efforts were made to address modular buildings as generic as
possible, which makes the outcomes of this research usable in non-conventional
modular systems. To be more specific, among different types of modular buildings
that are discussed in the first section of Chapter 2, the outcomes of this research
would be able to serve 3D, Hybrid Cored, and Hybrid Podium modular systems. However, other systems can still benefit from the outcomes of this research as
major parts of the outcomes are independent from the type of modularization
system.

- The focus of this research is more on design phase of the project and interactions
with lifecycle phases of the modular buildings.
• Site-built components of modular building systems are discussed at a level of detail that is directly related to modular assemblies. Further details about these components rely upon other frameworks developed for site-built constructions.

1.5. Organization of the Dissertation

This dissertation is in the format of Thesis by Publication (Article Thesis). The first chapter is an introduction to the research and consists of the statement of the problem, research goals and objectives, scope and limitations. Chapter 2 provides a basic literature review on the main areas of this research, while Chapter 3 describes the methodology followed for the study. Six papers form the next six chapters (Chapters 4 to 9). Two of these papers, which are presented in Chapters 4 and 7, are already accepted for publication, respectively, in “ASCE Journal of Construction Management and Engineering” and “ASCE Journal of Computing in Civil Engineering”. The paper that forms Chapter 5 is still under preparation, while the remaining three papers are still in the review process for publication. The first three papers (Chapters 4 to 6) are related to development of an information management framework for multi-story modular buildings. The last three papers (Chapters 7 to 9) represent a new information interpretation mechanism and a supporting infrastructure for its implementation in information exchange related to engineering analysis BIM uses. Chapter 10 discusses
how the concepts discussed in Chapters 7 to 9 could be leveraged along with the
information management framework discussed in Chapters 4 to 6, to automatically
interpret structural analysis models of modular buildings directly from their building
information models. The last chapter (Chapter 11) summarizes the outcome and
contribution of the research reported in this dissertation. This dissertation includes three
appendixes. The first one includes the list of tables describing the Task Units and
Exchange Models discussed in Chapter 5. The second appendix includes exchange
requirement specifications discussed in Chapter 5. The last appendix contains the Model
View Definitions discussed in Chapter 6.

### 1.6. References

(Jan. 5, 2015).


drafting and design for modular construction manufacturing.” *Proceedings of the
Computing in Civil Engineering 2012*, American Society of Civil Engineers, June


MBI (Modular Building Institute). (2009). “Modular Building and the USGBC ’s LEED 
TM,” *Modular Building Institute*.


CHAPTER 2: Literature Review

Although each of the chapters 4-9 in this dissertation has a separate literature review, this chapter provides an overall literature review on modular buildings and information modeling techniques. Chapters 4-9 contain much more detailed review of relevant literature.

2.1. Modular Buildings

Modular building construction is one of the growing construction sectors that has the potential to replace some conventional construction methods with more industrialized technologies (McGraw Hill Construction 2011). In this system, activities are shifted from construction site to controlled factory environment as much as possible. As shown in Figure 2-1, prefabrication makes it possible to run the construction and manufacturing phases in parallel.
2.1.1. Types of Modular Buildings

There are several different types of prefabrication systems being used around the world for construction of multi-story modular buildings (MBI 2011). Each of these systems has its own special features and characteristics and pose their own challenges (Annan et al. 2008). One can categorize modern multi-story prefabricated buildings in the following five major groups based on their construction sequences and components: 3D, Open House, Hybrid Cored-Modular, Hybrid Podium, and Framed Unit systems (Ramaji and Memari 2013). In the following, these systems are briefly discussed.

2.1.1.1. 3D System

In this modular system, the building consists of many 3D (volumetric) modules stacked vertically and attached horizontally to form the building. Each module can be
part of a unit, one complete unit, or even contains more than one small unit (Lawson and Ogden 2010). The size of a module is a function of location of the module in the building, manufacturing and transportation limitations, and available construction equipment. Usually, the floor areas of the modules in this systems are in the range of 100 to 600 square feet (Lawson and Richards 2010). In addition, because of transportation limitation, as a rule of thumb, width, length, and height of modules are usually in the range of 12-16 ft, 40-60 ft, and 10-12 ft, respectively. These ranges along with the most common regulations for truck transportation of prefabricated elements are listed in Table 2-1; some of these ranges and maximum regulations vary in different states (Garrison and Tweedie 2008).

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Width (ft.)</th>
<th>Height (ft.)</th>
<th>Length (ft.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical Dimensions</td>
<td>12-16</td>
<td>10-12</td>
<td>40-60</td>
</tr>
<tr>
<td>Limitations with permit</td>
<td>12</td>
<td>14</td>
<td>80</td>
</tr>
<tr>
<td>Limitations without permit</td>
<td>16</td>
<td>16</td>
<td>120</td>
</tr>
</tbody>
</table>

Based on the load paths in this type of structure, 3D system can be classified in three main categories including “Wall-load bearing”, “Post-supported”, and “Open-
ended” systems (Lawson and Richards 2010). These three systems are described in the following.

**Wall-load bearing systems:** In wall-load bearing systems, gravity loads (live and dead) are carried by walls that transfer loads to the lower structure or foundation (Lawson et al. 2005).

**Post-supported systems:** In this system, the module’s corner posts (column) support the gravity loads transferred to such posts by module’s edge beams. Some interior posts within the walls may be used in this system to share resisting the gravity loads and help the corner posts to transfer the loads to the foundation or the sub-structure (SteelConstruction.info 2015).

**Open-Ended systems:** Modules in this system are made up of a series of parallel moment frames with a constant spacing. The two ends of the module in the longer direction are open and this provides enough space for installation of full height glazing panels or openings (Lawson et al. 2014).

### 2.1.1.2. The Open House System

The Open House system may consist of various combinations of framing and modular systems and is developed to enhance the flexibility of space planning in modular buildings. This system is made of two frames. One of these frames is the interior structure of the modules, the second being the exterior frame, which consist of columns
on exterior edge of the building and in the middle rows where adjacent modules meet. The construction procedure of the Open House system has three main stages. First, the columns of the external frames are erected at the job-site. Next, the modules are installed between the columns erected in the first stage. At the end of this stage, the columns are positioned in the hollow spaces within the modules. In the last stage, by installation of a uniform floor system, the internal modules frames and external frames will be integrated together to form a uniform structural system, as shown in Figure 2-2 (Birgersson 2004; Lawson and Ogden 2008).

Figure 2-2. Main components of the Open House system
2.1.1.3. **Hybrid Cored-Modular Systems (Cluster)**

As the height and number of stories of a building increase, the magnitude of the total gravity and lateral loads increase, and therefore, the size of load-bearing elements in lower modules needs to be larger beyond what is needed just for gravity loads. Beyond a certain point, the size of these elements and the weight of the individual modules becomes unreasonably high; while generally, due to transportation, erection, and installation difficulties, prefabricated modules should be kept light. In cluster systems, the problem is solved by considering staircases and elevator cores as structural units that resist the lateral loads and limit the size of internal elements of the modules. An example for this system is the Student Village building, which is a 25-story building located in Wolver Hampton City of U.K, where the stacked modules rely on a concrete core to resist lateral loads (Lawson et al. 2012; MBI 2010). A schematic sketch of this system is shown on the left hand side of Figure 2-3.

2.1.1.4. **Hybrid Podium-Modular Systems**

One of the limitations of modular buildings is short bay spans in lower stories that is normally desired for retail or parking spaces. The hybrid podium-modular system is a solution in modular projects that needs longer bay spans in lower stories (Lawson and Richards 2010). In podium-modular systems, below-ground and some of the bottom stories (usually one or two stories) are built using conventional structural steel or
concrete frames with long spans. Then, modular parts of the building are installed on top of this site-built structure. In other words, the podium is like an above-ground foundation for modular parts where the modules transfer their loads to podium beams. Figure 2-3 shows a schematic sketch of this system on the right hand side. As shown in this figure, usually module edges are aligned with podium beam centerlines to minimize unnecessary loading eccentricity on podium beams and columns.

Figure 2-3. Schematic illustration of the hybrid cored-modular system (Left) and Hybrid podium-modular system (Right)

2.1.1.5. **Framed Unit Systems**

In this system, the main load resisting structure of the building is not constructed as modular; rather, as depicted in Figure 2-4, it is a conventional structural frame. After erection of the main structural skeleton, prefabricated modules are placed and fitted
between beams and columns of the structural frame. One of the most important advantages of this system is that the module properties do not depend on the plan shape, number of stories, and lateral loads. Since the main structural frame bears the loads, all of the units from different stories could be identical. In addition, module elements are designed just to carry their own gravity and construction loads; therefore, the modules in this system are lighter compared to the other 3D system types.

![Figure 2-4. Schematic illustration of construction of a building with Framed Unit system](image)

2.1.2. Economic Model

As shown in Figure 2-5 (a), based on the research done by Fawcett et al. (2005), about 75 percent of the labor and material used in modular building projects are related to off-site activities. In addition, among different scales of prefabrication, on-site activities are mostly minimized in volumetric modular systems. Furthermore, as depicted in Figure 2-5 (b), volumetric modularization can financially benefit from prefabrication
more than other modularized systems. This benefit is estimated to be up to 90 € per square meter (almost $15 per square feet) and is calculated considering earlier rent, shorter borrowing period, less snagging, and less on-site inspection (Fawcett et al. 2005). For example, in Student Village project in U.K. that is a 25-story student dormitory, using volumetric prefabrication, construction completed about six months earlier and the client was able to rent the units one academic year earlier (MBI 2010).

(a) Distribution of labors and materials                        (b) Financial benefits

![Figure 2-5. Comparing modularizations in different scales](Courtesy of National Audit Office (NAO))

Based on the National Audit Office (NAO) report (Fawcett et al. 2005), in addition to the time saving in modular building projects, construction cost reduces 10% because of benefits of prefabrication.
Although time, labor, and material savings in modular building construction reduce the projects costs, but like any other industry, there are many hidden costs associated with these projects. Some examples are: (a) Overhead related to preparing a manufacturing plant including equipment, employees, maintenance, utility, etc. (b) Anticipated profit for the offsite fabricator as a business enterprise, (c) Transportation cost that greatly depends on transportation route and size of modules, (d) Module installation cost that needs skilled labor and special equipment, (e) Design fees that are usually higher compared to conventional construction due to more coordination required with other disciplined (Ludeman 2008; Smith 2011).

In Figure 2-6, the cost breakdown of modular building projects is shown and compared with that of site-built projects (Fawcett et al. 2005; Lawson et al. 2014). As shown in this figure, material waste in modular projects could be reduced by 33%. In addition, site overhead and personnel costs may be decreased by 50%. Based on the National Audit Office (NAO) report (Fawcett et al. 2005), even in highly prefabricated projects, about 30% of the costs are site-costs. These costs could be broken down as follows: Foundations (4%), Services (7%), Cladding (13%) and Finishing, etc. (6%) (Lawson and Ogden 2010). Although design and production costs decrease with the increase of the number of modules manufactured in a factory, based on this report, the production costs may be increased up to 10% for design and management in a typical modular project.
2.1.3. Drivers and Drawbacks

Modular construction has many advantages along with some disadvantages when compared to site-built construction. That is why it cannot be an absolute replacement for the site-built constructions. Understanding the pros and cons of this technology would be very helpful in proper and right-in-time decision making process and to see the appropriateness of modularization as a solution for a given specific project. Therefore, many efforts have been expended around the world to identify drivers and drawbacks of using modular technology in building constructions.
Lu (2009) surveyed 600 design firms (A/E) and 11,000 general contractors (GC) in the United States. The results indicated the following attributes as major benefits of modularization of buildings: shorter project time, shorter construction schedule, reduction of the need for skilled craft workers, greener construction, lower construction cost, and higher quality. Transportation restraints, inability to make changes onsite, limited design options, and lack of experience among stakeholder indicated to be major drawbacks of this type of construction (Lu 2009).

The multi-story modular building industry has started to grow in Europe, especially in the U.K., at a faster pace compared to the United States. Goodier and Gibb (2005) surveyed 75 construction organization, including 39 clients and designers, 13 contractors, and 23 offsite suppliers and manufacturers. As shown in Table 2-2, shorter construction time, higher quality, more consistency product, and lower snagging and defects are the four top advantages of prefabricated buildings. Furthermore, as shown in Table 2-3, higher project cost, longer lead-in time, client resistance, and lack of guidance and information are the top major barriers of the adopting prefabrication technologies in buildings (Goodier and Gibb 2005).
Table 2-2. Advantages of offsite construction in U.K. (Goodier and Gibb 2005)

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Clients/designers</th>
<th>Contractors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% of respondents</td>
<td>% as 1st choice</td>
</tr>
<tr>
<td>Increased quality</td>
<td>79</td>
<td>28</td>
</tr>
<tr>
<td>More consistent product</td>
<td>77</td>
<td>18</td>
</tr>
<tr>
<td>Reduced snagging &amp; defects</td>
<td>79</td>
<td>8</td>
</tr>
<tr>
<td>Increased value</td>
<td>51</td>
<td>5</td>
</tr>
<tr>
<td>Increased sustainability</td>
<td>49</td>
<td>3</td>
</tr>
<tr>
<td>Reduced initial cost</td>
<td>44</td>
<td>3</td>
</tr>
<tr>
<td>Reduced whole life cost</td>
<td>41</td>
<td>0</td>
</tr>
<tr>
<td>Increased flexibility</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>Greater customisation options</td>
<td>33</td>
<td>3</td>
</tr>
<tr>
<td>Increased component life</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td>Other</td>
<td>18</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 2-3. Disadvantages of offsite construction in U.K. (Goodier and Gibb 2005)

<table>
<thead>
<tr>
<th>Barriers</th>
<th>Clients/designers</th>
<th>Contractors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% of respondents</td>
<td>% as 1st choice</td>
</tr>
<tr>
<td>More expensive</td>
<td>67</td>
<td>54</td>
</tr>
<tr>
<td>Longer lead-in times</td>
<td>46</td>
<td>8</td>
</tr>
<tr>
<td>Client resistance</td>
<td>38</td>
<td>13</td>
</tr>
<tr>
<td>Lack of guidance and information</td>
<td>33</td>
<td>5</td>
</tr>
<tr>
<td>Increased risk</td>
<td>36</td>
<td>0</td>
</tr>
<tr>
<td>Little codes &amp; standards available</td>
<td>33</td>
<td>3</td>
</tr>
<tr>
<td>Other</td>
<td>31</td>
<td>18</td>
</tr>
<tr>
<td>Negative image</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td>Not locally available</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>No personal experience of use</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>Obtaining finance</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>Insufficient worker skills</td>
<td>21</td>
<td>0</td>
</tr>
<tr>
<td>Reduced quality</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>Restrictive regulations</td>
<td>13</td>
<td>0</td>
</tr>
</tbody>
</table>

McGraw Hill Construction (2011) researched the whole building prefabrication industry of the United States. Based on this research, as shown in Figure 2-7(a), lower
construction cost, shorter project time, better meeting of owner demands, move competitive market advantages, and higher quality are the major driving factors for using prefabrication in building construction. In addition, as shown in Figure 2-7(b), the following factors are identified as the major current reasons for not using prefabrication in some projects: lack of experience in architectural firms in designing for prefabrication, incapability of in some types of projects, lack of interest by clients in modular construction, and availability of the local prefabrication shop (McGraw Hill Construction 2011).

(a) Drivers

(b) Challenges

Figure 2-7. Drivers and challenges of prefabrication in United States (McGraw Hill Construction 2011)
The drivers and drawbacks identified in these three studies are summarized and listed in Table 2-4.

Table 2-4. Drivers and drawbacks of using modular technology in building constructions

<table>
<thead>
<tr>
<th></th>
<th>Drivers</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lu 2009</strong></td>
<td>Shorter project time</td>
<td>Transportation restraints</td>
</tr>
<tr>
<td></td>
<td>Shorter construction schedule</td>
<td>Inability to make changes onsite</td>
</tr>
<tr>
<td></td>
<td>Less need for skilled workers</td>
<td>Limited design options</td>
</tr>
<tr>
<td></td>
<td>Greener construction</td>
<td>Experience among stakeholder</td>
</tr>
<tr>
<td><strong>Goodier and Gibb 2005</strong></td>
<td>Shorter construction time</td>
<td>Higher project cost</td>
</tr>
<tr>
<td></td>
<td>Higher quality</td>
<td>Longer lead-in time</td>
</tr>
<tr>
<td></td>
<td>More consistence product</td>
<td>Client resistance</td>
</tr>
<tr>
<td></td>
<td>Lower snagging and defects</td>
<td>Lack of guidance and information</td>
</tr>
<tr>
<td><strong>McGraw Hill Construction 2011</strong></td>
<td>Lower construction cost</td>
<td>Lack of experience in architectural firms</td>
</tr>
<tr>
<td></td>
<td>Shorter project</td>
<td>Prefabrication is not applicable</td>
</tr>
<tr>
<td></td>
<td>Better meeting of owner demands</td>
<td>Clients are not interested prefabrication</td>
</tr>
<tr>
<td></td>
<td>Having competitive market advantages</td>
<td>Local prefabrication shop is not available</td>
</tr>
</tbody>
</table>
2.2. Information Modeling

Database management systems are widely used in information-oriented industries to increase productivity. To effectively manage the database, it should be effectively designed using an information model (Halpin and Morgan 2010). Lee (1999a) defines information as a representation of concepts, relationships, rules, constrains, and operations to specify information semantics for a specific Universe of Discourse (UoD). UoD is defined by Halpin and Morgan (2010) as the application area from which the information model is being developed. Information modeling is a technique for specifying data requirements within the application domain for using a shared database (Lee 1999b).

Three major types of information modeling methodologies have widely been used in information technology (IT) industry during past decades: Entity-relationship (ER), Object-Role Modeling (ORM), and Object Oriented (OO) (Balsters et al. 2006). These methodologies are described briefly in the following:

ER was introduced in 1976 by Peter Chen and is still the most widely used information modeling approach (Chen 1976; Halpin and Morgan 2010). ER focuses on the concept of entities and their relationships that could be applied to describe information requirements (Lee 1999b). During the past four decades, several versions of ER have been presented, and there is still no standard notation for this methodology. Figure 2-8 shows an example of this method of information modeling.
ORM is the only type of Fact-Oriented (FO) Method that is widely adopted in the industry. This method introduced in the early 1970s, and as a semantic information modeling method views all facts as objects playing a role or being part of a relationship. An example of this methodology is shown in Figure 2-9. Unlike to the ER method, in the ORM method the attributes are not included in the base model. Although this method results in a larger diagram, this attribute-free approach has many advantages such as simplicity, stability, and ease of validation (Halpin 2005, 2006).
In the OO approach, either data or behavior are nested within the objects. The OO approach focuses on identifying objects from the application domain and then captures the operations and functions. This method is mainly used in programs written with object-oriented languages and could be used for database design. By representing concepts in terms of objects and roles, the OO approach presents a powerful method for conceptual modeling. Unified Modeling Language (UML) that is an extended version of ER is the most influential type of OO information modeling that does not yet have a standard notation. UML includes class diagrams that could be used for specifying static data structure from operations to low-level design decisions (Halpin and Morgan 2010; Lee 1999b; Rumbaugh et al. 1991). An example of a diagram using UML is shown in Figure 2-10.

![Diagram of OO information modeling method](image)

Figure 2-10. An example of OO information modeling method
With improvements in the manufacturing industry, products have become more complicated and the need for an organized product information model has increased. Product modeling is a key element for implementation of advanced technologies in manufacturing (Zhao and Liu 2008). As products become more complicated, more experts and engineers need to be involved in projects and the need for a resilient and robust way of information exchanges has increased. In the 1980s, International Standard Organization (ISO) started an effort to address this need by developing a standardized product data model supplemented with a standard for the exchange of the product model data (STEP) (Spiby 1992). This product information model is developed in Express language, which is a combination of ER and OO information modeling approaches (Pratt 2001; Zhao and Liu 2008). By introducing STEP in 1992, it became the main approach for information modeling in a wide range of industries such as mechanical and electrical systems, ship building process, and furniture manufacturing to name a few (Pratt 2005).

In 1994, International Alliance for Interoperability (IAI), later changed the name to buildingSmart International, started to use STEP and express language to develop a standard non-proprietary information model for the whole lifecycle of building construction (Spiby 1992). The result of this effort is the Industry Foundation Classes (IFC) data schema. Data models developed using Express language could be represented in either plain textual format or a graphical representation called Express-G, although the textual format is richer and some information may be lost in graphical representation.
All definitions of the information model developed for building industry are written in IFC schema file using Express language. According to Eastman (1999), schema is the unit of definition that defines UoD and declares objects associated with its dependency and purposes. IFC files use the schema and instantiate information related to the objects in the project. So, schema is like a repository of objects for representation of a building information in IFC files. In other words, as shown in Figure 2-11 IFC file represents project-specific information in the data structure defined in the IFC schema, which along the files are stored in text-based file formats using “exp” and “ifc” file extensions. IFC schema gives meaning to the IFC file, without which IFC will be some meaningless lines of text.

![Figure 2-11. Information extraction from data stored in IFC file](image)

There are two types of entities defined in the IFC schema: Objects and Links. Objects can be either physical or non-physical object properties or components used in the building industry such as beam, wall, project’s owner, time, length. Since Express is
not a fully object-oriented language, STEP lacks links that connect objects together. This problem is solved in Express by defining some entities that play the role of links. Links (also known as Relations) are the entities that connect object entities together. An example is IfcRelAggregates that is used for defining an object composed of many other Objects such as Ramp that is made of railings, flight, and slab. Another example is IfcRelAssigns that is used for assigning a two Objects together like assigning a client to a project.

In an entity definition block in the IFC schema, five important types of definitions could be found including super-type, sub-types, attributes, and rules. As mentioned before, STEP is using the object-oriented concept. Therefore, each entity could be a sub-type of another entity, and in this case inherits all attributes of its super-type. At the beginning of the block, all sub-types and super-types of the entity are declared, if there is any. Afterward, immediately after sub-types and super-type declarations, specific attributes of the entity that are not inherited from its super-type are defined. The attributes of an entity in STEP could be classified in three categories including those introduced in the following.

- Explicit attributes are information units that characterize the properties of the entity. An explicit attribute could be a single value like an integer number, a set of values with a same type, or another Concept entity value.
- Derived attributes are not loaded as data like Explicit attributes, but derived from explicit attributes of that entity.
• Inverse attributes are like connectors that could be used to relate two or more Concepts entities together using Link entities.

The last part in the block is rule definition that are used in the schema to define constraints. Rules are defined in form of equations and are shown after the “WHERE” line in the entity definition block. The value of the rules could be true, which means it is following the rule; false, that means it is violating the constrain; and unknown, which means some attributes are missing.

In Figure 2-12, the definition of IfcWall is shown in the schema as an example for Object entities. As shown in this figure, the IfcWall is the super-type of IfcWallElementedCase and IfcWallStandardCase, while it is the subtype of IfcBuildingElement, and has a non-inherited attribute named PredefinedType, which is constrained by two rules named CorrectPredefinedType and CorrectTypeAssigned.

Figure 2-12. An example for definition of an object in IFC schema
In addition, the definition of IfcRelAggregates in the schema is shown in Figure 2-13 as an example for Link entities. As shown in this figure, it does not have any sub-types, it is a subtype of IfcRelDecomposes, it has two non-inherited attributes named RelatingObject and RelatedObjects, and it is restricted by one rule named NoSelfReference.

```xml
ENTITY IfcRelAggregates
    SUBTYPE OF (IfcRelDecomposes);
    RelatingObject : IfcObjectDefinition;
    RelatedObjects : SET [1:?] OF IfcObjectDefinition;
    WHERE
      NoSelfReference : SIZEOF(QBQ(Temp <: RelatedObjects | RelatingObject :=: Temp}) = 0;
END ENTITY;
```

```xml
ENTITY IfcRelAssigns
    ABSTRACT SUBTYPE OF (OMNCF)
      IfcRelAssignsToActor
      IfcRelAssignsToControl
      IfcRelAssignsToGroup
      IfcRelAssignsToProcess
```

Figure 2-13. An example for definition of a type of relationship in IFC schema

ifcXML is an implementation of ISO-10303 Part 28 Edition 2 standard that provides an automatic conversion from the EXPRESS language defined in ISO 10303 part 1 to XML language. Extensible Markup Language (XML) is a simple and very flexible text-based markup language that is developed to provide a digital format that is both machine and human readable format. Nowadays, XML is playing an increasingly
important role web-based data exchanges. By mapping IFC content to XML format using XSD language, construction industry can benefit from a widely globally accepted system that is tested in numerous industry areas. XML Schema Definition (XSD) is a schema definition language used for developing information models in XML format. Although ifcXML structure is not as rich as the IFC data format, it is very useful for data extraction, model merging and information transmission such as schedules, quantity take-offs, Requests-For-Information (RFI)'s, and ecommerce communications (Nisbet and Liebich 2007). A comparison of IFC and ifcXML schemas is shown as an example in Figure 2-14.

<table>
<thead>
<tr>
<th>Original IFC EXPRESS definition</th>
<th>Definition in the ifcXML schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTITY IfcProperty</td>
<td>&lt;xs:element name=&quot;IfcProperty&quot;</td>
</tr>
<tr>
<td>ABSTRACT SUPERTYPE OF (ONEOF</td>
<td>type=&quot;ifc:IfcProperty&quot; abstract=&quot;true&quot;</td>
</tr>
<tr>
<td>(IfcComplexProperty</td>
<td>substitutionGroup=&quot;ex:Entity&quot; nillable=&quot;true&quot;/&gt;</td>
</tr>
<tr>
<td>,IfcSimpleProperty));</td>
<td>&lt;xs:complexType name=&quot;IfcProperty&quot;</td>
</tr>
<tr>
<td>Name : IfcIdentifier;</td>
<td>abstract=&quot;true&quot;&gt;</td>
</tr>
<tr>
<td>Description : OPTIONAL IfcText;</td>
<td>&lt;xs:complexType name=&quot;IfcProperty&quot;</td>
</tr>
<tr>
<td>END_ENTITY;</td>
<td>abstract=&quot;true&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:complexContent</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:extension base=&quot;ex:Entity&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;Name&quot; type=&quot;ifc:IfcIdentifier&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;Description&quot; type=&quot;ifc:IfcText&quot; nillable=&quot;true&quot; minoccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:sequence&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:extension&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:complexContent&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:complexType&gt;</td>
</tr>
</tbody>
</table>

Figure 2-14. Defining IfcProperty entity in IFC and ifcXML schemas (Nisbet and Liebich 2007)
2.3. Information Exchange Standardization

To be able to exchange information between two different software applications, the way that information is structured should be standardized. A standard in construction is defined by Vries (2006) as an approved specification of a limited set of solutions to actual or potential matching problems, developed to benefit the party or parties involved, balance their needs, and be used repeatedly or continuously during a certain period of time by a substantial number of target parties. Open standard is a non-proprietary, publicly available standard. Open standard for interoperability has many advantages compared to direct translation. One advantage is decreasing the number of required translators. With an open standard available, it is not required to develop a different translator between every two individual units; we just have to develop a single translator between the open standard and each unit. Eliminating the need of having access to proprietary file formats, handling software changes, and minimizing errors in translation by testing are other advantages of using an open standard format compared to direct translation (Bloor and Owen 1995; Gielingh 2008; Laakso and Kiviniemi 2012). In addition to literature review included in this section, Chapters 5 and 6 provide more information related to information exchange standards.
2.3.1. **International BIM Standards**

buildingSMART International (bSI) (formerly named International Alliance for Interoperability (IAI)) initiated an effort in 1994 to standardize ontology-based semantic and information exchanges to support construction industry (Nawari and Sgambelluri 2010). The intention of bSI is achieving an improved construction process from planning, to design, to construction, to operation, and to maintenance process using a standardized machine-readable information model for different types of facilities, new or old, which contains all appropriate information, created or collected, in a format useable throughout its lifecycle by all (NIBS 2012).

As depicted in Figure 2-15, information exchange standardization efforts in bSI, are focused on three major areas: 1) Information Delivery Manual (IDM), 2) International Framework for Dictionary (IFD), and 3) Industry Foundation Class (IFC) data schema. IDM is a standard for the construction projects processes and information requirements, IFD is a data dictionary and a standard for the terminology and data model for the information related to construction projects, and IFC is a standard data schema for data management and information exchanges in construction industry. At the conjunction of IDM and IFC, Model View Definitions (MVDs) are defined. MVDs are translation of the exchange requirements defined in IDM, to IFC schema language. In the following sections, each of these parts is defined in more detail.
2.3.1.1. **Information Delivery Manual/Model View Definition**

IDMs and MVDs are to specify the information Exchange Requirements (ERs) and standardize the way such information would have to be exchanged using IFC file format. They explain the exchange scenario in a human readable format, as well as in a computer interpretable way for software vendors to adopt and implement the standard (NIBS 2012). IDM and MVDs support the integrated construction processes by serving the technical implementation needs of the software vendors and provide role—based process workflow for the end user (Laakso and Kiviniemi 2012).
An IDM identifies and documents information requirements and data exchanges in construction projects. These documents have to be expressed in human-readable form (Nawari 2012). IDM is an integrated reference for processes and BIM data requirements in a specific set of activities in certain types of projects. IDM specifies where a process fits, why it is relevant, who creates and consumes the information, what the information is, and how the software vendors should support the exchanges of this information (Eastman et al. 2010; Laakso and Kiviniemi 2012; Wix and Karlshojej 2010).

MVDs integrate and standardize the ERs coming from one or many IDM processes to the most logical Model Views that will be supported by software tools. Implementation of these components specifies structure and format for data to be exchanged using a specific version of the IFC data schema. In other words, it standardizes the way that the information for a certain information exchange has to be organized, and then helps to show how the information has to be digitally modeled and exchanged using the IFC data schema (Nawari and Sgambelluri 2010; NIBS 2012).

### 2.3.1.2. Industry Foundation Class

The IFC schema was developed by International Alliance for Interoperability (IAI) to address the interoperability problems in building construction industry. Now, it is widely adopted by the industry and software vendors, and it is the standard file format of NBIMS-US. IFC is a format for representation of building objects, their attributes,
relationships, and inheritances (Laakso and Kiviniemi 2012; Nawari and Sgambelluri 2010). Using IFC as the open standard file format for information management and exchanges can guarantee sustainable information modeling and seamless information exchanges in construction projects.

Using a layered model, the IFC schema takes advantage of both structuralist and minimalistic approaches by (Tarandi 1998). As depicted in Figure 2-16, the structure of the IFC schema is divided in four layers, including: domain, interoperability, core, and resource. These layers follow a restrictive hierarchy and each layer is independent from all upper-level layers. The resource layer consists of the Resource schema that is a set of basic definitions such as geometry, measurement, cost, date and time, quantity that could be used for describing objects in higher layers. The core layer includes the Kernel and Extension Modules and contains abstract concepts that are used for entity definitions in higher layers. The Kernel determines the model structure and decomposition, providing basic concepts regarding objects, group, relationships, type definitions, attributes and roles. Core extensions, including process, product, and control extensions, are specializations of classes defined in the Kernel. Space, site, building, building elements, and annotations are some examples for abstract concepts defined in core extensions. The interoperability layer holds building entities such as beam, column, window, occupant, and flow segments that are shared between multiple construction applications. This layer provides the interface for domain models, thus providing an exchange mechanism for enabling interoperability across domains. The domain layer is the highest level of IFC data structure and contains entities for different AEC domains or types of applications,
such as architecture, structural engineering, and HVAC (Heating, Ventilation, and Air Conditioning), among others. Footing, pile, plate, chiller, and boiler are examples of entities included in this layer (Laakso and Kiviniemi 2012; Liebich and See 1999; Liebich and Wix 2000; Wix and See 1999).

![IFC data structure](image)

**Figure 2-16. IFC data structure**

### 2.3.1.3. International Framework Dictionary

For any free information flow, three requirements have to be addressed: a format for information exchanges, a planned process model, and a standardized description of what information you exchange actually is. buildingSMART International has addressed the last requirement by developing the International Framework Dictionary (IFD) library, which in simple term is a standard for a terminology dictionary (Bell et al. 2008; NIBS 2012).
IFD is an open library that defines concepts and terms semantically and makes it possible to assign a Globally Unique ID (GUID) to each information piece in the IFC format. As a result, an exact discretion of a component can be correctly extracted by using a proper tool, as long as the correct GUID is given. For example, the project architect can define and describe a column in a language other than English, and then the structural engineer in the United States will be able to understand and extract the exact description of that column. While plain textual based information like names and descriptions are exchanged between project actors, the related GUIDs are used by computers to extract the required information. IFD provides a mechanism to develop a dictionary to connect building information from existing database to a standard data model (Bell et al. 2008; Laakso and Kiviniemi 2012; NIBS 2012).

Contents within the Data Dictionary can be divided in two different categories: 1) Subjects (Term): concepts that could be represented by a name, and be distinguished and recognized from other concepts, and 2) Characteristics (Properties): concepts that their meaning cannot be defined using other concepts but through a description. Subjects are concepts that are defined, and characteristics are defining concepts. Door, window, column, door frame are some examples for Subject concepts. Characteristic concepts include properties like behavior, environmental influence, function, measure, and unit. Figure 2-17 illustrates how a subject (door) can be defined as an aggregated object and be described using different characteristics (properties) (Bell et al. 2008; NIBS 2012).
2.3.2. Major Efforts Related To Structural Aspects

Many efforts have been made so far to extend the NBIMS-US to address interoperability issues in different areas. Since practical implementation of the developed information exchange standards is the responsibility of software developers, the Building Lifecycle Interoperable Software (BLIS) Group was founded in 1999 to fill the gap between publication of the standard and its implementation in software. BLIS introduced MVDs in 2006 as an official element for interoperable information modeling standardizations to show how data should be exchanged between different types of applications; and by this means benefits the implementers of IFC software (Laakso and Kiviniemi 2012). This group evaluates MVDs and IFC concepts proposed for different purposes of information exchanges and add it to the repository if accepted. Table 2-5
summarizes some of the efforts made for MVDs development. Not all of these projects are completed or evaluated by BLIS; the status column in this table shows the status of each effort (BLIS 2015) that could be Idea, Draft, Proposal, Candidate, or Official, respectively. For more clarification, two examples of these efforts are discussed in more detail in the following. In addition, the status of standard available for structural design to structural analysis information exchanges are discussed in Chapter 9 in detail.

Table 2-5. IFC Solution Factory MVDs Projects

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Reference No.</th>
<th>Name</th>
<th>Status</th>
<th>Reference No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic HandOver to Facility Management</td>
<td>Draft</td>
<td>GSC-001</td>
<td>Extensibility</td>
<td>Idea</td>
<td>VBL-003</td>
</tr>
<tr>
<td>Architectural Design to Building Energy Analysis</td>
<td>Candidate</td>
<td>GSA-003</td>
<td>Indoor climate simulation to HVAC design</td>
<td>Proposal</td>
<td>HUT_HVAC-001</td>
</tr>
<tr>
<td>Architectural Design to Circulation/Security Analysis</td>
<td>Proposal</td>
<td>GSA-002</td>
<td>Landscape design to road design</td>
<td>Idea</td>
<td>CRC_CI-002</td>
</tr>
<tr>
<td>Arch. Design to Quantity Takeoff for Cost Estimating</td>
<td>Candidate</td>
<td>GSA-004</td>
<td>Masonry Structural Design to Structural Analysis</td>
<td>Draft</td>
<td>LIF_DCP-001</td>
</tr>
<tr>
<td>Architectural Design to Spatial Program Validation</td>
<td>Candidate</td>
<td>GSA-001</td>
<td>Precast Concrete Exchanges</td>
<td>Candidate</td>
<td>PCI-001</td>
</tr>
<tr>
<td>Concept Design BIM 2010</td>
<td>Official</td>
<td>GSA-005</td>
<td>Modular Bridge Design to Structural Design</td>
<td>Draft</td>
<td>PSI-001</td>
</tr>
<tr>
<td>Design to Code Compliance Checking [ICC 2006]</td>
<td>Proposal</td>
<td>ICC-001</td>
<td>Road design to landscape design</td>
<td>Idea</td>
<td>CRC_CI-001</td>
</tr>
<tr>
<td>Early Concept Design to Analysis</td>
<td>Draft</td>
<td>GSA-006</td>
<td>Space Req. and Targets to Thermal Simulation</td>
<td>Draft</td>
<td>HUT_HVAC-002</td>
</tr>
<tr>
<td>Nordic Energy Analysis (subset of CDB-2010)</td>
<td>Candidate</td>
<td>NOW-001</td>
<td>Structural design to structural analysis</td>
<td>Proposal</td>
<td>VBL-001</td>
</tr>
<tr>
<td>Architectural design to landscape design</td>
<td>Idea</td>
<td>CRC_CI-003</td>
<td>Structural Design to Structural Detailing (ATC-75)</td>
<td>Draft</td>
<td>ATC-001</td>
</tr>
<tr>
<td>Architectural design to structural design</td>
<td>Draft</td>
<td>VBL-002</td>
<td>Wood Structural Design to Structural Analysis</td>
<td>Draft</td>
<td>LIF_DCP-002</td>
</tr>
<tr>
<td>Architectural design to thermal simulation</td>
<td>Proposal</td>
<td>VBL-007</td>
<td>Architectural design to quantity takeoff - level 1</td>
<td>Idea</td>
<td>VBL-004</td>
</tr>
<tr>
<td>Architectural Programming to Architectural Design</td>
<td>Draft</td>
<td>BSI-001</td>
<td>Architectural design to quantity takeoff - level 2</td>
<td>Draft</td>
<td>GSC-002</td>
</tr>
<tr>
<td>Curtain Wall Design to Energy Analysis</td>
<td>Draft</td>
<td>UNSW-001</td>
<td>Architectural design to quantity takeoff - level 3</td>
<td>Idea</td>
<td>VBL-006</td>
</tr>
<tr>
<td>Extended Coordination View</td>
<td>Idea</td>
<td>ISG-001</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.3.2.1. Structural Design to Structural Detailing (ATC-75)

This project was carried out by the Applied Technology Council (ATC) to address the interoperability issue of structural elements information exchanges between structural D/A model and structural BIM model. A structural model may interact with four types of models including: 1) the architectural BIM model, 2) the structural D/A model, 3)
detailing model, and 4) the construction model. The NBIMS-US methodology, including development of IDM and MVD, was used in this project for standardization of information exchanges (Dean 2010). In the first step, the structural engineering business process map was developed. The whole structural design processes is divided in four stages consisting of: 1) defining the structural systems, 2) structural model development, 3) performing structural analysis for verification, and 4) extracting structural drawings, specifications, and shop drawings. Afterward, the ERs were recognized for each information exchange and by binding those to the IFC data schema, the MVD was developed (ATC 2013).

A benchmarking test was carried out in this project on a variety of BIM and structural D/A software to systematically quantify the state of interoperability and success of information transfer in today’s available BIM and structural tools. The focus of these tests is in the following areas: 1) geometric coordinate transfer, 2) material properties transfer, 3) curved and shaped geometric transfer, and 4) geometry transfer of sloped members. An identical simple model that covers all four focused areas was developed for the benchmarking tests. One benchmarking test was done before developing the IDM and MVD, and one was done after. The software vendors implemented the developed MVDs by modifying their software based on feedback from the project and the results of the first benchmarking test. The comparison of two benchmarking tests showed significant improvements in seamless information exchanges among these software (ATC 2013).
2.3.2.2. Precast/Prestressed Concrete Constructions

This research was done to develop IDM and MVDs addressing the interoperability issues in the precast/prestressed concrete industry. First, an IDM was developed for planning, design, documentation, construction and fabrication phases of construction projects using precast concrete elements (Panushev et al. 2010; Venugopal et al. 2012a). Next, five different Semantic Exchange Models (SEM) were defined and bound to the IFC file format. These five SEM supports the following five use cases:

- Clash detection among different domain-specific models like MEP and structural models. In this model view, element boundaries are the most important information.
- Structural analysis and design of the precast parts. This view represents parts in terms of nodes and axes, but does not represent the 3D object geometry. However, element weight and loading carried by them are included in the model.
- Fabrication of precast elements. In this view, the focus is on external boundaries of precast parts and hollow cores (internal boundaries).
- Aggregation representation. In this view, the geometry of the parent assembly is derived from the geometry of individual parts that aggregate the parent.
- Production and delivery sequencing. Items like piece counting and erection sequencing are of major importance in this view, while geometry is not addressed.
2.4. BIM in Modular Building Projects

At different stages of a project, information technology (e.g., BIM) in its most basic form has the potential to transform our understanding, means, and methods of performing design and construction in modular construction. During past decades, much effort has been made to evaluate benefits and challenges of implementing BIM in modular constructions. McGraw-Hill Construction in conjunction with National Institute of Standards and Technology (NIST) identified BIM technology as a major driving factor fueling interest in modular constructions (McGraw Hill Construction 2011). Lu and Korman (2010) identified the following areas beneficial for application of BIM in modular building projects: visualization, code reviews, fabrication/shop drawings, communication, cost estimating, construction sequences, conflict and clash detections. Furthermore, Nawari (2012) states that the following aspects could be enhanced using BIM in modular projects: speed, sustainability, safety, constructability, quality, and shortening construction time. Additionally, Song and Song and Song and AbouRizk (2006) found that virtual visualizations of shop production environment at a realistic level have numerous benefits for managerial purposes, while BIM could be used as a tool for this purpose. In the two case studies they carried out in their research, design coordination identified to be the best BIM use in modular projects, while difficulties in BIM implementation process itself, regardless of the software capabilities, identified as the greatest challenge for leveraging BIM in these projects. Because of these difficulties and inefficiencies, most of the small and medium modular companies cannot afford the
special man hour and team for BIM alignment in their projects. One of the case studies was an 110,000 SF healthcare facility expansion with CM at risk delivery method. In this project, $44,000 was spent in this project for BIM implementation, while the resultant saving estimated to be $220,000 at the end of the project (Lu and Korman 2010).

Huang and Krawczyk (2007) prototyped a web-based BIM platform that provides information filtering questionnaire to assist customers select their appropriate design components. Alwisy et al. (2012) developed a process model along with a supporting algorithm and tool for automation drafting and design of prefabricated elements of buildings. An intelligent system was developed in this research that utilizes BIM as an information platform to integrate architectural and structural design, modularity concepts and framing best practices to help clients during decision-making process. Precast concrete elements are one of the simplest prefabricated building element. Eastman et al. (2009) researched the supply chain of precast/prestressed elements. Neelamkavil (2009) has researched the potentials for automating activities in modular construction industry. He identified BIM as one of the most powerful potential platform for implementing automations in modular building projects. A more detailed literature review on application of BIM in modular building projects is included in Chapter 5.
2.5. Application of BIM for Engineering Analysis

According to CIC Research Group (2013), engineering analysis BIM Use is a process for determination of the most effective engineering method using an intelligent tools, design specifications, and BIM model. As a result of more rigorous and reliable analysis, saving time and cost by automating the process and achieving better quality and higher performance in buildings, are the most significant potential values of using BIM for engineering analysis purposes (CIC Research Group 2013). As Schinler and Nelson (2008) state, departing the traditional implemented process for engineering analysis, engineers can shorten the time they spend on modeling, design, and coordination with other disciplines.

To be able to use BIM models for engineering purposes, it needs to be transferred to engineering Design/Analysis (D/A) software. During the past decade, many efforts have been made to present a redundant and seamless infrastructure and platform to extract information from BIM model to be used in engineering D/A tools. These efforts could be classified in two categories. In the first category, these efforts are focused on the process of exchange to develop infrastructures and frameworks required for these exchanges. This mostly includes identification of exchange requirements and standardization of the way these information needs to be exchanged using IFC file format. Examples are IDM and MVDs developed for different engineering purposes. In the second category, efforts are focused on developing tools supporting the exchange processes. These tools are developed in different forms such as stand-alone software,
add-ons, or plug-in for server-based platforms such as BiMserver™ (Jiang et al. 2015; Liu et al. 2014; Wu 2013; Yu et al. 2013).

IDMs and MVDs developed by Virtual Building Laboratory (VBL) for architectural design to structural design, and structural design to Structural analysis information exchanges are an example of the efforts made in the first category. Another example is IDM and MVDs developed by U.S. General Services Administration (GSA) for conceptual design to building energy analysis.

There are also several examples for efforts in the second category. Liu et al. (2010) developed an integration tool to improve the IFC project file and export the structural model to PKPM structural analysis (a Chinese software) tool file format. Hassanien Serror et al. (2008) came up with an infrastructure, called Shared Computer-Aided Structural Design (sCAsD), for earthquake simulation of a structure using IFC file format within DOSE integrated construction management software environment. Zhang and Issa (2013) developed a tool in Java language for partial information extraction from IFC files. This tool benefit from IFC format ontology and an ontology-augmented model index. Wang et al. (2013) developed a tool to extract information required for structural D/A from IFC file of structural BIM. Qin et al. (2011) used an xml-based unified FEM format and programmed with C++ language for bidirectional converting information from IFC file format and translating to different proprietary structural analysis file formats. Polter et al. (2014) presented an under-development integrated cloud-based structural analysis platform that uses IFC file of the structural analysis model for some
minor design iterations such as thickness of a wall. Zhang et al. (2014) developed a web-based platform for bidirectional conversion of the architectural IFC model to proprietary structural modeling software file formats such as e2k and s2k. In addition, many efforts have used XML-based open standards for development of tools for information exchanges between BIM and engineering analysis software (Guzmán and Zhu 2014; Nepal et al. 2013).

A more detailed discussion and literature review on engineering analysis uses of BIM are provided in Chapters 7 and 9.

### 2.6. Summary and Application

Modular building construction is a growing industry, but due to its young age, the efficiency of projects in this industry is low. BIM as a tool potentially can address this issue by facilitating information exchanges and engineering design of these building and creating a more efficient collaboration environment, among other benefits. One of the key elements in BIM collaboration system is the seamless semantic information exchange. buildingSMART International, as an international organization, provides the infrastructure required for standardization of BIM information exchanges and development of interoperable framework for different building industry sectors.
Many efforts have been made so far to take advantage of the rich BIM information for engineering analysis purposes. Structural design/analysis and energy performance analysis are two of the most important types of these engineering analysis BIM uses. Although some BIM software are capable of addressing limited engineering analyses, since they are not usually as sophisticated as the stand-alone engineering analysis software, many of these efforts are made to develop translators between different engineering analysis software and IFC, which is the BIM open standard file format, or to extract required information from BIM model. Usually, these translators and information extractors could be found in three different forms consisting of a stand-alone package, an add-on for other software, or an extension to a server-based platform.

2.7. References


Group, The Pennsylvania State University, bim.psu.edu website, University Park, PA, USA, <bim.psu.edu> (Feb. 2, 2015).


and Construction, 17, 134–161.


MD, USA.

Lee, Y. T. (1999b). “Information modeling: From design to implementation.” National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA.


feature modeling for construction information extraction from a building information model.” *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, 27(5), 555–569.


CHAPTER 3: Methodology

3.1. Introduction

To facilitate implementation of BIM in multi-story modular buildings, different aspects of these projects need to be researched. In this research, the product- and process-related information that may be generated or used at different stages or phases of the project as well as the structural design/analysis-related information interpretations are studied and researched. Each of these aspects was set as an objective in order to achieve the main research goal that is increasing the productivity in multi-story modular building projects. Several tasks were defined to be accomplished for each of these objectives. Since these objectives are interconnected and partially supporting each other, defining their related tasks made it possible to design the time schedule and the sequence of the tasks to have a rich enough input for each of the research activities. These objectives along with their related tasks are discussed in more details in the following sections.
3.2. Objectives and Tasks Outline

Bernold and Lee (2009) recommend that a complex investigation such as this study be sub-divided into small task components, which are then handled separately. In the following, the three main objectives of the research and their associated tasks are outlined.

**Objective 1:** Develop a Product Architecture Model (PAM) based on conventional modular systems and identify how a multi-story modular system is aggregated from different assemblies and subassemblies. The outcome of the work that is done to accomplish this object is discussed in Chapter 4.

![Figure 3-1. Primary tasks for Research Objective 1](image)

**Objective 2:** Develop a BIM framework and its supporting infrastructure for information exchanges in modular building projects that address the interoperability issue of the software in this industry. The work that is done regarding this objective is discussed in Chapters 5 and 6.
**Objective 3:** Develop a mechanism for information exchange between different domain-specific tools and its implementation for structural design/analysis model generation of modular buildings. The outcomes of the work regarding this objective are discussed in Chapters 7, 8, 9, and 10.
3.3. Further Explanation of the Tasks

The tasks outlined in previous section are described in more details in the following subsections.

3.3.1. Tasks for Objective 1

To achieve this objective, the following four tasks were carried out:

Task (a). Recognize different modular systems:

The first task was to recognize the most conventional modular building systems. The focus of this task was on structural and general architectural components. This was done by literature reviews, conducting (limited) interviews with different parties in this industry, and site-visits. In addition to geometry and types of components, the functionality and application of each component was evaluated as well.

Task (b). Develop hierarchy of modular systems to assemblies and subassemblies:

This task divided modular buildings into their assemblies and subassemblies to show different options for each component.

Task (c). Define attributes and properties of the components required during building lifecycle:

In this task, relevant properties and attributes of each component required for product lifecycle management are defined. An attribute could be a general attribute like the
weight of one modules or a very detailed engineering attribute like the stiffness of connection within the modules.

Task (d). Categorize applicable components based on the Levels of Detail (LoDs):

In the last task, the hierarchy, properties, and attributes resulting from previous tasks are categorized based on the definition of Levels Of Development standardized by the NBIMS-US. The NBIMS-US has categorized BIM information pieces in six Levels Of Development (LODs) and has defined them by description of their characteristics. In this task, different hierarchy, properties and attributes are labeled based on their best fit to the NBIMS-US LODs description.

The proposed modular building PAM provides a comprehensive information database for future analysis and evaluation of modular buildings. The proposed PAM is the result of (limited) interviews and site visits (Modular manufacturers visited or interviewed include Capsys Corps, Deluxe Building System, NRB Inc., Mark Line Industries, and Triumph Modular companies), and review of the constructed modular buildings. More detailed methodology for development of the PAM in discussed in Chapter 4, while application of the PAM in other parts of the research is pointed out in Chapters 5 and 6.
3.3.2. Tasks for Objective 2

In developing the BIM framework for modular buildings, the following four tasks were considered.

Task (a). Develop process map of the design phases:

In this task, a process map for design activities of the project was developed. As NBIMS-US suggests, this process map is developed based on the Business Process Map and Notation (BPMN) and covers the lifecycle of modular buildings from early stages of design to manufacturing and construction phases. Additionally, in design areas, where interpreted information exchanges are handled for structural analysis BIM Use, the process map is expanded into more details by developing some sub-process map to catch all the sub-activities.

Task (b). Define Task Units and Exchange Models that are recognized in the process map:

The proposed process map contains many Task Units (activities) and Exchange Models (information exchanges). Defining these Task Units (TUs) and Exchange Models (EMs) is the second task under Objective 2. The TUs definition contains five parts, including project stage, discipline(s), input EMs, output EMs, and a short description of the TUs. The definition of EMs contains the following four parts: type (document or model), project stage, involved disciplines, and a short description of the EM.
Task (c). Map the Exchange Models to the proposed PAM:

This task involved mapping the EMs to the proposed PAM to come up with the Exchange Requirements (ERs) Specification. The ER Specification specifies the information about physical properties or attributes of different assemblies or elements that have to be exchanged in different EMs. In other words, while the PAM defines the information pieces about different aspect of the building, EMs describe the information needed to be exchanged at each stage of the project, and ERs Specification itemizes the descriptive EMs to come up with a definite specification of the exchange requirements. Since the LoDs are defined in the proposed PAM, the LOD of the BIM model at each stage of the project could be measured based on the exchange modules.

Task (d). Define structural Model Views:

The output of the first three tasks under this objective is IDM in National BIM Standard – United States terminology. The last task under this objective was the definition of structural Model Views based on the Industry Foundation Classes (IFC) file format data schema. Nawari and Sgambelluri (2010) define the Model View Definition (MVD) as the software developer interface of the exchanges. In this task, the functional specification of the IDM is translated to a human-readable format that later could be used to store information in a digital format. In this task, the defined information exchanges in IDM are organized in IFC specification hierarchy schema to make it possible to map the required information to the IFC predefined entities. In
this research, many new MVD concepts are defined to address relevant specific properties of modular buildings, while using existing library concepts as much as possible. The developed MVD could be implemented by software developers to make information exchanges of the modular building models seamless and reliable, and in this way address the interoperability issue of model exchanges in modular building projects.

The proposed process map and Exchange Requirements Specification is the result of (limited) interviews and site visits (as mentioned earlier), and review of some constructed modular buildings available in the literature. More detailed discussion on the methodology used for development of the IDM is included in Chapter 5, while the MVD development methodology is discussed in Chapter 6.

3.3.3. Tasks for Objective 3

Research Objective 3 has five primary tasks that are performed to develop a mechanism for structural information interpretation from architectural BIM in modular building projects.

Task (a). Categorize structural related data into direct and interpreted information categories:

The first task focused mainly on recognition of information within the structural design/analysis model of modular buildings that could be exchanged directly from the
BIM model and the information that needs to be interpreted before being exchanged to the structure analysis/design software. In other words, the physical information and their structural properties and attributes from the PAM are divided into two categories, which are direct and interpreted information.

Task (b). Map the structural related interpreted information to the related BIM information:

This task is the mapping of the interpreted information recognized in Task (a) to the related BIM model information. In other words, the information in the BIM model upon which the interpreted information could be extracted or be input by the structural engineer are specified. The result of this task is specification of the inputs and outputs of each interpretation (translation) unit. These inputs and outputs are all based on the hierarchical information of the PAM.

Task (c). Design a translation mechanism for interpretation:

This task defines a mechanism for translation of the interpreted information from the BIM model. In this task a series of interpretation mechanisms is designed to connect the inputs and outputs of the translation units that are defined in Tasks (b).

Task (d). Bind the interpretation mechanism to the IFC file format Data Structure:

Binding the translation engines to the IFC file format data structure is the fourth task under Objective 3. This task specifies the IFC file instances that should be replaced by the interpreted information in the form of new instances based on the designed interpretation mechanisms in Task (c). IFC is an open standard file format for BIM
models; therefore, binding these engines to the IFC makes it possible to use them in large variety of BIM tools.

Task (e). Testing the proposed interpretation mechanism:

The last task was testing the proposed interpretation concept by a case study. In this test, an interpretation mechanism and a tool are developed for automating generation of structural analysis models of conventional buildings directly from their building information models. This mechanism is a portion of the interpretation mechanism designed for structural modeling of modular buildings.

A more detailed discussion on development of an automated interpretation mechanism for structural modeling of modular buildings is provided in Chapter 10. The methodology that was developed for design of automated interpretation processes (Interpreted Information Exchanges (IIEs)) is discussed in Chapter 7, while the methodology used for testing this concept is explained in Chapter 9. Chapter 8 provides information on the methodology used for development of a platform that can be leveraged for implementing the IIE concept.
3.4. **Research Process Summary**

This research was carried out through accomplishing several tasks related to three research objectives. First the Product Architecture Model (PAM) of multi-story modular buildings was developed. The PAM is a structured breakdown of the building elements and their related attributes and properties. To achieve this objective, a comprehensive literature review was carried out, and a series of site visits and interviews with industry experts were conducted. The next objective was information exchange standardization in these projects focusing mainly on structural aspects. To satisfy this objective, an IDM and a MVD were developed based on the characteristic and workflow of modular building projects identified by the site-visits, the interviews, and the literature review conducted for the first objective. Although the developed IDM and MVD are focused on structural aspects of modular buildings, it could be expanded and/or modified to support other BIM application areas. The last objective was facilitating the structural design of complicated structure of modular buildings by development of a mechanism to interpret structural design/analysis model of modular buildings directly from their architectural BIM. In addition, a software platform was developed for implementation of such mechanisms that can be used for automating the interpretation process. This platform imports the building’s architectural BIM and generates the IFC file of its structural design/analysis model.

These objectives are also bound by a holistic multidisciplinary approach considering information technology and structural engineering principles. For more
clarification, the relation among different objectives, including a diagram picture of the research, is depicted in Figure 3-4.
Figure 3-4. Visual Representation of the Objectives

1. Develop Product Architecture Model

2. Develop Lifecycle BIM Framework for Modular Buildings

3. Develop a mechanism for information interpretation for structural modeling
3.5. References


CHAPTER 4:  Product Architecture Model (PAM)

This paper is currently accepted for publication in the *ASCE Journal of Construction Engineering and Management*. Republishing of the material in this dissertation is with permission from American Society of Civil Engineers (ASCE). This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers.

Product Architecture Model for Multi-Story Modular Buildings

Issa J. Ramaji, Ph.D. Candidate¹, Ali M. Memari, Ph.D., P.E.²

¹ Ph.D. Candidate, Department of Architectural Engineering, Penn State University, 104 Engineering Unit A, University Park, PA 16802, E-Mail: ramaji.issa@gmail.com
² Professor, Department of Architectural Engineering and Department of Civil and Environmental Engineering, Penn State University, 222 Sackett Building, University Park, PA 16802, E-Mail: memari@engr.psu.edu
Abstract: Modular building construction method is the industry’s effort for industrialization of building construction. Similar to other types of production, a product information model is required for lifecycle information management of modular buildings. In the research reported here, the mutual hierarchy of typical multi-story modular systems is identified and represented in a Product Architecture Model (PAM). This provides the industry with an information model required for industrial design of this type of construction. Since modular construction is neither heavily project-based as conventional construction methods nor significantly mass-produced as manufactured products, none of the current information management frameworks in manufacturing and construction industries is necessarily capable of addressing the needs of such projects. In the research reported here, this need is addressed by development of the Product PAM for the most conventional modular systems. The PAM represents the hierarchy of modular building components along with their interactions, functionality, and attributes. This paper initially reviews major types of modular systems, and then focuses on the development methodology, validation, architecture, and content of the PAM. This is followed by an explanation of its different uses, including extension of the current construction industry’s information management frameworks and platforms such as Building Information Modeling (BIM) to support modular construction, and providing a basis for product-base design of such buildings.

Keywords: Modular Building, Product Architecture Model, Product Lifecycle Management, Level OF Development, Information Exchange, Building Information Modeling, Prefabrication.

4.1. Introduction

Industrial production of prefabricated components involves the use of applicable standards and certain production lines in manufacturing plants (CIB 2010; Kamar et al.
According to Teicholz (2013), while labor productivity doubled in most non-farm industries, it has decreased in the U.S. construction industry by 10 percent from 1964 to 2004. In spite of the fact that new materials and systems are being used in projects, the construction process has not changed significantly. Factory manufactured homes, modular homes, panelized components, and prefabricated structural frames are examples of efforts initiated to address the need for industrialization of building construction. According to Smith (2011), prefabrication in construction industry is “evolutionary not revolutionary”, and similar to the medical field, this industry has evolved based on experience, with both successes and failures. Unfortunately, contrary to the potential and promise of modular building construction, some may still associate it with cheaper, lower quality, and/or temporary housing. One huge challenge is transformation of the conventional design/construction approach to one based on manufacturing that requires appropriate framework, tools, and technology that can change the perceived image of modular construction (Giles and Lara 2006; Ikuma et al. 2010; Taghaddos et al. 2012).

The use of a product development model for design and delivery of modular buildings can result in a more product-based industrial design approach (Giles 2008). Moving toward industrial production of modular buildings is not just tied to the design stage but the entire process of design, manufacturing, construction, maintenance, and deconstruction/reuse stages (Aram et al. 2013; Doran and Giannakis 2011; Giles 2008; Moghadam et al. 2012; Ramaji et al. 2014; Schoenborn 2012; Solnosky et al. 2014; Song and AbouRizk 2006; Yu et al. 1999). Therefore, similar to other industrialized products, the Product Lifecycle Management (PLM) concept needs to be implemented in modular
building projects to increase productivity. Product information model is an essential aspect of PLM systems that should represent different elements and attributes of products along with their relationships (Sudarsan et al. 2005). Information modeling is a technique for specifying the data requirements within the application domain for using a shared database (Lee 1999b). An information model spans the lifecycle of a product and includes its conceptual representation and description of a Universe of Discourse (UoD) for a product. UoD is defined by Halpin and Morgan (2010) as the application area from which the information model is being developed. Information models should be designed to be capable of accurate expression of a user's conception of some portions of the UoD (Eastman and Fereshetian 1994).

According to Sudarsan et al. (2005), to use a product information model for an industry, it should be generic, non-proprietary, extensible, independent of the production method, and capable of capturing all product aspects commonly being considered in that industry. The National Institute of Standards and Technology (NIST) developed a Core Product Model (CPM) that can generically serve the production industry as a base-level information model (Fenves et al. 2008). CPM represents the physical and functional properties of a product’s components as well as their behavior and properties. Several extensions are also added to the CPM to enable it represent the hierarchy of product’s components, generating functional models of products, and semantic definition of product assemblies in order to enhance interoperability of Computer Aided Design (CAD) systems used for information management of products. The Open Assembly Model (OAM), the Design-Analysis Integration Model (DAIM), and the Product Family
Evolution Model (PFEM) are some other alternatives for such information models serving the production industry. These abstract information models are developed to better serve 1) products that are made up of several assemblies within which system-level tolerances are of great importance, 2) projects within which computer-aided engineering is being vastly used to integrate the analysis and design stages, and 3) products that are highly customizable within which minimizing the customization cost is an important factor (Fenves et al. 2003; Sudarsan et al. 2005).

In addition to information management, information models have also been extensively used during the past two decades for product family design to optimize design, minimize production waste, manage supply chain, standardize components of products, and manage product changes and options (Jiao et al. 2007; Ulrich et al. 2011). Similar efforts have been observed in the construction industry for development of information modeling and management frameworks for lifecycle management of buildings, which is mostly aligned with advancements in Building Information Modeling (BIM) technology (Amann 2002; Eastman et al. 2011; Laakso and Kiviniemi 2012; McGraw Hill Construction 2012; NIBS 2012; NRC 2009). Development of Information Delivery Manuals (IDMs), Model Views Definition (MVDs), and standardization of Levels of Development (LODs) are examples of such efforts, which are generally related to identification of information requirement in typical construction projects, as well as flow and semantic representation of the information (NIBS 2012). One of the main differences between these efforts and other information frameworks in the production industry, such as CPM and its extensions, is that the structure of building elements and
their properties are not considered in the construction industry. Due to the custom-design nature of site-built construction, the hierarchy and relations between elements and their attributes are not generally considered as is the case in the production industry. However, this aspect is important in modular construction, especially in multi-story modular building projects (Giles 2008). On the other hand, modular construction is not fully industrialized because it shares many elements and characteristics with conventional construction industry, including some site-built parts and on-site activities. As a result, the information modeling frameworks available for both site-built construction and production industry is not suitable in addressing the need of modular construction (Alwisy et al. 2012; Eastman and Sacks 2008). In addition, the proprietary nature of modular systems has made current frameworks even less applicable in multi-story modular projects (Badir et al. 2002; O’Connor et al. 2015).

As illustrated in Figure 4-1, a generic information model called Product Architecture Model (PAM) is developed in the research reported here to address the need of multi-story modular building industry for a comprehensive product model, and to integrate the hierarchy and relationships among components and attributes of modular buildings into the current information management frameworks (Fujita and Yoshida 2004; Rai and Allada 2003; Ramaji and Memari 2015a; Sause and Powell 1990; Stone et al. 2000; Yu et al. 1999). Although the PAM is generic, it follows the mutual hierarchy of typical modular building systems that are being built around the world. For example, in all of these systems, the building may consist of two parts: site-built and prefabricated; the latter is composed of several modules whose dimensions are important attributes. The
PAM is structured based on such high level mutual hierarchy along with common attributes of the objects forming the hierarchy. The PAM serves a variety of modular systems and can be expanded to contain the information that may be generated or used throughout the building’s lifecycle. Product architecture is defined by Ulrich et al. (2011) as a model that describes allocation of functional elements of a product and interaction with its physical elements. In addition, Fixson (2005) defines it as a comprehensive representation of the fundamental structure of a product that includes product characteristics and type of components as well as components’ interfaces. According to Ulrich et al. (2011), PAM contains the following three information aspects: 1) arrangement of functional elements, 2) mapping of functional elements to physical components, and 3) specification of interfaces among interacting physical components.

The PAM presented in this paper is developed based on the conventional modular systems discussed in the following section, which specifies system component information and identifies aggregation of multi-story modular buildings from different assemblies and subassemblies. Object-oriented nature of the developed PAM makes it suitable for expansion to any discipline-specific uses. For example, the PAM presented in this paper is expanded for structural uses and contains structural attributes of the building components.
Figure 4-1. Limitations of production and construction industries to address the needs of modular building industry and the way that the PAM can fill the gap

4.2. Anatomy of Modular Buildings

During the past two decades, numerous multi-story modular buildings with different innovative systems have been built around the world (MBI 2011). Multi-story modular buildings have their own special features and characteristics and pose their own challenges (Annan et al. 2008). Considering construction sequences and component
types, one can categorize modern multi-story prefabricated buildings in the following six major groups: 2D, 3D, Hybrid Cored-Modular, Hybrid Podium-Modular, and Framed Unit systems (Ramaji and Memari 2013), which are briefly discussed in the following paragraphs. It is noteworthy that the construction industry uses the term “modular building” for all these systems except the 2D system.

In 2D (planar) systems, prefabricated floor and wall panels are installed and assembled along with columns, if any, to shape the whole building. First, columns and walls of a story are erected, followed by installation of floor cassettes. A good example of this system is the 30-story hotel near Dongting Lake in China that was erected in just 15 days (Hilgers 2013).

In 3D modular systems, the building consists of many volumetric modules stacked vertically and attached horizontally to form the building. Each module can be part of a unit, one complete unit, or even contain more than one small unit (Lawson and Ogden 2010). The size of a module is a function of the module location in the building, manufacturing and transportation limitations, and available construction equipment. Usually, the module floor areas are in the range of 100 to 600 square feet (Lawson and Richards 2010). In addition, because of transportation limitation, as a rule of thumb, width, length, and height of the modules are usually in the ranges of 3.5-5 m, 12-18 m, and 3-4 m, respectively (Garrison and Tweedie 2008).

As the number of stories increases, the magnitude of the total gravity and lateral loads increases, and therefore, the size of load-bearing elements in lower story modules
need to be larger beyond what is needed just for gravity loads. Accordingly, beyond a certain point, the size of these elements and the weight of the individual modules could become a challenge for transportation, erection, and installation because ideally, prefabricated modules should be kept light. In the hybrid cored-modular system, the problem is solved by considering staircases and elevator cores as part of the structural system that resist the lateral loads and limit the size of internal elements of the modules. An example for this system is the Student Village building, which is a 25-story building located in Wolverhampton city of the United Kingdom (U.K.), where the stacked modules rely on a concrete core to resist lateral loads (Lawson et al. 2012; MBI 2010).

One of the limitations of modular buildings is the short bay spans in lower stories, whereas ideally longer spans are normally desired for retail or parking spaces. The hybrid podium-modular system is a solution in modular projects that offers opportunity for longer bay spans in lower stories (Lawson and Richards 2010). In podium-modular systems, the below-ground and some of the lower stories (usually the lowest one or two stories) are site-built using conventional structural steel or concrete frames with long spans, which support the upper modular parts of the building.

In Framed Unit System, the main load resisting structure of the building is a conventional structural frame. After erection of the main structural frame, prefabricated modules are placed and fitted between beams and columns of the structural frame that support the modules. One of the most important advantages of this system is that the module properties do not depend on the plan shape, number of stories, and lateral loads.
Since the main structural frame carries the gravity and lateral loads, units for all stories could be identical. In addition, the elements making up the modules are designed just to carry their own gravity and construction loads; therefore, the modules in this system are lighter compared to the other 3D system types.

4.3. Development of the PAM

The PAM is developed using a three-tier method, with the first recognizing the hierarchy of components that may be aggregated in a modular building. Identifying properties and attributes related to the lifecycle of modular buildings and assigning them to their respective components forms the second tier of this method. Most of the activities for extensions of the PAM are carried out in this tier. The last tier categorizes the identified components and related attributes based on Levels of Detail (LoDs). In the following sub-sections, each of these steps is discussed separately. The required information for development of the PAM has been acquired by carrying out a comprehensive literature review, conducting limited interviews with different expert parties in this industry, and through a series of modular factory site-visits. In addition to the geometry and types of the components, the functionality and application of each component are surveyed as well. Furthermore, to validate the outcome, the (currently)
tallest multi-story modular building is modeled with the PAM to check the capability of the model to capture physical components and their basic information.

An object-oriented approach is used for development of the PAM. Like any other object-oriented information model, each object contains a series of attributes and has at least one relationship (connection) with other objects that may describe association, composition, etc. Objects are information units that represent a physical component or assembly. An object could be an atomic component such as a column or a layer of insulation, or it could be a complex assembly such as prefabricated part of a building, foundation, or a volumetric module. The attributes are information units that describe physical objects. An attribute could be describing a physical property of a simple atomic component such as thickness of an insulation layer, or could be a characteristic information about a complex object resulting from all internal elements of the assembly, such as the overall thermal resistance of a wall assembly. In the PAM, as shown in Figure 4-2, the attributes of each object are listed under the object’s name within the box representing the object, while the relationships are shown with lines connecting objects. Different forms of these lines represent different types of relationships. Since the object-oriented approach is used for development of the PAM, a conventional object-oriented graphical language called Unified Modeling Language (UML) is used for its representation. UML is a general-purpose modeling language used for system design graphical visualization (Booch et al. 1999; James et al. 1999).
4.3.1. Development of Product Breakdown Hierarchy

For product development hierarchy, modular building systems are divided into parts, which are broken down into their assemblies and subassemblies. The breakdown process is continued to a finer level where the components are typical low level construction objects such as beams, columns, insulation layers, etc. To show different options for each component and distinguish their relations and interactions, the breakdown is developed based on the design and production sequences of the assemblies and functionality of the subassemblies. The production sequences of different assemblies are in a fair parallelism with the design sequences of these objects, although they flow in reverse direction. The production sequences are from components to assemblies (bottom-up), while the design processes are from assemblies to subassemblies and details (top-down). For example, as shown in Figure 4-2, modular buildings are divided into two parts, site-built and modular. The rationale behind this classification is that these two parts are constructed separately and belong to two different construction and manufacturing sequence branches. Another example is breaking down the volumetric modules into the following five subassemblies: floor, ceiling, walls, openings, and columns. This classification is needed since these subassemblies are usually prefabricated and later assembled to form a module for different functionalities. In the design process, the building is first broken down into different modules, then subassemblies such as ceilings, floors, and openings are specified, followed by more detailed design of the components of these subassemblies.
The PAM contains two layers consisting of integration and abstraction, as shown, respectively, in Figure 4-2 and Figure 4-3. The abstraction layer contains some abstract objects that are referred to in the integration layer. An abstract object is a generically defined object that represents components sharing some identical properties or characteristics. Unlike the abstraction layer, all objects in the integration layer are concrete objects, which represent specific physical components. Some of the objects in this layer inherit their properties from the abstract objects and are the connection between these two layers. The relationship between these two layers is not bidirectional; only abstraction layer can inherit from the abstraction layer and use it as a resource. Enhancing resiliency of the PAM, avoiding multiple objects with minor differences, and minimizing the size of the PAM are three main reasons for considering an abstract layer in the PAM. This enhances the flexibility of the PAM to support new products and innovative assemblies with no or minor extensions. For example, there are numerous types of Structurally Insulated Panels (SIPs) made up of different assemblies and materials available in the industry that provide structural strength and thermal resistance. Since the PAM defines the objects based on their functionality and attributes that affect the building’s properties, all these products are modeled with the same abstract objects in the PAM. In such a system, if a new product becomes available in the market that has the same functionality and main attributes, it would be supported by the PAM with no required changes.
Figure 4-2. Integration Layer of the PAM developed for multi-story modular buildings

Since the PAM is structured to address all the stages of modular building projects, it includes a LoD 500 layer. This layer is left blank as information units related to this layer are out of scope of this research.
Figure 4-3. Abstract objects defined in the Abstraction Layer of the PAM
The objects in abstraction layer belong to three different abstract class objects consisting of linear, planar, and opening objects, whose names start with LOT (Linear Object Type), POT (Planar Object Type), and OOT (Opening Object Type), respectively. All these class objects have an interface realization to a connection that is also broken down into more details in this layer. These abstract objects that are used for defining different concrete objects in the integration layer contain, respectively, six, seven, and one abstract subtypes, respectively. Each of these abstract objects could be super-type of only one or multiple concrete objects. For example, POT-01 is used just for defining Slab object, while LOT-01 is used for both Perimeter Beam and Middle Beam objects in the integration layer. Moreover, Connection is the only interface object developed in the PAM, and has two sub-types consisting of Bolted and Welded Connections.

In the UML notation, the connectors define the relationships between different objects. The following four types of relationships are used for development of the PAM: composition, interface realization, inheritance, and dependency. Each of these relationship types, which are defined subsequently, has a special symbol in the PAM, as shown in Figure 4-4.

![UML notation of the four types of relationships used in the PAM](image)

Figure 4-4. UML notation of the four types of relationships used in the PAM
**Composition:** Composition connector makes a string type of aggregation and connects the parts to the whole assemblies. In this type of relationship, parts may be created after the assembly, but once created, their continued existence depends on the existence of other parts, and by deletion of the assembly, all parts disappear.

**Interface Realization:** This type of relationship connects an object to an interface, which is also an object but connects two objects together. Therefore, an interface object needs to be connected to at least two objects using interface realization connectors. All implementation rules of the connections are defined in the interface object. In the PAM, connections between building components are represented by interface objects, which are automatically created when assemblies are broken down into subassemblies and components, although additional information may be added during the other project phases. Therefore, unlike their attributes, the interface objects (connections) do not have any associated LoDs and are black-colored with no assigned symbols.

**Inheritance:** This type of relationship creates a link between a child object and its parent in the object-oriented environment. The parent generalizes the behavior and properties of the objects sharing similar characteristics with minor differences. In this relationship, the child inherits all the attributes of the parent, while it may have its own additional special attributes.

**Dependency:** In this type of relationship, a supplier object is related to a client object. The supplier object provides the information needed by the client and any changes in it
may affect the client. This kind of relationship is usually used for grouping closely related information.

A few additional features have been added to the UML notation to fit the scope of this research and make it applicable for multi-layer representation of the PAM. For example, the objects and attributes are color-coded with symbols to reflect the LoD they belong to. For more clarification, a prototype of an object is shown in Figure 4-5. As shown in this figure, all the information about an object and its related attributes are contained in a box. The name of the object is located at the header of the concept box and is color-coded based on its LoD category. Blue, green, brown, orange, violet, and yellow represent, respectively, LoD 100, LoD 200, LoD 300, LoD 350, LoD 400, and LoD 500. Besides color-coding the object box, the LoD of the objects are also represented at the top right hand side of the object box with “#”, “*”, “-”, “+”, “^”, and “::” symbols, respectively for the above mentioned LoDs. Because LoDs are independent from the attributes and their related objects, the attributes may have a color other than that of the object box. In this case, the LoD of the attribute is specified at the top right hand side of the attribute’s name as shown in Figure 4-2.
4.3.2. Assigning Attributes to Assemblies and Components

Some additional information need to be generated in each stage of a product lifecycle. Examples of such information include high level attributes such as number of stories, wind load, or layout of the building plan, which are related to the higher level objects, or low level attributes such as rigidity of a connection between two members or the thickness of a wall board. In this step of the PAM development, first, the attributes required for product design and lifecycle management of different assemblies and components are identified. These attributes are related to functionality, constraints, and relations/interfaces among different entities. Next, the proposed attributes are mapped to
the related objects in the developed hierarchy from the previous step. These attributes could be related to a simple component like a column or a group of components resulting from the formation of an assembly such as a SIP panel. As an example, as shown in Figure 4-3, structural panel has several attributes, including resistance, stiffness, and R-value, which are related to engineering analysis domain and are derived from parameters such as material properties of different layers, cross section or thickness of each layer, and related connections.

In the PAM, the object that has an inheritance relationship, inherits its attributes from their parent objects. In the modified UML notation used in this paper, if the object is inherited from one of the abstraction layers, the name of the parent concept is noted right under the header at the right side. For example, in the object displayed in Figure 4-5, LOT-01 from the abstraction layer is referred to as the parent; therefore, the object’s attribute set includes all attributes inherited from LOT-01 along with attributes specifically defined for it. The special attributes of the objects, if there are any, are listed in the object box under the name of the referenced abstract object. A dashed line separates the referenced abstract object from the special attributes of the concrete object. If the object is not referred to any abstract object, the attributes are listed immediately under the header of the box.
4.3.3. Categorizing Objects Based on Level of Detail

In this step, the Levels of Detail (LoDs) for modular buildings are defined based on the Levels Of Development (LODs) standardized in the National BIM Standard – United States® (NBIMS-US™) (2012). As per American Institute of Architects (AIA) (2013) definition, LOD specifies the minimum content required to be included in a model and authorized uses of the information. Standardization of LODs enables Architecture, Engineering, and Construction (AEC) practitioners clearly specify and articulate the content and reliability of BIM in models during various stages of projects (Associated General Contractors (AGC) BIMForum 2013). As a result, in addition to the abstraction and integration layers, there are six sub-layers defined in parallel with the other two layers to show different LoDs. These layers are named after the LODs defined in the NBIMS-US™-Version 2. This high level classification determines the extent to which assemblies have to be designed at each LOD along with the attributes needed at that level. According to National BIM Standard – United States-Version 2 (NIBS 2012), there are six Levels Of Development that start with LOD100, which is a generic representation of the building. Other LODs including LOD 200, LOD 300, LOD 350, LOD 400, and LOD 500 progressively require more information and detail for representation of the building in terms of size, shape, location, quantity, orientation, and some non-graphic information. Categorizing the PAM into different LoDs specifies the information needed to be represented at each LOD of the product design. All objects contained in each layer have LoD Corresponding to its layer. The LoD of an object indicates the level of development at which the object and its general location in the project should be
specified. On the other hands, the attributes may not belong to the LoD of the object they belong to. In other words, the LoD of the attributes are independent from the LoD of the objects, although the majority of the attributes have the same LoD as their related objects.

Objects up to LoD 300 belong to the integration layer, while the rest are included in the abstraction layer. The integration layer refers to the abstraction layer for showing information about higher LoDs. Above LoD300, breaking down of the assemblies is continued in the abstraction layers. The association of each object and attribute to a certain LoD layer could be realized either by colors or symbols.

The difference between Level of Details (LoDs) and Levels Of Development (LODs) should be distinguished when the PAM is used for information requests. According to AIA (2013) and AGC BIMForum (2013), LoDs specifies the extent of details included in the BIM model for each element, while LOD indicates the degree to which one can rely on each element’s information when using the model. In other words, LoD measures input to the element, while LOD measures reliable output. The LoD and LOD concepts measure enrichment of the content of a product’s model from two different perspectives, while they use the PAM’s LoD definition and its hierarchy as a scale. These two perspectives are further explained in the following.

**LOD in a project-specific product design model:** In this cast, the LOD specifies the extent to which the information is designed and could be used for other purposes by other disciplines. In other words, it specifies the extent to which each piece of information contained in the product’s design model is reliable for decision making applications. For
example, if the LOD of a structural frame model is 300, it means that it is not ready for fabrication purposes, but the size, shape and orientation of the members are reliable enough to be used by the architect.

**LoD in a project-specific product model:** In this cast, the LoD specifies the information that could be found in the model. It specifies the available details regardless of the reliability of the information included in the model. For example, the architect may use a model from previous projects that contains some structural information such as the details of a beam-to-column connection; in this case, the LoD is high for that connection, but since the structural engineer has not reviewed it yet, its LOD is low and the steel fabricator cannot use this connection information to prepare the shop list.

The LoD definition in the PAM provides a manageable definition of the information that may be requested by different actors at different lifecycle phases of the product. It provides a common language for project disciplines to ask for a design model with a certain level of enrichment. For example, the structural engineer may ask for a LOD 200 model to be able to start the structural design development of the building. Having the PAM classified based on LoDs, the architect knows he/she should distinguish the volumetric-modular and planar-modular parts of the building and has to specify information such as location and dimensions of the modules along with general information about their main sub-assemblies including floors, walls, ceilings, columns, and openings in the model.
For more clarification, the differences among these two concepts are depicted in the workflow shown in Figure 4-6, where the Receiver Unit (RU) asks for a model and specifies the minimum desired LOD using definition of the LoDs in the PAM. The requested LOD may be an identical LOD for all the building’s objects or may be a list of LODs that specifies the desired design development for different objects separately. Based on the requested LOD and definition of its equivalent LoD in the PAM, the Sender Unit (SU) prepares a list of information that should be included in the model, and accordingly prepares a product model with a LOD higher than that requested by the RU.

Figure 4-6. Workflow of using the LoDs defined in the PAM for defining exchange requirements in modular building projects
4.3.4. Validation

The intention of the PAM is to provide the construction industry with a repository of objects and attributes that could be used for generation of the information model for typical modular buildings. In other words, if we consider the information model of a building as a Lego that represents the building, the PAM provides the Lego pieces and the connectors that are required for connecting them together. In this example, the hierarchy specified in the PAM acts as an instruction that represents general constraints for assembling the Lego pieces to represent the modular building and shows how the pieces and connectors should be assembled. Considering the intention of the PAM, this repository should contain all the objects representing components and elements of a typical modular building; in addition, the relationships defined in the PAM should not be different from the mutual high-level architecture of such buildings. The PAM is iteratively revised and extended to capture all possible assemblies and related attributes being used in these buildings. During the iteration process, the PAM is compared with different commonly used modular building system types, and is improved accordingly to enable it capturing all the elements and attributes being used in this type of construction, and to make sure it is generic enough that it does not violate the hierarchy of typical modular systems. With such development methodology that is aligned with the objective of PAM development, the information model for typical modular buildings can be developed leveraging the PAM. To test this, the developed PAM is validated through generation of an information model for the Student Village building in the U.K., which is currently the world’s tallest ever-built modular building. The system being used in this
25-story building is hybrid modular that includes a concrete core and a podium structure. The hybrid system of the building makes it a good option for validation of the PAM, as it is a complex building with a large variety of assemblies and components used in typical modular construction. The building includes a concrete shear wall as the primary lateral load bearing system and one story of podium at the ground level. The façade of the building is made up of several lightweight panels that are attached after the modules are installed.

The validation is performed by testing whether the objects and relationships defined in the PAM are sufficient and applicable for generating product information model of this building. To do so, the building is broken down into several assemblies and subassemblies based on the information available related to fabrication and construction of the building. Then, each of the objects along with their basic attributes will be modeled by the objects, attributes, and relationships defined in the PAM. The outcome of this effort is shown in Figure 4-7, which is an information model for the Student Village building. Since the PAM is designed to be generic enough to capture all the components and basic properties of typical modular buildings, one metric of the validation can be the number of the building components and related basic information in the Student Village that could not be modeled using the objects, attributes, and relationships defined in the PAM. Another metric would be consistency of the architecture and the relationships between assemblies and subassemblies of the case study building with the hierarchy embedded in the proposed PAM. As illustrated in this figure, the PAM is capable of capturing all the physical assemblies, components, and element used in the building along
with the basic attributes and properties required for design coordination in this building. The information related to this building was obtained based on the available published literature, while some of the building attributes were estimated based on relevant available information (Lawson et al. 2014, 2012; Lawson and Ogden 2010; Lawson and Richards 2010; MBI 2010).

The case study building does not contain all the components and assemblies of all the modular building systems. However, since the PAM is extended and revised to capture different modular systems throughout its iterative development process, this validation case study shows that the PAM is capable of serving the systems it is designed for. The targeted systems are typical modular building systems briefly discussed in the previous section. The PAM is developed using a bottom-up approach, within which first, typical modular buildings are identified and investigated, and then the PAM is developed in a manner that is capable of serving the identified modular systems. As a result of such approach, the consistency of the PAM with targeted modular systems and its ability to capture the shared and specific information units of these systems is continuously tested throughout the development process. Hence, as shown in the case study, the PAM can be leveraged for information modeling of the targeted modular building systems.

As the schematic figure of the building illustrated in Figure 4-7 shows, the hierarchy of the components and properties in the PAM is consistent with the structure of the elements in this building. The dotted area in the schematic figure represents the site-built part of the building including foundation, podium, and concrete shear core, while
the rest is solid and represents the modular parts of the building including volumetric models and façade panels. The modular and site-built parts of the building are differentiated, respectively, by red and green colors in the plan of the building in Figure 4-7. A sketch of a module is shown at the bottom of this figure. The modules that are post bearing are made up of five module parts: roof, walls, ceiling, columns (posts), and openings. The fact that the walls of the module are not load bearing is reflected by setting the resistance and stiffness attributes of the walls equal to zero. The components and elements aggregating each of the module parts are specified right under the objects representing these assemblies. This validation test shows that the PAM contains a rich library of object and attributes to represent a typical modular building system, and indicate that the structure of the PAM is consistent with the hierarchy of assemblies, subassemblies, and elements in conventional multi-story modular building system.
Figure 4-7. Information model for the Student Village building developed in the case study for validation of the PAM
4.4. Overview of the PAM

In this section, the information included in each of these LoD layers of the PAM is discussed in more detail. In the LoD 100 layer, a modular building is divided into modular and site-built parts. Modular is the part that is prefabricated at the plant (factory) and shipped to the jobsite for erection. Site-built is the part that follows the traditional on-site construction method. Location, height and number of stories, plan layout, building orientation, and lateral loading information are the attributes defined for the building. Although the loading information belongs to LoD 350, but since it is related to an object in LoD 100 layer, it is located in this layer.

In the LoD 200 layer, the site-built and modular parts are broken down into their elements. Modular parts are defined to be composed of many prefabricated modules that could be either volumetric or planar. Weight, dimensions, limitations related to size and weight, manufacturing limitations such as available materials and possible wall or floor systems, position, and hoisting information are attributes of a typical module. Both planar and volumetric modules inherit these attributes from their super-type. Volumetric modules have side gaps as one additional attribute. Side gap is the space that exists between two adjacent modules after installation and may be used as a space for Mechanical, Electrical, and Plumbing (MEP) ducts.

Since this is a PAM for multi-story modular buildings, the only sub-type defined for planar module is the façade. In addition to the attributes that façade inherits from the “Planar Module” and “Module” objects, it has a special attribute named wind load that is
used for structural design purposes. The volumetric modules are decomposed to the following main five assemblies: floor, ceiling, wall, column, and opening. For more clarification, each of these sub-assemblies is shown on a volumetric module in Figure 4-8. The floor assembly is associated with two attributes including occupancy and live load. Since all occupied spaces need floor and there may be different types of occupancy in a module, occupancy attribute is assigned to the floor instead of the module. In addition, since live load is a function of occupancy, it is considered to be associated with the floor as well.

Figure 4-8. Breakdown of a volumetric module prototype into its main sub-assemblies
The site-built part is composed of many site-built components that could be foundation, podium, or structural core. Each site-built component needs to have position and layout information at this layer. In addition, height and number of stories along with the structural system should be specified in this LoD for the podium. Moreover, there are three enumerated structural systems associated with the structural core, including 2D braced frame, 3D braced frame, and shear wall. In 2D frame system, the core-frames are located between modules, and each frame provides a lateral support for modules just in one direction. Unlike 2D-frames, shear wall cores and 3D frame systems support the structure in both directions.

The LoD 300 layer connects the abstraction layer to the integration layer and contains many objects that inherit their attributes and aggregations from their super-types (parents) in the abstraction layer. Objects in this layer are used to aggregate some of the objects in the upper layer. In other words, many assemblies in the upper layer are decomposed by a number of objects in this layer. Each object in this layer could be part of more than one assembly in the upper layer. For example, while perimeter beam, middle beam, joist, slab, steel deck, sheathing, and insulation also decompose the floor part of the module floor, the same objects excluding slab and steel deck decompose the ceiling part of the module. The composition relationship in this layer does not mean that the part should necessarily be included in the assembly, but may be used as an option. In addition to the modular parts, general structural members of the site-built structures and façades also need to be defined in this layer.
LoD 350 mostly contains the output of the design-development stage, while LoD 400 mainly includes outcomes of the detailed design phase. LoD 500 is related to the as-built information and facility management. Since these two areas are out of the scope of this project, no information regarding LoD 500 are included in the PAM; but since the PAM is developed generic enough to be extended for other project disciplines, project areas, and purposes, the empty layer is included in the PAM.

4.5. Applications of the PAM

The developed PAM provides the multi-story modular building industry with a generic extendible information model that could be used as a basis for industrialized product design and lifecycle information management in projects. Some of the applications of the PAM are discussed in this section.

Similar to the proprietary and generic information models that exist in manufacturing industry, the PAM could be used as a foundation for information management of modular buildings. Currently, BIM is one the most novel information management systems being used in construction industry. To benefit from the capabilities of BIM technology for information management of modular buildings, one can integrate the PAM with available BIM frameworks and platforms such as information exchange standards and open standard file formats like Industry Foundation Classes (IFC). As per
this need, the authors used the PAM for creation of an Information Delivery Manual (IDM) for the information exchanges between architectural and structural design groups. The PAM provides the information unit repository needed for defining exchange requirements between different disciplines at different stages of the project. To do so, after identifying the activities and flow of the information in the process map, the exchange models, which represent information exchanges, are mapped to the PAM to specify the content of each exchange model defined in the IDM. The mapping resulted in an exchange requirement table, part of which is illustrated in Figure 4-9. Since the objects and attributes of the PAM are labeled based on the LoD, this specification also indicates the minimum LOD for different elements of a model at different stages of the projects.

Figure 4-9. Leveraging the PAM for defining exchange models required for structural design of modular buildings
The PAM could also be used for enhancing interoperability between the software serving modular construction. This could be done by using the PAM for defining concepts required for standardization of information representation in open standard file formats for semantic information exchange in these projects. As an example, in other parts of the research, the PAM along with the IDM that is developed consistent with the PAM are used for defining different concepts and property sets in a Model View Definition (MVD) developed for architectural BIM to structural design model information exchange. The data structure defined in this MVD follows the hierarchy of the objects and related attributes defined in the PAM. In addition, the MVD is designed to capture all the structural assemblies along with their structural attributes specified in the PAM. Since an MVD is already available for conventional site-built construction and some companies have implemented it, the MVDs developed in this research needed to be consistent with the concepts already defined in order to minimize efforts required for implementation of the new MVD. Hence, the current site-built structural MVDs are restructured in the MVD that the authors developed to address modular buildings’ needs. Although many of the concepts are based on the available MVDs, some new ones are also developed in this work to address special properties and assemblies of modular buildings defined in the PAM.

Application of the PAM as the basis for development of information exchange standards and specifications for different uses of BIM will integrate all these standards and minimizes inconsistency between such information exchange frameworks that address the same type of projects. In addition, the PAM acts as an integration platform
that could be used and extended for studying modular buildings in other areas and stages such as MEP design or facility management. Once extended, it provides a holistic information model that could be very helpful for coordination of modular building projects, especially between different disciplines. Better coordination and information management in complex projects (e.g., modular construction) could prevent numerous design and decision making errors that increase the cost and duration of the project.

In addition to the application of the PAM for information and product family design, it could be used for other purposes as well. For example, the PAM is an information resource for software developers to identify the features their software should support, and to make their tool applicable in multi-story modular projects. For instance, based on the assemblies defined in the PAM, software developer can add several components and elements to their drawing tools and libraries to make it possible to author a BIM model of a typical modular building in a reasonable amount of time. In addition, for the elements that are already supported in BIM software and are shared with modular buildings, the special attributes defined in the PAM for those elements could be added to the elements’ predefined properties in the software for better support of modular buildings. In addition to the software developers, the engineering companies benefit from the outcome by using modular technology in their projects. By illustration of the anatomy of the multi-story buildings along with specifying important attributes used in this project and showing different options for each functional component of building, the PAM could act as a technical, managerial, and coordination guide for the stakeholders of modular projects, especially those wishing to enter this market niche.
4.6. Summary and Concluding Remarks

During the past two decades, several types of modular systems have been used for construction of multi-story buildings around the world. Although the components and hierarchy of elements could be different in each of these systems, a mutual architecture exists in all of the common systems. This paper has identified the shared architecture of such modular building systems. Based on the identified architecture, a generic information model called Product Architecture Model (PAM) is presented in this paper. The PAM is also extended to capture the special assemblies and properties that may be considered in some of these systems. The PAM is an essential component for industrial design of such buildings and fills the gap that exists between the product-based nature of information models in the production industry and the project-based nature of information modeling frameworks in the construction industry.

As for any other industrial product, modular buildings are composed of assemblies, subassemblies, and elements. The PAM contains the hierarchy of product assemblies along with the associated attributes and properties of each component. In the PAM, objects and their attributes that are categorized based on Levels of Detail (LoDs), are mapped to their functionalities and limitations, and contain constraints and relations between different entities in an organized way. The PAM is designed to be generic to address different systems of multi-story modular buildings. In addition, the PAM is designed flexible to be expandable for capturing more information in a certain area. For example, the PAM presented in this paper is expanded to include the structural aspect,
which makes it usable for structural design purposes in addition to the general information it contains for design coordination among different disciplines. The generic and expandable nature of the PAM makes it an open information model and provides a shared information platform that could be used and updated by industry practitioners and scholars who work on different aspects of modular building. The hierarchy of the developed PAM is consistent with the way the building and corresponding modules are broken down into smaller sub-assemblies in the manufacturing plants. The data structure and capability of the PAM for modeling modular buildings are validated in this paper through a case study on development of an information model for a tall modular building (Student Village) using the PAM. In the case study, the PAM was evaluated by checking whether it is capable of capturing all the components and basic properties of the building. The outcome showed that the information model developed using the objects, attributes, and relationships defined in the PAM are capable of proper representation of the basic building information. The case study also indicated the structure of the PAM to be consistent with the hierarchy of the assemblies, subassemblies, and components of congenital modular building systems. In addition to the hierarchy, many LoDs are defined in the PAM to provide a common language for requesting a model with a certain level of development and a scale for measuring it. These levels follow the definition of the standardized LODs in the NBIMS-US™-Version 2 (NIBS 2012). In addition, UML language is used for graphical representation of the PAM. To enhance the flexibility of the information model for supporting new products and assemblies, the objects that belong below a certain level of detail are defined in abstract form.
The challenges in development of the PAM were mostly related to having a variety of modular system, proprietary nature of some of the systems, and diversity of different types of materials and assemblies that are used in these projects. There are several proprietary types of modular building systems to be used in these projects. In this research, the preference was to develop an information model that is capable of addressing the more conventional systems. In each of these systems, assemblies could be made up of several different types of sub-assemblies, materials, and elements. Hence, the PAM was designed generic enough to be capable of modeling different combinations and assemblies of the elements commonly being used in this industry. This is done by defining many abstract objects in the PAM to represent different components and assemblies sharing the same overall geometry and functionality. Consideration of several abstract objects minimized size of the PAM, as simplicity is one of the key characteristic of a properly designed operational information model.

The PAM addresses the modular building industry’s need for an information model as a base for product design and product information management. Similar to Manufacturing industry, the PAM could be used for development of product lifecycle and information management frameworks to increase productivity in modular building projects and to better meet owners’ demand. For example, the authors used the PAM for development of an information delivery manual for structural design of modular buildings, while the PAM was used as a repository for defining exchange requirements. The PAM provides information objects required for development of any other BIM information exchange framework required for different uses of BIM by different
disciplines. In addition, the PAM and its data structure are used by the authors for defining a Model View Definition (MVD) for structural physical and analytical representation of modular buildings in Industry Foundation Classes (IFC) file format. Similarly, the PAM could be used for development of several other MVDs for other project areas to address interoperability among BIM software packages and as a result facilitate implementation of BIM in these projects. Using the PAM as the base information model for development of IDM and MVDs related to different aspects of modular building projects results in a holistic set of information exchange standards serving this industry.

In addition to providing a basis for information management of modular buildings, the PAM provides a basis for designers to review different combinations of components with the same functionality through the lifecycle of the building, and select the most appropriate elements and assemblies based on their needs, criteria, and limitations. This will result in minimizing waste, optimizing the design based on demand, and coming up with a less expensive and high quality building. In addition, the PAM could be utilized for implementation of BIM in multi-story modular building projects. It clearly documents different aspects of the modular system that may be modeled in a BIM framework. Furthermore, the PAM specifies the information that may be required during the lifecycle of the modular buildings, and this makes software developers aware of the capabilities needed to be included in their software.
4.7. Acknowledgements

The authors wish to thank Tom O’Hara from Capsys Corps, John Erb from Deluxe Building System, Don Engel from NRB Inc., John Morrison from Mark Line Industries, and Cliff Cort from Triumph Modular for their review and devoted time during this research.

4.8. References


braced frames of modular steel buildings.” Journal of Earthquake Engineering,
Taylor & Francis, 13(1), 1–21.

concrete reinforcement supply chain.” Automation in Construction, Elsevier, 35, 1– 
17.

construction in Malaysia.” Journal of Architectural Engineering, American Society 
of Civil Engineers, 8(1), 19–23.

guide. Addison-Wesley Longman Inc., Reading, Massachusetts, USA.

state of the art report.” International Council For Research And Innovation In 
Building And Construction (CIB), CIB Publication No.329, Swiss, Switzerland.

construction sector perspective.” Supply Chain Management: An International 


Schoenborn, J. M. (2012). “A case study approach to identifying the constraints and barriers to design innovation for modular construction.” A Master of Science Thesis, Virginia Polytechnic Institute And State University, Blacksburg, VA, USA.


CHAPTER 5: Information Delivery Manual (IDM)

A Product-Oriented Information Exchange Schema for Multi-Story Modular Building Projects

Issa J. Ramaji\textsuperscript{1,2}, Ali M. Memari, Ph.D., P.E.\textsuperscript{2}, John I. Messner, Ph.D.\textsuperscript{3}

\textsuperscript{1} Department of Architectural Engineering, Penn State University, 104 Engineering Unit A, University Park, PA 16802, E-Mail: ixj111@psu.edu
\textsuperscript{2} Professor, Department of Architectural Engineering and Department of Civil and Environmental Engineering, Penn State University, 222 Sackett Building, University Park, PA 16802, E-Mail: memari@engr.psu.edu
\textsuperscript{3} Charles and Elinor Matts Professor, Department of Architectural Engineering, Penn State University, 104 Engineering Unit A, University Park, PA 16802, E-Mail: jmessner@engr.psu.edu

Abstract: During the past decade, Building Information Modelling (BIM) has significantly influenced the construction industry. Yet, none of the available BIM tools and frameworks focus on modular construction, thereby, implementation of BIM in these

\textsuperscript{2} Corresponding author. Tel.: +1 (814) 441 2495.
projects is relatively more expensive and tedious compared to site-built constructions. In this paper, an information framework developed for multi-story modular buildings is presented. The information exchange standardization methodology suggested by buildingSMART® International (bSI) is unable to capture the hierarchy of the building components, which is an important characteristic of modular buildings as manufactured products. Extensions to the IFC data schema are proposed in the Product Architecture Model (PAM), which is an information model for multi-story modular buildings. This paper shows how the PAM is integrated with other components of the framework to capture the information related to the industrialized production-based nature of these buildings.

**Keywords:** Modular Building, Building Information Modeling, BIM, Product Architecture Model, PAM, Information Exchange, Information Delivery Manual, IDM, Prefabrication.

### 5.1. Introduction

Prefabrication and modular construction in the Architecture, Engineering, and Construction (AEC) industry is a growing trend (Giles and Lara 2006). Based on the McGraw Hill Construction Report (2011), the size of this industry sector has expanded
by more than 50 percent in the five years from 2006 to 2011. Many aspects of the building can be prefabricated even if the design was not initially focused on prefabrication approaches (Haas et al. 1999). Prefabricated modules of buildings can be divided in two major classes: 2D or panelized systems, and 3D or modular systems (Badir et al. 2002; Nawari 2012). Panelized systems are planar assemblies that are often used as wall, roof, or floor building components. 3D modular systems are complete units that can be broken into many volumetric assemblies (Hallowell and Toole 2009; Ramaji and Memari 2013). Permanent modular buildings (made of volumetric modules) are more than 40% of the modular construction industry in North America, which has about 2.5 billion dollar of annual revenue. The rest of the market is for leasing of the relocatable buildings (MBI 2012, 2013).

There is more potential for modular buildings than what actually is in the industry to replace the conventional construction of buildings (McGraw Hill Construction 2011). Despite the simplicity of design and construction of modular single-family dwellings that brings about straightforward and speedy erection at the job site, the same cannot be stated for multi-story modular buildings (Ramaji and Memari 2015a; Solnosky et al. 2014). The construction methods and engineering issues of modular buildings become more complicated with increase in the number of stories and combining modular parts with site-built components such as shear walls or podiums. Some of the main challenges associated with multi-story modular buildings include: (a) complex structural and Mechanical/Electrical/Plumbing (MEP) system design of the modules and buildings; (b) need for considering fabrication limitation and preferences at the design phase; (c)
complex interactions between modules, supporting structures, and façade components; (d) complicated site and logistics management; (e) need significant collaboration level between different disciplines through all stages of project; and (e) lack of proper information management framework and software tools for addressing the aforementioned difficulties of modular building (Lu 2009; McGraw Hill Construction 2011).

There is potential for tackling these challenges by leveraging a successful technology and processes that are widely adopted in other sectors of the construction industry. The currently available advancements in computational technology can transform the way traditional projects are being performed (Jeong et al. 2009; NIBS (National Institute of Building Sciences) 2012). Proper information exchange and integration of different project phases are two fundamental needs within the construction industry. Since the modular multi-story building industry is still at a relatively early stage of development, there are limited codes, standards, and computer software to support the processes required to support the efficient design and construction of modular buildings. As a result of this shortcoming, integration level of the information in this industry is relatively low (MBI 2011).

Building Information Modeling (BIM) has significantly influenced the construction industry during the past decade. BIM is defined by National BIM Standard – United States® (NBIMS-US™) as “a computable representation of the physical and functional characteristics of a facility and its related project/lifecycle information using
open industry standards to inform decision making for realizing better value” (NIBS (National Institute of Building Sciences) 2012). Based on the McGraw Hill Construction Report (McGraw Hill Construction 2012), BIM adoption within the U.S. industry has surged from 28 % in 2007 to 71 % in 2012. Research has shown that the application of BIM in design stages could lead to better design efficiency, material selection, procurement, and coordination of the projects, and as a result has been shown to decrease the cost with an estimate of approximately 10 % in several case studies (Fallon and Palmer 2007). It has also been reported to reduce the delivery time by 20 to 50 % by improving the communication between stakeholders (Bayramoglu 2001; Gallaher et al. 2004). A McGraw Hill Construction Report (2012) estimates that BIM adoption in a project can increase the project profitability by 36% in the firms using this technology.

Leveraging BIM can significantly enhance efficiency and productivity in modular multi-story building projects, in a similar manner as site-built projects (Li et al. 2008). Implementation of BIM in an integrated project such as modular buildings requires a strong understanding between different actors regarding each other’s information requirements, and it requires a high level of interoperability between software (Aram et al. 2013; Singh et al. 2011). The development of an information framework could clearly identify information exchange requirements and is one of the very first steps to address interoperability issues to make it possible to use BIM in an industry sector (Jung and Joo 2011; Ramaji et al. 2014; Singh et al. 2011). The processes of multi-story modular building projects are different from those of site-built construction in many aspects. There are three additional stages in modular building projects that make related activities
and information exchanges different from the conventional construction projects: manufacturing, transportation/installation, and decomposition/reuse. Beside, although modular assemblies primarily use the same or similar components as site-built construction, a different production-based approach is taken in these projects, especially in design of the assemblies. To address these special characteristics of modular building projects, additional production-related information are needed to clearly communicate the production methods for these projects and should be addresses in the related frameworks and tools. Therefore, this niche area of construction industry requires focused attention to support BIM adoption on the projects (Nawari 2012). Song and AbouRizk (2006) identified the difficulties in the BIM implementation process itself, regardless of the software capabilities, as the greatest challenge for leveraging BIM in these projects. Because of these difficulties and inefficiencies, most small and medium sized modular companies cannot afford the time and team requirements for BIM adoption on their projects.

To address these challenges, an information exchange schema was developed for multi-story modular building projects. The framework was documented per the Information Delivery Manual (IDM) structure defined by bSI (Wix and Karlshoej 2010). As per NBIMS-US definition, which is developed by buildingSMART® alliance (bSa) as the ally of bSI in the United States, an IDM specifies the discrete processes in the targeted business case and provides a reference for expected outcomes, data requirements, and information exchanges within the projects. An IDM specifies the process which include information exchanges; why the exchanges are relevant; who
creates and consumes different information exchanges; what tasks will use the information; and how the software vendors should support the exchanges of this information (Eastman et al. 2010; Laakso and Kiviniemi 2012; Wix and Karlshoej 2010). In addition, an IDM supports technical implementation needs of the software vendors for the development of new BIM tools or extending present tools to support a specific BIM Use.

The IDM development methodology suggested by bSI defines exchange requirements based on the non-structured information units identified by means of surveys or using entities of the Industry Foundation Classes (IFC) schema. In this approach, the interaction between building elements and related information units is not considered. In modular projects, the hierarchy of components and related properties are fundamental information that are considered throughout the entire lifecycle of a building. As an example, the information units related to subassemblies of an assembly should be generated or requested after the assembly is defined in the design; or as another example, two components that are related to different systems of modular buildings may not be allowed to exist in a modular building at the same time. Thereby, definition of the exchange requirements needs to follow the product anatomy of building assemblies to yield a seamless and coherent information framework. To address this need in the information framework developed in this research, an extension to the bSI IDM methodology is proposed, which integrates the hierarchy of the information in modular buildings with the framework. The developed framework envisions the overall process of these projects with a more extensive focus
on the design phase. In addition, it is extended to capture detailed information exchanges required for structural design. In the following sections, potential benefits and recent efforts in the implementation of BIM in modular building projects are discussed, followed by an explanation of the extended methodology and description of the framework’s components.

5.2. BIM in Modular Building Projects

During past decades, many efforts have been made to evaluate benefits and challenges of implementing BIM in modular construction. McGraw-Hill Construction, in conjunction with National Institute of Standards and Technology (NIST), identified BIM technology as a major driving factor fueling interest in modular construction (McGraw Hill Construction 2011). Lu and Korman (2010) identified the following areas beneficial for the application of BIM in modular building projects: visualization, code reviews, fabrication/shop drawings, communication, cost estimating, construction sequences, and conflict and clash detection. Furthermore, Nawari (2012) states that the following aspects could be enhanced using BIM in modular projects: speed, sustainability, safety, constructability, quality, and shortening construction time. Additionally, Song and AbouRizk (2006) found that virtual modeling of the shop production environment at a realistic level have numerous benefits for managerial purposes, and BIM could be used as
a tool for this purpose. In the two case studies carried out by Lu and Korman (2010),
design coordination identified to be the best BIM use in modular projects. One of the case
studies was a 10,220 m² (110,000 SF) healthcare facility expansion. In this project,
$44,000 was spent for BIM implementation, while the gross saving was estimated to be
$220,000 at the end of the project (Lu and Korman 2010).

Mohsen et al. (2008) researched construction phase simulation of a modular
building in a case study, which was named “The Village”, comprising five modular
dormitory buildings located in Allentown, Pennsylvania. Due to a lack of a proper BIM
tools, a CAD model was used to optimize the installation of modules and site
management during the activities including site access analysis, lifting radius calculation,
and crane selection and positioning. The simulation resulted in accurate and proper
management of the resources and prevented any delays to the project schedule (Mohsen
et al. 2008). The authors felt that having an architectural building information model for
the project would have reduced the simulation time and cost, and would have increase the
accuracy of the model used for modularization.

Huang and Krawczyk (2007) prototyped a web-based BIM platform that provides
information filtering questions to assist customers during the selection of appropriate
design components. The main goal of the platform was to investigate the possibility of
integrating customizable modular house manufacturing with internet and new
information technology, in which the user customization is not just limited to finishing
material. In this prototype, using the prepared information model for individual modules,
all design suggestions were represented online, changes requested by clients could be reflected, and the final building could be virtually shown on the construction site (Huang and Krawczyk 2007).

Alwisy et al. (2012) developed a process model along with a supporting algorithm and tool for automated drafting and design of prefabricated elements of buildings. The tool that is developed in the research inputs the BIM and generates the shop drawings for the prefabricated parts. The designed algorithm goes through three main stages, including generation of the building information model, generation of modular construction manufacturing BIM, and detailed shop drawing generation (Alwisy et al. 2012).

Precast concrete elements are one of the simplest prefabricated building element. Eastman et al. (2010) researched the supply chain process of the precast/prestressed elements. Based on this research, adoption of BIM in construction projects and advancement in software capabilities for addressing different needs of precast concrete industry would result in many improvements in these projects. Examples are more efficient detection of clashes and constructability issues, reduced inconsistency in placing and shop drawing of elements, accurate cost estimation, and better logistic and site management (Aram et al. 2013). During the whole supply chain process of the precast concrete elements including design, detailing, fabrication and erection, BIM could be used as an integrating information platform for creation, collaboration, and decision making to improve the performance and productivity of the related projects (Aram et al. 2012).
Neelamkavil (2009) has researched the potentials for automating activities in the modular construction industry. He identified BIM as the most powerful potential platform for implementing automation into modular building projects. He categorized potential automation technics in modular building projects into three main areas: design, material handlings, and business processes. The material handling category is divided in two sub-categories, including prefabricated component manufacturing and components assembling process. The role of BIM is integrating information flow between the above mentioned categories. Other potential applications of BIM in modular construction identified in the research include: automated design, automated supply network and material management, robotic automation, automated construction site, virtual reality and simulation, automated assembly scheduling, and automated sensor-based control (Neelamkavil 2009).

McGraw Hill Construction Report (2011) discusses the results of research on current drivers and barriers for using modular technology in the building industry. Considering different BIM uses and their potential values, discussed in BIM Project Execution Planning Guide (CIC Research Group 2013), one can conclude that the following five out of seven drivers could be supported by BIM: 1) decreasing project cost, 2) reducing project duration, 3) better meeting owner’s demand, 4) increasing the quality of the product, and 5) increasing safety at the construction site. In addition, the following four out of top eight barriers for use of modularization could be tackled by using BIM: 1) misunderstanding of some disciplines about the information they have to deliver to other disciplines, 2) unfamiliarity of the project actors with the modular
building project design, 3) manufacturing and construction processes, and 4) possible
deficiency in quality of the modules and high cost of project due to potential
mismanagement and errors. Some of the major incentives for leveraging BIM in multi-
story modular building projects are as follows:

- **Improved Module Coordination:** Modular Buildings contain many complicated
elements, interfaces, connections, and detailing. Leveraging BIM can improve
coordination and offer a better understanding of the geometry of the modules and
their internal elements prior to manufacturing activities.

- **Reduction in Rework:** Since timing and sequences are very important factors in
modular building projects, especially in the installation phase, a single mistake could
be very costly and halt the entire project. Using BIM for reviewing the delivery and
installation process can improve the coordination and reduce the need for rework.

- **Reduction in Repetitive Errors:** Modular building construction is highly repetitive;
many modules are built based on identical design documents. As a result, occurrence
of an error in one part of a module can be repeated many times in other similar
modules. BIM, as an information management technology, can be used to avoid such
problems in these projects

- **Improved Engineering Analyses:** There is no special and widely accepted
computational or modeling tool for design and analysis of modular buildings.
Leveraging BIM in modular building projects can yield more accurate models, while reducing the cost and time of modeling efforts.

- **Improved Quality**: Fabrication of modules using high-tech machineries under controlled factory environment can bring about more precise fabrication of components and much tighter assemblies. Leveraging BIM in the projects for design of modules and generation of manufacturing models and documentation can help to take advantage of this potential, especially by implementation of model-driven fabrication.

- **Improved Efficiency through Module Repetition**: Due to different limitations related to manufacturing, construction, transportation, and as a cost-cutting choice, similar modules could be found in different projects. Therefore, the models developed for previous projects could possibly be used in a new project, which allows the BIM model to be reusable and lead to a decrease in the BIM modeling effort.

- **Improved Communication on Complex Projects**: Since modular building projects have two additional stages of manufacturing and transportation, these projects need high level of collaboration between different disciplines from early stages of the project and BIM can be a very powerful tool to facilitate communication.
5.3. Development of the Information Framework

Due to the product-based design perspective that characterizes modular building projects, the IDM development methodology suggested by bSI is modified to consider the building components’ hierarchy, interactions, and attributes. IDM is one essential part of the efforts coordinated by bSI for facilitating BIM information exchanges. bSI (formerly named International Alliance for Interoperability (IAI)) was initiated in 1994 to standardize ontology-based semantic and information exchanges to support construction industry (Nawari and Sgambelluri 2010). In the methodology presented by the bSI, the IDM has three main components including process map, Exchange Model (EM) specification, and Exchange Requirement (ER) specification. The overall processes of the project activities along with the information exchanges are specified in the process map. The EM specification consist of a set of tables describing several information exchange packages considered in the process map. Each EM represents one type of model typically being exchanges in a construction project. While this specification defines EMs in a plain-text narrative format, the ER specification defines them in the form of a table to specify the information needed for the model in an itemized format.

The methodology used in this research for development of the information framework utilizes the Product Architecture Model (PAM) of multi-story modular building for defining the exchange requirements. The PAM, developed by the authors as part of the research discussed, is a product information model capturing all the physical and non-physical objects and properties aggregate in modular buildings along with their
interaction. The components of the developed information framework and their relationships are shown in Figure 5-1. As shown in this figure, in addition to the EMs table, activities identified in the process map are also defined in a separate table set named Task Unit (TU) specification, while each of the tasks generating or consummating an EM is considered to be one unit. Although the EMs and TUs are in a narrative format but are defined based on the hierarchy and attributes described in the PAM. The objects and relating attributes defined in the PAM are used as a repository for development of the exchange requirements in the ER specification. In this specification, EMs from the EM specification and object along with their attributes from the PAM are crossed over to clearly define different exchange requirements. The proposed PAM, process map and Exchange Requirements specification are revised based on feedback from the modular building manufacturers and designers to come up with a validated, reliable and comprehensive model.

Figure 5-1. Anatomy of the IDM
As one part of the bSI requirement, the scope of every IDM that standardizes information exchanges in a specific business use case should be clearly defined based on the OmniClass Construction Classification System (OCCS) (Wix and Karlshoej 2010). OCCS organizes the information specifically designed for the construction industry from product to project aspects. It is made up of many hierarchical independently defined tables and each table classifies a different facet of the building construction information (OCCS2015). The scope of the developed information framework for modular buildings is specified in Table 5-1 in terms of the focus areas, project phases, and actors, based on Tables 32, 31, and 33 of the OmniClass, respectively. As shown in this table, the targeted phases are criteria definition, design, and coordination, while the actors are MEP team, architect, structural engineers, manufacturer, and construction manager. Design and documentation services are the two main focus areas of this framework.

In addition to defining the IDM scope based on the OmniClass system, the scope is also specified based on the Tetralogy of BIM (ToBIM). ToBIM, developed by bSa, structures different use cases of BIM through the whole lifecycle of the building. ToBIM breaks down the building lifecycle into 4 stages including design, procurement, assembling, and operation. Each stage contain 4 areas, while each of these areas includes four elements. Therefore, 16 BIM use cases are identified for each stage of the building lifecycle. The use cases that the developed IDM intends to address are listed in the following:
• Design → Architecture: General architectural design of the building such as layout of the building along with the size and location of the modules.

• Design → Structure: Structural design of the building from conceptual design to detailed design with focus of the modular part.

• Design → Enclosure: General enclosure design such as size, location, and weight of the enclosure panels.

• Design → Systems: General system design such as size and location of ducts and required space for running the pipes between or through the modules.

• Design → Quantity: Quantity take-off for the modules and their internal components.

• Procure → Selection: Selection of the building structural materials and enclosure types.

• Assemble → Acceptance: Acceptable tolerances for assembly of the structural elements and erection of the modules.
Table 5-1. The modular buildings IDM scope

**Focus Areas of the Use Case (OmniClass Table 32)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-41 00 00</td>
<td>Design Services</td>
</tr>
<tr>
<td>32-41 71 00</td>
<td>Designing</td>
</tr>
<tr>
<td>32-49 00 00</td>
<td>Documentation Services</td>
</tr>
<tr>
<td>32-49 11 00</td>
<td>Modeling</td>
</tr>
<tr>
<td>32-49 11 11</td>
<td>Inputting Project information</td>
</tr>
<tr>
<td>32-49 11 13</td>
<td>Creating BIM Content</td>
</tr>
<tr>
<td>32-49 11 15</td>
<td>Assembling BIM Content</td>
</tr>
<tr>
<td>32-49 11 19</td>
<td>Creating BIM Views</td>
</tr>
<tr>
<td>32-49 91 00</td>
<td>Updating</td>
</tr>
</tbody>
</table>

**Phases of the Use Case (OmniClass Table 31)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-21 11 00</td>
<td>Conceptualization Phase</td>
</tr>
<tr>
<td>31-30 00 00</td>
<td>Criteria Definition Phase</td>
</tr>
<tr>
<td>31-40 00 00</td>
<td>Design Phase</td>
</tr>
<tr>
<td>31-50 20 00</td>
<td>Coordination Phase</td>
</tr>
</tbody>
</table>

**Actors of the Use Case (OmniClass Table 33)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>33-21 00 00</td>
<td>Design Disciplines</td>
</tr>
<tr>
<td>33-21 11 00</td>
<td>Architecture</td>
</tr>
<tr>
<td>33-21 31 14</td>
<td>Structural Engineering</td>
</tr>
<tr>
<td>33-21 31 17</td>
<td>Mechanical Engineering</td>
</tr>
<tr>
<td>33-21 31 17 11</td>
<td>Plumbing Engineering</td>
</tr>
<tr>
<td>33-21 31 21</td>
<td>Electrical Engineering</td>
</tr>
<tr>
<td>33-41 00 00</td>
<td>Construction Disciplines</td>
</tr>
<tr>
<td>33-41 09 21</td>
<td>Manufacturing Services</td>
</tr>
<tr>
<td>33-25 00 00</td>
<td>Project Management Disciplines</td>
</tr>
<tr>
<td>33-25 16 00</td>
<td>Construction Management</td>
</tr>
</tbody>
</table>
The process of developing the information framework was divided in three phases: 1) process investigation; 2) product investigation; and 3) integration of the product and process information. The process investigation includes the development of the process map and related EM and TU specifications. The investigation of the product was carried out by developing the PAM. The identified information about the process and product were integrated in the ER specification. Each of these phases are discussed in the following sections.

5.4. Process Investigation

As bSI suggests, after defining the description and scope of a Use Case, development of a process map is the first step in development of IDM to recognize what activities take place, who the actors are, and what information will be exchanged between different actors. Process mapping is a step-by-step description of the activities performed to achieve a goal using a specific input and producing a defined set of outputs (Marrelli 2005). The process map was developed using the Business Process Map and Notation (BPMN). BPMN provides a graphical notation for illustrating different steps in a business process (OMG 2015).
5.4.1. Structure of the Process Model

The developed process map is divided in two inter-connected diagrams. The first (see Figure 5-2) specifies the overall process of design activities in multi-story modular projects. The second is the structural design process map (see Figure 5-3) which is an expansion of the content within the Structure Engineering swimlane in Fig. 2. The structural design process map is designed to be consistent with the Applied Technology Council (ATC) (ATC 2013) structural engineering business processes for conventional constructions (ATC-75), although extensions are made to capture special needs of multi-story modular buildings.

The overall process map includes two types of swimlanes. In the first, the swimlanes are color-coded, each representing activities carried out by a specific actor or a group of actors. The colors orange, green, dark blue, violet, and bright blue represent, respectively MEP designers, architectural designer, structural engineer, manufacturer, and construction manager. The swimlanes of the second type are not colored and hold some information exchanges related to the activities in the immediate upper or lower swimlanes. The process map includes three types of objects: Task Units (TUs), Exchange Models (EMs), and connectors, defined as follows:

**Task Units:** TUs are located in the discipline-specific (colored) swimlanes and represent a group of activities that take place by certain disciplines to provide a specific deliverable. The TUs are represented by color-coded round-corner boxes and follow the same color-code as the swim lines. The plus sign symbol that marks some of the TUs
indicates that the container TU is expanded in another process map, that is, the one developed for the structural design activities. In addition, the curved arrow symbol indicate that the container TU and related information exchanges may take place recursively in a loop to achieve an acceptable result through multiple iterations.

**Exchange Models:** Each TU requires input to get started, or it generates some information to be used by another TU. These inputs and outputs are organized in different EMs and are represented by data sheets in the process map. Each EM specifies the information that needs to be exchanged between two TUs. Similar to the TUs and swimlanes, the EMs are color-coded as well. Since each EM is always generated by the discipline that sends the information, an EM’s color is the color of the actor that sends this set of information.

**Connectors:** The connectors relate two objects and specifies the flow of the information by delivering different EMs to TUs as messages.
Figure 5-2. Design process for a modular building project
5.4.2. Process Overview

In the first step of the design process, the conceptual design must be developed by the architect according to the comments and feedbacks provided by construction manager.
and manufacturer regarding constructibility. In such review process, the project limitation and preferences are identified and discussed with the modular manufacturer after reviewing site conditions and available resources and equipment. Then, considering the contractor’s preferences and limitations, the manufacturer reviews the available resources and materials and investigates the transportation limitations to identify design preferences and limitation, and then discuss them with the architect.

Once a schematic design is developed by the architect, it would be shared with the structural and MEP groups for review. These two disciplines comment on the building schematic design or add some discipline-specific information to the model, and then ask the architect to revise the schematic design accordingly. These activities and exchanging information between architect and engineering disciplines may be repeated to come up with an acceptable solution. As depicted in Figure 5-3, the structural activities in this phase include defining the general structural system of the building to specify the main load bearing parts of the building for resisting gravity and lateral loads, defining the structural system of the modules to specify how the module’s internal loads and the imposed external loads are resisted by the module, and authoring a preliminary structural BIM model of the building with an appropriate LOD.

Next, in the overall process map, the architect develops the architectural design in detail and sends the information to the engineering disciplines. Based on this architectural model, the MEP and structural engineers develop their discipline-specific BIM models in more details, and send those to the architect along with their comments and request for
changes. These activities may take place repeatedly to come up with a common-agreed solution. The activities in design development phase of the structural design include generation of structural D/A model by exporting the BIM model to the analysis software, design of the structural detailing, and updating the structural BIM model with outcomes of these activities.

Afterward, in the overall process, the manufacturing and construction documents would be sent to the manufacturer and contractor to feed other phases of the project. In case of existence of any request for changes, the activities and information exchanges in the design development phase may take place iteratively to achieve an agreement.

5.4.3. Exchange Models and Task Unit Specification

As discussed in the previous section, many TUs and EMs are developed to represent different activities and information exchanges among different disciplines of the multi-story, modular building projects. To standardize the activities in different TUs, a table is defined for each TU. An example of a TU table is shown in Figure 5-4, where the table specifies 1) the project stages within which the activities take place, 2) the disciplines responsible for doing the task, 3) the input information required to get the activities started along with the title of the TUs that are responsible to provide those specific input EMs, 4) the output and information packages that are needed to be exchanged to other disciplines after completion of the tasks along with the delivery
information related to each output EM, and 5) a description of the activities contained in the TU.

[TU.A.2] Preliminary Design of Building Architecture

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Preliminary Design Phase (31-20 10 21)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Architecture (33-21 11 00)</td>
</tr>
<tr>
<td>Input</td>
<td>EM.MA.1 (Manufacturing Criteria)</td>
</tr>
<tr>
<td></td>
<td>EM.CA.1 (Construction Criteria)</td>
</tr>
<tr>
<td></td>
<td>EM.MA.2 (MEP Criteria)</td>
</tr>
<tr>
<td></td>
<td>EM.SA.2 (Structural Concept Model)</td>
</tr>
<tr>
<td>Output</td>
<td>EM.AS.2 (Architectural Concept Model)</td>
</tr>
<tr>
<td></td>
<td>EM.AE.2 (Architectural Concept Model for MEP Engineers)</td>
</tr>
<tr>
<td>Description</td>
<td>The objective of this task is to prepare the architectural model to a certain level that the structural engineer can start evaluating different structural systems. The building is broken down into separate modules in this level. In addition, in this task, the architect has to prepare the preliminary layout of the required core that is going to be used as a corridor, elevator shaft, staircases, etc.</td>
</tr>
</tbody>
</table>

Figure 5-4. Sample Task Unit Description

For better recognition, the titles of the TUs are coded. A TU name is formulated in the form of “[TU.X.Z]”, where TU shows that the table is related to a Task Unit, X represent the actor(s) of the task, and since an actor might be responsible for doing many tasks, another variable Z is defined in the TU name to represent the project phase within which the task is being carried out. The defined code for different actors and phases are shown in Table 5-2. The actors and project phases included in this code are classified based on the OmniClass system. An example is [TU.A.2], where “TU” means it is a Task Unit, “A” means the actor is the architect, and “2” means this task is being carried out at
the preliminary design phase. A TU may be made up of several sub-TUs; in this case, the project phase number in the sub-TUs names is expanded in lower levels. For example, [TU.S.2.1], [TU.S.2.2], and [TU.S.2.3] are all subtasks of the [TU.S.2] and take place in order of their phase number.

Table 5-2. The code name used for the EMs and TUs tables

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Symbols</th>
<th>Actors</th>
<th>Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C:</td>
<td>Construction Manager</td>
<td>0: Conceptualization Phase</td>
</tr>
<tr>
<td></td>
<td>M:</td>
<td>Manufacturer</td>
<td>1: Criteria Definition Phase</td>
</tr>
<tr>
<td></td>
<td>A:</td>
<td>Architect</td>
<td>2: Preliminary Design Phase</td>
</tr>
<tr>
<td></td>
<td>S:</td>
<td>Structural Engineer</td>
<td>3: Design Development Phase</td>
</tr>
<tr>
<td></td>
<td>E:</td>
<td>MEP Engineers</td>
<td>4: Coordination Phase</td>
</tr>
</tbody>
</table>

Similar to TUs, a table is developed for each EMs to define the properties of the information exchange packages. A prototypes of the EM tables is illustrated in Figure 5-5. As shown in this figure, an EM table includes some descriptive information including type of the EM enumerated from (digital) model and document, the project phase info within which the information is going to be exchanged, the disciplines involved in that specific exchange including senders and receiver of the EM, and a description of the information that needs to be included in the EM. Similar code name are used for naming the EM tables. An EM name is expressed in the form of “[EM.XY.Z]”, where EM shows that the table is related to an Exchange Model, X represents the sender of the information, Y represents the receiver of the EM, and since many EMs might be exchanged between
two specific disciplines in different phases of the project, another parameter $Y$ is defined in the EM name to represent the project phase within which the information exchange takes place. An example for the convention used for EM table is [EM.MA.1], where “EM” means it is an Exchange Model, “MA” means “M” (the manufacturer) is sending information to “A” (the architect), and 1 means information is exchanged at the criteria definition phase.

[EM.MA.1] Manufacturing Criteria

<table>
<thead>
<tr>
<th>Type</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Criteria Definition Phase (31-30 00 00)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Manufacturing Services (33-41 09 21) Architecture (33-21 11 00)</td>
</tr>
<tr>
<td>Description</td>
<td>This exchange model contains information about the limitations and preferences of the manufacturing group. These criteria can be related to the dimension, weight, and material selection. Satisfying these criteria will reduce the cost and time, and will increase the quality of manufacturing and transportation.</td>
</tr>
</tbody>
</table>

Figure 5-5. Sample Exchange Model

The developed process map contains 17 TUs briefly described in Table 5-3. It also includes 13 EMs for delivering information to these TUs. The list of these units along with a short description of each is shown in Table 5-4.
<table>
<thead>
<tr>
<th><strong>TUs</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>TU.A.0</td>
<td>The Architect works on the main design ideas and prepare a preliminary conceptual architectural model of the building based on the needs of the owner.</td>
</tr>
<tr>
<td>TU.C.1</td>
<td>The construction manager reviews the defined project, and by considering the present status of the construction equipment and available resources recognizes the limitations for the site-built parts of the buildings as well as general properties of the modules such as size and weight. In addition, storage of the modules in the construction site and their installation are reviewed and limitations are recognized in this task.</td>
</tr>
<tr>
<td>TU.M.1</td>
<td>Manufacturer reviews the defined project, and by considering the present status of the manufacturing facilities and available resources recognizes the limitations for the prefabricated parts of the buildings. In addition, the transportation of the modules from the manufacturing plant to the construction site is reviewed and limitations are recognized in this task.</td>
</tr>
<tr>
<td>TU.A.2</td>
<td>The objective of this task is to prepare the architectural model to a certain level that the structural engineer can start evaluating different structural systems. The building is broken down into separate modules in this level. In addition, in this task, the architect has to prepare the preliminary layout of the required core that is going to be used as a corridor, elevator shaft, staircases, etc.</td>
</tr>
<tr>
<td>TU.S.2</td>
<td>In this task, the structural engineer reviews the architectural conceptual design, estimates design loads, recognizes the most appropriate structural system for the building and modules, and investigates if the building needs a secondary load resisting system.</td>
</tr>
<tr>
<td>TU.S.2.1</td>
<td>In this task, the structural engineer reviews the architectural conceptual design, estimates design loads, and investigates different structural systems for the building to decide the existence of the secondary load resisting system for lateral loads, type of the foundation, and requests probable changes in the layout or other general architectural properties of the building.</td>
</tr>
<tr>
<td>TU.S.2.2</td>
<td>Based on the defined general structural system of the building in task TU.S.2.1, the internal structural system of the modules’ floor, ceiling, walls, bracings, etc. is defined in this task.</td>
</tr>
<tr>
<td>TU.S.2.3</td>
<td>The first structural building information model would be developed based on the Architectural Concept Model and the defined structural systems in tasks TU.S.2.1 and TU.S.2.1. This model includes some information about the general structural elements of the modules and site-built structures.</td>
</tr>
<tr>
<td>TU.M.2</td>
<td>The primary goal of this task is the preliminary MEP design and to recognize the location, path, and required space of the equipment.</td>
</tr>
<tr>
<td>TU.A.3</td>
<td>The objective of this task is to develop the architectural model to a certain level that the structural engineer can start modeling and design of the building’s structural members. In this task unit, the architect may revise the design based on the feedbacks from structural designer.</td>
</tr>
<tr>
<td>TU.S.3</td>
<td>In this task, the structural engineer performs detail analysis of the structure and completes the structural design of the building to the detail design level and production of structural manufacturing and construction drawings. In this task unit, the structural engineer may revise the design based on the feedback from architecture.</td>
</tr>
<tr>
<td>TU.S.3.1</td>
<td>In this task, after some preparation, the Structural Model is used for generating the Structural D/A Model. After importing the model, it would be modified if required to make it ready for structural design and analysis.</td>
</tr>
<tr>
<td>TU.S.3.2</td>
<td>Design of the general structural members of the building is the main goal of the building. At the end of this task, general information about the main structural members of the building, including sizes of the beams, columns, bracings, foundations, shear walls and required reinforcement at different part of the structure will be specified.</td>
</tr>
<tr>
<td>TU.S.3.3</td>
<td>The details of the structural members such as the connection details, steel member splices, and concrete reinforcement patterns are designed and specified in this task.</td>
</tr>
<tr>
<td>TU.S.3.4</td>
<td>Using the structural design data resulting from tasks TU.S.3.2 and TU.S.3.3, in this task, the Structural Concept Model is modified and updated to prepare the Structural Model.</td>
</tr>
</tbody>
</table>
The main objectives of this task are design and modeling of MEP equipment of the building.

Document preparation for manufacturing of prefabricated parts of the building along with their interfaces with site-built structures is the main objective of this task.

Document preparation for construction of site-built parts of the building along with their interfaces with prefabricated structures is the main objective of this task.

Table 5-4. Exchange Model (EM) descriptions

<table>
<thead>
<tr>
<th>TUs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM.AC/M.1</td>
<td>This exchange model contains the preliminary architectural conceptual design. It includes general information about the project such as site location, building layout, and number of stories that the Construction Manager and Manufacturer need to identify their preferences and limitations in the project.</td>
</tr>
<tr>
<td>EM.CM.1</td>
<td>This exchange model contains information about the limitations and preferences of the construction group. These criteria are related to the limitations of construction site, available equipment, and skilled workers. Satisfying these criteria will reduce the cost and time, and will increase the quality of construction.</td>
</tr>
<tr>
<td>EM.MA.1</td>
<td>This exchange model contains information about the limitations and preferences of the manufacturing group. These criteria can be related to the dimension, weight, and material selection. Satisfying these criteria will reduce the cost and time, and will increase the quality of manufacturing and transportation.</td>
</tr>
<tr>
<td>EM.AS.2</td>
<td>The objective of this exchange model is to provide the structural engineer with conceptual design of the building, desired breakdown of the building into modules, and preliminary layout of the core and other site-built structures for structural conceptual design. This exchange model can be sent iteratively to come up with an agreement between actors.</td>
</tr>
<tr>
<td>EM.SA.2</td>
<td>Reviewing the architectural conceptual design and sending feedback to the architect are the main objectives of this information exchange. Also, this exchange model contains information such as the structural limitations of the received architectural model, suggested structural system, possible required secondary structural system, suggestion for modification of the core and other site-built structures, and estimated location and size of the structural elements. This exchange model can be sent iteratively to come up with an agreement between actors.</td>
</tr>
<tr>
<td>EM.AE.2</td>
<td>The objective of this exchange model is to send architectural conceptual design, building’s layout and breakdown, spaces and occupancies information, and the building core layout to the MEP engineers.</td>
</tr>
<tr>
<td>EM.EA.2</td>
<td>This exchange model contains MEP engineers’ requirements, including the location, path, and a rough estimation of the required size of the ducts and spaces, along with the preferred material for the building components that may affect their design.</td>
</tr>
<tr>
<td>EM.AS.3</td>
<td>In this exchange model, the architect passes the developed architectural design of the building to the structural engineer for design development of the building’s structure. In this module, the architect has considered the suggestions and limitations of the structural engineer. This exchange model can be sent iteratively if there are still structural conflicts and feedbacks on the architectural model. Sending the revised architectural model is continued until no structural conflicts remain. In this exchange model, the architect has to inform the structural engineer about the related limitations of the other disciplines.</td>
</tr>
<tr>
<td>EM.SS.3</td>
<td>In this information exchange, the structural engineer generates structural D/A model by exporting information from the structural model. Depending on the model and the structure, the structural engineer may have to prepare model for being exported and modify imported information in the structural D/software to add/modify some information to make it ready for analysis and design.</td>
</tr>
<tr>
<td>EM.SA.3</td>
<td>In this information exchange, the structural engineer passes the designed structural model of building to the architect for review. This exchange model can be sent iteratively if there are still conflicts</td>
</tr>
</tbody>
</table>
between the revised structural design and architectural model of the building. Sending the revised architectural model is continued until no conflicts and feedback remains.

| EM.AE.3 | In this exchange model, the architect passes the designed architectural model of the building to MEP engineers for their design development activities. In this model, the architect has considered the suggestions and limitations of the MEP engineer. This architectural model includes occupancies, exact size and location of the available spaces for MEP equipment, size and material properties of the components that are important for MEP design of the building, etc. |
| EM.EA.3 | In this exchange model, the MEP engineers send the MEP model of the building to the architect to inform him/her about the exact location, path and size of the required spaces and ducts along with the specifications of the insulation, windows, and other related component and materials. |
| EM.SM.4 | This module contains fabrication model that the manufacturer needs for fabrication of the building’s structure, including geometry, acceptable tolerances, material information, connection details, designed concrete mix information, etc. |
| EM.SC.4 | This exchange model passes the structural model of the site-built part of the building, like foundations and shear walls, if any, to the construction manager. In addition, it contains the required information for installation of the prefabricated arts including connection details of the modules together, module’s hoisting method, connection of the module to the site-built parts, and acceptable tolerances. |

5.5. Product Investigation

Recognizing information units of a product is another essential part of the every information framework, which is addressed by development of an information model of the modular building called PAM (Ramaji and Memari 2015a). The proposed modular building PAM provides a comprehensive information database for future analysis and evaluation on the modular buildings. The PAM is an product-based information model that represent fundamental structure of the product and shows how its component interact to aggregate the product (Fixson 2005; Ulrich et al. 2011). A part of the developed PAM is shown in Figure 5-6, which illustrates a layered hierarchy of the assemblies, subassemblies, and components of multi-story modular buildings along with their attributes that may be concerned through the lifecycle of these buildings.
The PAM is layered based on the Levels of Details (LoDs) of the physical components and their attributes. Layering is carried out based on the definitions of Levels of Development (LODs) standardized in NBIMS-US. The NBIMS-US is presented and maintained by bSa. The NBIMS-US has categorized BIM information units in six LODs and has defined them by description of their characteristics. Layering the information units of buildings into several LoDs help to clearly specify what information unit should be included in a model to achieve a certain LOD. In the LoD 100, modular buildings are broken down into site-built and modular parts. The layer LoD 200, splits the modular parts into planar and volumetric modules. The LoD 300 decomposes the modules to several module parts including floor, walls, ceiling, columns, and openings. The module parts are broken down into their components in LoD 350. The layer LoD 400 adds some information such as allowable tolerances required for manufacturing of the modules. Information related to the facility management of these buildings are included in the LoD 500. In the PAM, attributes are not necessarily from the same LoD that their objects are defined in, as different aspects of an object may be concerned at different stages of the project.

Up to LoD 300, all components of the PAM are concrete objects. A concrete object represents a physical component that may be used in a modular building. In the higher LoDs, all the components are defined by abstract objects. Each abstract object represents a family of the components that has the same function and similar concerned attributes. Defining abstract objects for detail information minimizes the size the
information model; makes the PAM generic; and enhances its flexibility to support a wide variety of components with minor differences in their characteristics.

The methodology used for development of the PAM includes four steps as follows:

1) **Recognizing different modular systems**: The first step was to recognize the most conventional modular building systems. This is done by literature review, conducting interviews with different parties in the industry, and site-visits.
2) Developing hierarchy of the modular systems to the assemblies and subassemblies: modular systems were divided into the assemblies and subassemblies to allow for the development of different options for each component. The functionality of these components were also surveyed in addition to their composition.

3) Defining attributes of the components required during the lifecycle of the building: In this step, the required attributes of each component required for product lifecycle management were defined. An attribute could be a general attribute like the weight of one modules or a very detailed engineering attribute like the stiffness of connection within the modules.

4) Categorizing the components and related attributes based on the Levels of Detail: In the last step, the hierarchy, properties, and attributes resulting from previous tasks were categorized according to the definition of the NBIMS-US’s standardized LODs. In this task, different hierarchy and attributes were labeled based on their best fit to the NBIMS-US LODs description.
5.6. Exchange Requirement Specification

To standardize information exchanges in different phases of a project, the product and the process of making the product needs to be researched first, followed by defining of the exchange requirements. The hierarchy and information units of the product, which is a modular building, are organized in the PAM. Additionally, the process of the activities and information exchanges are specified in the developed process map and the related Exchange Model (EM) and Task Unit (TU) specifications. In the last step, the ER specification integrates the PAM and the process map to clearly define the information that should be included in each standardized information exchange.

Figure 5-7 shows one part of the developed ER specification table, which in complete form has an extensive number of rows that list all the objects and related attributes defined in the concrete layer of the PAM in separates rows, and contains different EMs defined in the process map in separate columns. In this structure, as shown in Figure 5-7, every cell of the figure is related to an EM and an information unit that could be an object itself or an attribute of it. These cells are used to specify inclusion of each information unit in each EMs.

Three types of inclusion are defined in the ER specification and are coded with three letters including “G”, “M”, and “T”, which stand for “Generated”, “Modified”, and “Transferred”, respectively. Leaving a cell blank means that inclusion of the information unit is not required in the related EM. Definitions for the codes are as follows:
• **Generated (G):** The information unit has been generated by the sender of the EM, but it has not been exchanged to the receiver.

• **Modified (M):** The information unit was generated and sent to a receiver, and now it may have been modified or old values may have been replaced by new values. In this case the receiver has to monitor these to modify his/her design according to these changes.

• **Transferred (T):** The information was generated and sent to a receiver and is now going to be exchanged without any changes. So, monitoring changes is not required in this type of inclusion.

![Figure 5-7. Sample Exchange Requirement (ER) specification](image-url)
Classification of the inclusion types by the above mentioned method has many benefits. It specifies the party responsible for authoring an information unit, the party authorized to change an information unit, when an authorized actor can modify an information unit, and when the receiver of the model should check an information unit for possible changes.

Since all objects and related attributes are categorized based on the LoD in the PAM, the LOD of each information unit in the ER specification could be measured at each stage of the project. The same color-code used in the PAM is used in the ER specification to show LOD of different information units. It should be noted that the satisfaction of the LOD in an information unit is regardless of the inclusion type. Using the LoD-layered information units for specifying information units contained in different EMs eliminates the need for a separate LOD specification table for each business use case. Furthermore, it shifts using LOD standards for defining required information from the final users to the framework and software developers, and in this way eliminates misunderstandings that may happen during interpretation from current descriptive LOD standards.
5.7. Uses of the Information Framework

The represented information framework with its unique characteristics could be used for many different purposes, some of which include:

- The proposed framework facilitates implementation of BIM in multi-story modular building projects by clarifying the information required by different disciplines at different stages of the project. The information framework specifies the model exchanges that should take place in multi-story modular building projects, as well as the content of these Exchange Models. This provides a basis for different project actors to generate and deliver information required by other actors at the right time.

- The exchange requirements defined in the framework could be used for standardization of semantic information exchanges in the design stage of multistory modular buildings. In Chapter 6, it is explained how the PAM is used along with the represented framework for developing Model View Definitions (MVDs) for architectural BIM to structural design/analysis model information exchanges.

- Since the developed information framework and the PAM are generically defined to capture different stages and disciplines of the multi-story modular building projects, beside structural design purposes, the framework could be extended to study other aspects of modular buildings such as facility management and MEP design. Once extended, it would provide an integrated information framework that holistically overviews different project activities in multi-story modular building projects.
• By demonstration of design considerations at different stages of the project, the framework can be used as a design guide by companies that intend to start using modular technology on their projects.

• The framework specifies the information that the discipline-specific software should be capable of supporting for modular constructions. This provides an information resource for software developers to update their tools in order to make them efficiently applicable in these projects.

5.8. Summary and Concluding Remarks

An information framework has been presented in this paper to identify the workflow of design activities in multi-story modular building projects, and to specify information flows between these activities. To address the product-oriented design approach in multi-story modular building projects, the bSI’s methodology for development of IDMs is extended to capture the hierarchy of building components and their related attributes. The bSI’s methodology is not designed to capture the relationship between different information units considered in the framework. Hence, this method is extended by adding a new component to the information framework, which is the Product Architecture Model (PAM) of multi-story modular buildings. The PAM encapsulates all physical components of different modular building systems along with their attributes. It
acts as a repository for the considered information units and contains the relationship and rule-sets that should be followed in the framework. As a result of extending the bSI’s IDM methodology, the anatomy of the modular building is comprehensively observed in the developed information framework. Integration of the PAM with the information framework provides a holistic information management framework that clearly specifies the information units of the product’s components along with the project stage in which they should be designed.

The developed information framework overviews the entire design process of these projects, and is extended to capture more detailed information for the structural design tasks and information exchanges. The generic nature of the developed information model and the PAM makes it possible to extend the framework to support other BIM uses in more details. In addition, since the encapsulated information in the PAM are labeled based on the Level of Detail (LoD), the Level Of Development (LOD) specification of the defined exchange requirements is embodied in the developed information framework, and if required, it could be used for defining a custom project-specific exchange requirement.

The information framework is composed of two main parts including process map and Exchange Requirement (ER) specification. The process map specifies an overview of the activities and the related information exchanges. Supplementary information about each of these activities and exchange information are provided in the Task Unit (TU) and Exchange Model (EM) definitions. The ER specification is the link between the
information exchange standard and the PAM. It clearly specifies the subset of the information units from the PAM that should be included in each EM.

In addition to facilitating execution of BIM in multi-story modular buildings, the proposed information framework outlines the basis for standardization of digital representation of building information models to address interoperability; provides a design guide for the companies intending to use modular building technology in their projects; and can be used as a resource for software developers to add required features to their software to support modularization.

5.9. Acknowledgements

The authors wish to thank Tom O’Hara from Capsys Corps, John Erb from Deluxe Building System, Don Engel from NRB Inc., John Morrison from Mark Line Industries, and Cliff Cort from Triumph Modular for their contribution to this research.

5.10. References

Computing in Civil Engineering 2012, American Society of Civil Engineers, June 17-20, Clearwater Beach, FL, USA, 221–228.


Construction, Bedford, MA, USA.


CHAPTER 6: Model View Definition (MVDs)

Extending the Current Model View Standards to Support Multi-Story Modular Building Construction

Issa J. Ramaji, Ph.D. Candidate¹, Ali M. Memari, Ph.D., P.E.²

¹ Ph.D. Candidate, Department of Architectural Engineering, Penn State University, 104 Engineering Unit A, University Park, PA 16802, E-Mail: ramaji.issa@gmail.com
² Professor, Department of Architectural Engineering and Department of Civil and Environmental Engineering, Penn State University, 222 Sackett Building, University Park, PA 16802, E-Mail: memari@engr.psu.edu

Abstract: Building Information Modeling (BIM) has significantly facilitated information management and collaborations in construction projects. During the past two decades, many Model View Definition (MVD) standards have been developed to enhance interoperability among different construction-related tools. As the focus of these standards were on conventional construction, they cannot efficiently support new emerging technologies such as multi-story modular buildings. Since modular buildings are not a completely different type of construction and are combined with site-built parts in many projects, the available standards need to be extended to support both modular and site-built constructions. In this paper, differences in data structures of these two types of construction are discussed and a methodology is presented for extending the current
MVD standards for capturing special needs of multi-story modular buildings. This methodology is clarified by providing an overview of the extension of the architectural to structural design/analysis MVDs for multi-story modular buildings.

**Keywords:** Model View Definition, MVD, Information Exchange, Information Delivery Manual, IDM, Product Architecture Model, PAM, Building Information Modeling, BIM, Modular Building.

### 6.1. Introduction

Modular construction is known for its economic advantages, sustainability, and high construction quality of the modules because of controlled factory construction environment. Initially, application areas of modular buildings were limited to single- or two-story dwellings based on wood-frame construction. Later, availability of alternative construction materials such as light-gage steel for the structural load bearing system in the modules made it possible to use this system for low-rise multi-story buildings. During the past decade, by integration of the modular system with site-built supporting structures such as shear walls, it became possible to use this technology in mid-rise and high-rise buildings. An example for high-rise modular projects in the U.S. is the 32-story Atlantic Yard B2 Building. The building will hold 363 market-rate and affordable-rate rental units and 4,000 square feet of retail space (Nonko 2013).
Modular building industry faces many challenges prior to achieving its real potential in the building construction industry. As a result of prefabrication, the construction phase in modular buildings is much shorter and simpler compared to conventional site-built methods. On the other hand, design activities in these projects are much more complicated due to elaborate detailing in the assembly of structural and nonstructural elements (Annan et al. 2009a). An example for these challenges is structural design of single modules and their interactions with other parts of the building (Lu 2009; McGraw Hill Construction 2011). Building Information Modeling (BIM) as a relatively new information management technology can be utilized to handle these complexities. BIM can potentially have many advantages for modular buildings at least as much as site-built construction (Nawari 2012). Due to special characteristic of multi-story modular building projects, implementation of BIM is potentially more economical compared to site-built construction projects. For example, these buildings are highly repetitive, and that leads to a decrease in the BIM modeling effort and speeds up the modeling processes. In addition, the module’s computer models developed for previous works could possibly be used in new projects with minor changes, which allow the BIM model to be reusable. Besides design complexities, high cost of rework, high occurrence chance of repetitive mistakes, and lower allowable manufacturing tolerances are some of the other modular building challenges that increase the justification to use BIM in these projects.

To benefit from different uses of BIM in a project, high level of interoperability is required between different BIM and engineering tools. From a technology standpoint, the
modular industry has been suffering from a lack of platform needed to improve interoperability and functionality (Annan et al. 2008). Currently, BIM and engineering tools developed for conventional construction cannot model and transfer all information units that are specific to modular construction, particularly information related to hierarchy and interaction of building assemblies (Annan et al. 2009b; Ramaji and Memari 2013; Solnosky et al. 2014). For example, information such as definition of a module as an assembly of module parts, and in turn a module part as an assembly of several elements may be lost during the information exchange process or cannot be modeled easily in the BIM tool in the first place. Location and consideration related to the module’s hoisting points designed for transportation and installation purposes is another example for such information.

To be able to properly transfer the information between say two different software products, both should be able to recognize the product, understand its scope, and use a common language for modeling information. To address this need, buildingSMART International presented Industry Foundation Classes (IFC) as an open standard file format to be used for digital modeling of the construction related information. IFC is a redundant language for information modeling (Venugopal et al. 2012b). For example a structural beam could be modeled using different entities including “IfcBeam”, “IfcStructuralCurveMember”, or any super-types of these entities. Moreover, it could be represented in many different ways such as the bounding box of the member using “IfcBoundingBox” entity, several boundary faces of the element using “IfcFace” entities, or a linear solid extrusion of the section of the beam using subtypes of
the “IfcProfileDef” entity. In addition, depending on the entities used for modeling of the element and its representation, different placement methods could be used. To devise an interoperable system, digital modeling of each information unit in the IFC language should be standardized. These standards are called Model View Definitions (MVDs) (Venugopal et al. 2012a). Nawari and Sgambelluri (Nawari and Sgambelluri 2010) define the MVD as the software developer interface of the exchanges. Since representation of information may be different for different BIM uses, several MVDs are developed for different uses of BIM (Cemesova et al. 2015; William East et al. 2012). Development of these standards are coordinated by buildingSMART International or its allies (Ufuk Gökçe and Umut Gökçe 2013).

So far, none of the developed MVD standards has been intended to support modular buildings. As a result, although modular construction can still benefit from these standards, by having seamless information exchanges for the elements it shares with conventional structures, the information that are specific to these modular buildings may be lost during the exchange. This issue could potentially increase the cost of BIM implementation in these projects, magnify complexity of BIM execution, and make the information management system error-prone. Developing a separate digital information exchange standard is not the best solution to address this issue for many reasons. First, most software developers have already implemented many of these standards and are reluctant to implement a new MVD standard for a specific type of structure. Second, multi-story modular buildings may have many site-built parts, and as a result need an information standard that could support both systems. Third, modular and site-built
constructions mostly use the same building elements and share too many common information units, thereby, a large portion of the model views and concepts defined in the currently available MVDs are usable in the prefabricated part of the modular buildings. Therefore, an alternative approach needs to be taken to address modular construction’s need for MVDs, and that approach is extending and enhancing the current MVDs.

In the following sections, first the information exchange standardization methodology presented by buildingSMART International is discussed. Next, identification of the modular construction’s special information units and interdisciplinary exchange requirements, developed as other parts of this research, are briefly discussed; the outcomes specify the information units that should be served in each MVD standard. Subsequently, the aggregation of objects in modular buildings is discussed and compared with the conventional construction, followed by presentation of a methodology for extending the current MVDs to support multi-story modular building projects. Afterward, the new model views and concepts developed for modeling special objects and information units related to hierarchy and properties of these buildings are discussed. Finally, the paper explains how the current structural MVDs are extended using the presented extending methodology to come up with MVDs capable of capturing special structural needs of multi-story modular buildings.
6.2. **Information Exchange Standardization**

To be able to exchange information between different software applications, the way that information is structured should be standardized. Compatibility of different software products for model transfer forms the basis of proper software interoperability. Gallaher et al. (Gallaher et al. 2004) conducted a study through NIST with the support of the department of commerce that looked at the cost of poor (or lack of) data transfer between software platforms. They quantified the cost of these inefficiencies in the U.S. at $15.8 billion annually across the following three cost categories: avoidance, mitigation, and delay. Mitigation costs associated with software and modeling capabilities include those related to losses due to reworking design file errors, verification of information in the models, and manually remodeling in multiple software platforms.

A standard in construction is defined by Vries (Vries 2006) as an approved specification of a limited set of solutions to actual or potential matching problems. Standards are developed to benefit the party or parties involved, balance their needs, and be used repeatedly or continuously during a certain period of time by a substantial number of target parties. An open standard is a non-proprietary and publicly available standard, which has many advantages for interoperability compared to direct translation. One advantage is decreasing the number of required translators. As depicted in Figure 6-1, with an open standard, it is not required to develop a different translator between every two individual units; we just have to develop a single translator for the open standard that will communicate with all units. Eliminating the need for having access to proprietary file
formats, handling software changes, and minimizing errors in translation by means of testing are other advantages of using an open standard format compared to direct translation (Bloor and Owen 1995; Gielingh 2008; Laakso and Kiviniemi 2012).

Figure 6-1. Direct Translators vs. Open Interoperability Standard

Two different approaches of minimalistic and structuralist (also known as explicit) are available for data exchange standards in IT (Laakso and Kiviniemi 2012; Vries 2006). The minimalist methodology is simpler and as a result could be adopted by the user community more easily. The process in minimalist approach is bottom up, i.e., starts with a small set of information and needs to be improved by experiments, testing, and iterative improvement, before being adopted. Once developed, tested, and adopted, the model would be richer than what is required (Tarandi 1998). The structuralist approach is more comprehensive and its development process is top down, i.e., first starts
6.2.1. **Standardization in Construction Industry**

To facilitate digital information exchange in the construction industry, the International Alliance for Interoperability (IAI) (later named buildingSMART) initiated standardization of ontology-based semantic and information exchange in 1994 (Nawari and Sgambelluri 2010). buildingSMART alliance (bSa), as the former U.S. chapter of buildingSMART International, developed and maintains the National BIM Standard-United States (NBIMS-US) to promote implementation of BIM in the construction industry and related software. The goal of NBIMS-US is to achieve an improved construction process throughout the whole lifecycle of the building using a standardized digital information model for different types of facilities (NIBS (National Institute of Building Sciences) 2012). Some of the efforts supported by bSa include creation of several information exchange standards, standardization of different Levels Of Development (LODs), and development of the BIM Execution Planning Guide (CIC (Computer Integrated Construction) Research Group 2013).

As depicted in Figure 6-2, efforts related to information exchange in buildingSMART are focused on three major areas: 1) Information Delivery Manual (IDM), 2) International Framework for Dictionary (IFD), and 3) Industry Foundation
Classes (IFC) data schema. IDM is a standard for the processes and information requirements in construction projects, IFD is a data dictionary and a standard for the terminology and data model for the information related to construction projects, and IFC is an open standard data schema for data management and information exchanges in construction industry. At the conjunction of IDM and IFC, MVDs are defined, which are the translation of the exchange requirements defined in IDM in the IFC language. In the following sub-sections, each of these concepts is briefly discussed.

Figure 6-2. Holistic Diagram of the efforts for information exchange standardizations in National BIM Standard – United States
6.2.2. Information Delivery Manual/Model View Definitions

IDMs and MVDs specify the information Exchange Requirements (ERs) and standardize how such information would have to be exchanged using IFC file format (Lee et al. 2013, 2015). They explain the exchange scenario in a human readable format, as well as in a computer interpretable way for software vendors to adopt and implement the standard (NIBS (National Institute of Building Sciences) 2012). IDMs and MVDs support the integrated construction processes by serving the technical implementation needs of the software vendors and provide role-based process workflow for the end user (Laakso and Kiviniemi 2012; Liu et al. 2013).

An IDM identifies and documents information requirement and data exchange in construction projects. These documents have to be expressed in human-readable form (Nawari 2012). IDM is an integrated reference for processes and BIM data requirements in a specific set of activities in certain types of projects. IDM specifies the information exchange process and its content and also identifies the authors and consumers of the information (Eastman et al. 2010; Laakso and Kiviniemi 2012; Wix and Karlshoej 2010).

MVDs integrate and standardize the ERs coming from one or many IDM processes to the most logical model views that will be supported by software tools. Implementation of these components specifies structure and format of data to be exchanged using a specific version of the IFC data schema. In other words, it standardizes organization of information exchange and shows how the information should be digitally modeled and exchanged using the IFC data schema (Gupta et al. 2014;
Representation of the building’s information units may be different in different uses of BIM, thereby, one special MVD is required for every BIM Use. An MVD standard is made up of several model views, each aggregated from a structured set of concepts. The model view’s concept may represent one or a set of relevant information units being called, respectively, leaf and adapter concepts.

6.2.3. Industry Foundation Classes

The IFC schema was developed by IAI to address the interoperability problems in building construction industry. Now, it is widely adopted by the industry and the software vendors, and is the main BIM open standard file format in the construction industry. IFC is a language for representation of building objects, their attributes, relationships, and inheritances (Laakso and Kiviniemi 2012; Nawari and Sgambelluri 2010). Use of IFC as the open standard file format for information management and exchange can enhance sustainable information modeling and seamless information exchange in the construction projects.

Using a layered model, the IFC schema takes advantage of both structuralist and minimalistic approaches (Tarandi 1998). The structure of the IFC schema is divided in four layers, including: Domain, Interoperability, Core, and Resource. These layers follow a restrictive hierarchy and each layer is independent from all the upper-level layers. The
Resource layer consists of a set of basic definitions such as geometry, measurement, cost, date and time, and quantity that could be used for describing objects in the higher layers. The Core layer includes the Kernel and Extension Modules and contains abstract concepts that are used for entity definitions in higher layers. The Kernel determines the model’s structure and provides basic concepts regarding objects, group, relationships, type definitions, attributes and roles. Core extensions, including process, product, and control, are specializations of classes defined in the Kernel. Space, site, building, building elements, and annotations are some examples for the abstract concepts defined in the core extensions. The Interoperability layer holds the building entities such as beam, column, window, occupant, flow segments that are shared among multiple construction applications. This layer provides the interface for domain models, thus providing an exchange mechanism for enabling interoperability across domains. The Domain layer is the highest level of the IFC data structure and contains entities for different Architecture/Engineering/Construction (AEC) domains or types of applications, such as architecture, structural engineering, and HVAC, among others. Footing, pile, plate, chiller, and boiler are examples of entities included in this layer (Laakso and Kiviniemi 2012; Liebich and See 1999; Liebich and Wix 2000; Wix and See 1999).

6.2.4. International Framework Dictionary

For any free information flow, three requirements have to be addressed: a format for information exchange, a planned process model, and a standardized description of
what the information actually is. buildingSMART International has addressed the last requirement by developing the International Framework Dictionary (IFD) library, which in simple term is a standard for a terminology dictionary (Bell et al. 2008; NIBS (National Institute of Building Sciences) 2012).

IFD is an open library that defines concepts and terms semantically and makes it possible to assign a Globally Unique ID (GUID) to each information piece in the IFC format. As a result, an exact discretion of a component can be correctly extracted by using a proper tool, as long as the correct GUID is given. For example, the project architect can define and describe a column in a language other than English, and then the structural engineer in the United States will be able to understand and extract the exact description of that column. While plain textual based information like names and descriptions are exchanged between the project actors, the related GUIDs are used by the computers to extract the required information. IFD provides a mechanism to develop a dictionary to connect the building information from existing database to a standard data model (Bell et al. 2008; Laakso and Kiviniemi 2012; NIBS (National Institute of Building Sciences) 2012).

Contents within the Data Dictionary can be divided in two different categories: 1) Subjects (Term): concepts that can be represented by a recognizable name and be distinguished from other concepts, and 2) Characteristics (Properties): concepts that need to be defined using a description. Examples of Subject concepts include door, window,
column, and door frame. Characteristic concepts include properties such as behavior, environmental influence, function, measure, and unit.

6.3. Investigation of Information Units

To identify the information units considered during the design process for multi-story modular buildings, the information exchange framework of multi-story modular buildings, which is developed as part of the research, is used (Ramaji et al. 2014). This framework identifies the processes and sequences of activities that modular buildings go through at different phases of projects and specifies the information exchange required among different disciplines.

Simulating physical buildings, digital models are made up of elements, assemblies, and systems. For a complex product, understanding the relationships between product’s components along with their hierarchies and attributes are fundamental requirements for development of any information framework (Sause and Powell 1990). This information was fed into the developed information exchange framework from the Product Architecture Model (PAM) of multi-story modular buildings as a reference for defining exchange requirements (Ramaji and Memari 2015a). The PAM is an information model capturing hierarchy of the components in multi-story modular buildings along with their interactions. To catch all information units that may be generated throughout the
lifecycle of these buildings, attributes of the physical objects were also identified and assigned to the related objects in the PAM. Moreover, to represent different Levels of Details (LoDs), the PAM was layered based on the definition of the Levels Of Development (LODs) standardized in NBIMS-US. This layering helps to clearly specify the importance of each information unit at each stage of the project. Overall description of the information exchanges is not adequate for precise definition of the exchange requirements and may cause misinterpretation. Hence, the proposed PAM was utilized for defining different exchange requirements to come up with an explicit information exchange framework. To do so, the buildingSMART International’s methodology for development of IDMs was modified to use the PAM as a repository for the information units of interest. The development steps of the information exchange framework is discussed in the following:

**Developing a process map of the design phases:** In this step, a process map for different activities of the project was developed. As NBIMS-US suggests, it was developed based on the Business Process Map and Notation (BPMN). This process map covers the lifecycle of modular buildings from early stages of design to manufacturing and construction phases. Additionally, in design areas, where the interpreted information exchanges are tackled for structural analysis BIM Use, the process map is expanded into more details by developing a sub-process map to catch all the sub-activities and related information exchanges.
Defining Task Units and Exchange Models that are recognized in the process map:
The process map contains many Task Units (activities) and Exchange Models (information exchanges). In this step, the activities taking place in these Task Units (TUs) and the information encapsulated in these Exchange Models (EMs) were clearly defined. The TUs definition contains five parts, including project stage, discipline(s), input EMs, output EMs, and a short description of the TUs. The definition of EMs contains the following four parts: type (document or model), project stage, involved disciplines, and a short description of the EM.

Mapping the Exchange Models to the proposed PAM: This step was needed to map the EMs to the proposed PAM and to come up with the Exchange Requirement (ER) specification. The ER specification specifies the information about physical properties or attributes of different assemblies or elements that have to be included in different EMs. While the PAM defines the information pieces about different aspect of the building and EMs describe the information needed to be exchanged at each stage of the project, the ERs specification crosses over these two to itemize the descriptive EMs and come up with a definite specification of the exchange requirements. Since the Levels of Detail are defined in the proposed PAM, the LOD specification of each EM is already embodied in the ER specification.

The resulting information exchange framework, which could be called an IDM in the buildingSMART International terminology, along with the PAM are recursively revised and improved based on the feedbacks received from the modular building
manufacturers and designers to come up with validated outcomes. The PAM identifies the information related to the modular building, while the developed information exchange framework specifies how such information should be generated and exchanged at different phases of the project. Since both of these explain the information exchange in a human readable format, the missing part for addressing interoperability in BIM information exchange is standardization of digital representation of the information that is to be exchanged. As suggested by Park and Holt (Park and Holt 2010), a properly formatted information exchange standard that can bring interoperability among multiple software products and tools, thus assuring compatibility, is an essential work that needs to be completed for any type of project. This need is addressed in this research by defining several model views and concepts to serve special objects and information units, while they could be used for extending current MVD standards. The outcomes are discussed in the following sections.

6.4. Development of the MVDs

The PAM indicates what the information units of the building are, while the information exchange framework specifies the sub-set of these information units that should be exchanged at each stage of the project among different disciplines. To make the digital information exchange possible, the way such information should be digitally
modeled in the IFC language would also have to be specified. Hence, MVDs are required to be developed based on the information exchange framework, and the PAM is to address this need. Due to modular nature of multi-story modular buildings, the hierarchical relationship among different components is a fundamental aspect of these buildings that should be considered in any supporting information standards or frameworks including the MVDs. In this section, the structure of the current MVDs is briefly reviewed, and the required extensions for making them usable in multi-story modular building projects are identified. To support these extensions, several additional model views and new concepts required for extending current MVDs are defined.

The simplified structure of site-built projects used in different buildingSMART International’s MVDs is shown in Figure 6-3. As shown in this figure, a project contains sites, sites contain buildings, each building is made of several stories, and each story includes many spaces and building elements. To reflect the modular characteristic of modular buildings, the structure of the modular construction’s MVDs should be created different from the current MVDs available for site-built constructions. In the data structure of modular buildings based on the PAM, a project includes sites, sites contain buildings, buildings contain building stories, which in turn are made up of modules, spaces, and building elements related to the site-built structure of the building or the connections that integrate modules together. In addition, the modules are made up of several module parts such as floor, walls, ceiling, etc., while each module part is made up of several building elements. Furthermore, in parallel, some of the high-level spaces such as building stories could be aggregated to form a building part. Comparing diagrams
depicted in Figure 6-3 and Figure 6-4, eight model views related to the following three types of objects are identified to be inserted in the current MVD standards to reflect the hierarchy of the multi-story modular buildings in the digital model: building part, module, and module part, which are discussed in the following:

- **Building Part:** This is defined as an aggregation of several space objects to form a zone that distinguishes different spaces of the building and the contained elements in terms of the construction method, construction time, or construction actors. In addition, this concept could be used for defining other types of differences such as thermal or lighting zone. “IfcSpatialZone” entity is used in the extended MVDs for representation of building parts, and since it is a non-hierarchical spatial decomposition, therefore overlapping is allowed between different zones. One model view is developed for this type of object, while the type of the building part is designed to be assigned as an attribute.

- **Module:** Module is a prefabricated component that is constructed in the factory and is delivered to the construction site to be installed. A module could be a prefabricated panel or a prefabricated volumetric module. Two model views are defined for this type of object, including “Volumetric Module” and “Planar Module” representing, respectively, 3D (volumetric) and 2D (Panelized) prefabricated modules.

- **Module Part:** Due to industrialized nature of modular buildings, each module is made up of many sub-assemblies that are delivering different functions in a module.
In addition, manufacturers divide modules into many sub-assemblies to enhance the manufacturing productivity. Module part is used to define these sub-assemblies. In the modular building, as defined in the PAM, the modules are broken down into five parts including wall, ceiling, floor, column, and openings. Five model views including “Module Floor”, “Module Wall”, “Module Ceiling”, “Module Column Assembly”, and “Generic Module Part” are developed based on this type of object.
Modular buildings contain many site-built parts, while a lot of different modules such as prefabricated walls or bathroom pods could be used in conventional construction. As a result, there is not a solid line between modular buildings and site-built construction. Therefore, the eight model views developed for extending the current MVDs are created in such a way that is applicable for both of these construction types and any combination of them. In the structure shown in Figure 6-4, if the three newly developed objects (building part, module, and module part) are eliminated from the diagram, the structure of the MVDs created for site-built construction, shown in Figure 6-3, would be obtained.

The elements and spaces of modular buildings and site-built construction are identical, while differences exist in a higher level of aggregations. Therefore, some model views and related concepts defined in the currently available MVDs that are related to spaces and building elements such as beams, columns, and stairs could be reused in the extended MVDs. There is a rich repository for different MVD standards that contains all the concepts that are developed so far. This repository provides most of the basic concepts required for development of new model views, but each new MVD standard addresses a different information exchange issue, and as the result needs several new concepts to do so. Therefore, 70 new concepts are defined in this study and used for development of eight model views identified to be required for extension of the current MVD standards to support modular buildings. These concepts represent the hierarchy of the discussed objects, aggregation of building components, and the special properties of the assemblies in modular buildings. Out of these 70 concepts, 38 are generic and 32 are IFC concepts. Generic concepts define information units independently from any digital
schema, while IFC concepts are defined to be used in information exchanges using IFC files. The name and code of the developed IFC concepts are listed in Table 6-1.

<table>
<thead>
<tr>
<th>Code</th>
<th>Concept Name</th>
<th>Code</th>
<th>Concept Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSU-003</td>
<td>Project Manufacturer</td>
<td>PSU-036</td>
<td>Transportation Route</td>
</tr>
<tr>
<td>PSU-005</td>
<td>Project Contractor</td>
<td>PSU-038</td>
<td>Transportation Time</td>
</tr>
<tr>
<td>PSU-007</td>
<td>Project Structural Engineer</td>
<td>PSU-041</td>
<td>Manufacturer</td>
</tr>
<tr>
<td>PSU-009</td>
<td>Building Part</td>
<td>PSU-043</td>
<td>Transporter</td>
</tr>
<tr>
<td>PSU-012</td>
<td>Spatial Zone Aggregation</td>
<td>PSU-045</td>
<td>Generic Nesting</td>
</tr>
<tr>
<td>PSU-014</td>
<td>Building Part Type Assignment</td>
<td>PSU-054</td>
<td>Element Part Aggregation</td>
</tr>
<tr>
<td>PSU-017</td>
<td>Module Attributes</td>
<td>PSU-056</td>
<td>Module Wall</td>
</tr>
<tr>
<td>PSU-019</td>
<td>Module Tag</td>
<td>PSU-058</td>
<td>Module Floor</td>
</tr>
<tr>
<td>PSU-021</td>
<td>Module Base Quantities</td>
<td>PSU-060</td>
<td>Module Ceiling</td>
</tr>
<tr>
<td>PSU-023</td>
<td>Module Dimensions</td>
<td>PSU-062</td>
<td>Module Column</td>
</tr>
<tr>
<td>PSU-024</td>
<td>3D-Module Specific Quantities</td>
<td>PSU-064</td>
<td>Generic Module Part</td>
</tr>
<tr>
<td>PSU-026</td>
<td>Perimeter Gaps</td>
<td>PSU-066</td>
<td>2D-Module Specific Properties</td>
</tr>
<tr>
<td>PSU-028</td>
<td>Module Common Limitation</td>
<td>PSU-067</td>
<td>Module Utilization Type</td>
</tr>
<tr>
<td>PSU-030</td>
<td>Mass Limit</td>
<td>PSU-069</td>
<td>Building Element Aggregation</td>
</tr>
<tr>
<td>PSU-032</td>
<td>Dimensional Limitations</td>
<td>PSU-084</td>
<td>Planar Element Part Base Quantities</td>
</tr>
<tr>
<td>PSU-034</td>
<td>Transportation Information</td>
<td>PSU-086</td>
<td>Planar Element Part Dimensions</td>
</tr>
</tbody>
</table>

Each of these concepts are documented in the standard template developed by buildingSMART International. As shown in the concept definition example depicted in Figure 6-5, the IFC concept documentation template has four main parts that are listed in the following:
• **Header:** This includes general information about the concept including reference code, version, and approval status, relationship with its immediate adaptor concept, history, author, and the document owner.

• **Usage in view definition diagram:** The section includes a diagram that shows the relations of the concepts with all their adapter concepts to the root variable concept, as is used in the MVD diagrams.

• **Instantiation diagram:** This section contains the instantiation diagram of the concept and specifies how different entities of the IFC schema should be used to represent a concept.

• **Implementation agreement:** The implementation agreement contains the business rules need to be considered for using the concept as defined in the instantiation diagram. It is in the form of several tables and each table is related to an IFC entity used in the instantiation diagram. These tables specify if an attribute of an entity must have a value, the type of the IFC entity that is allowed to be given to an attribute, the fixed value of an attribute to use, etc.
IFC Release Specific Concept Description (IFC4)
Project Manufacturer

Reference: PSU-003  Version: 1.0  Status: Draft

Relationships: Extends adapter concept "Project Actor Assignment"

History: Created 01.01.2015

Authors: Issa Ramaj

Document Owner: Penn State University

Usage in view definition diagram:

Instantiation diagram:

Implementation agreements:

IfcRelAssignsToActor
- GlobalId
- OwnerHistory
- Name
- Description
- RelatedObjects
- RelatingActor
- ActingRole

IfcActor
- GlobalId
- OwnerHistory
- Name
- Description
- ObjectType
- TheActor

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects — by The BLIS Consortium — All Other Rights Reserved

Figure 6-5. MVD Concept Documentation
The generic concepts are also documented based on the buildingSMART International documentation template that has three sections, where the first two are identical to the first two section of the IFC concept template. The third section is related to definition of the concept. Since generic concepts are not tied to any digital file language, their definition should be in a human interpretable plain text.

As mentioned in the previous section, the PAM includes several objects and many related attributes. The concepts used in the developed MVDs, including those borrowed and those newly defined, along with the developed model views serve these objects and attributes, while their relationships are based on what is defined in the PAM between objects and their attributes. The developed model views address the high-level assemblies (up to LoD 300) that are not supported by the current MVDs, while the attributes of these objects are assigned as leaf or adapter concepts. On the other hand, most of the low-level domain-specific information (LoD 350 and above) are represented by adapter concepts, while their properties are defined by several property- or quantity- sets.

Having model views reflecting the hierarchy of modular buildings along with their encapsulated concepts that represents special components and attributes of these buildings, one can extend each of the current MVD standards to support modular construction. To do so, the following three steps need to be followed to come up with a holistic extended MVD standard that addresses the needs of both modular and site-build constructions for the targeted BIM use:
1. **Updating high-level model views:** In this step, high-level model views of the current MVD standard have to be modified to be integrated with the developed modular building-specific model views. The typical high level model views of the current MVDs are “Project”, “Site”, “Building”, and “Building Story”, although some of these model views may not exist in some of the current MVD standards. Integration of these model views should be carried out in the current high level model view, using aggregation and containment relationship concepts that refer to the modular building-specific model views. For example, the “Building Part” model view should be integrated with the “Building” model view using a containment concept in the “Building” model view.

2. **Aggregating the low-level model views:** In this step, the low level model views of the current MVD should be integrated with the added modular building-specific model views. The integration should take place in the added model views, while the integrating concept depends on the type of the object being defined in the low level model view and its relationship with the object represented in the added model view. An example is aggregating the “Beam” and “Column” model views as the low level objects with the “Volumetric Module” as the added model view.

3. **Extending the modular building-specific model views:** In this step, the modular building-specific model views are extended based on the special information units and property sets required in the BIM Use the MVD is targeting. The eight developed modular building-specific model views only serve general properties, characteristics, and hierarchy of modular buildings. To use these model views for a certain BIM Use, some special concept may be required to be added to these model views. An example is the
structural lateral resistance of a module’s wall that should be added to the “Wall Module Part” model view as a property-set, to be used for structural modeling and analysis.

For better illustration of how this methodology could be used for extension of the current model views, the current MVD standards for structural design and structural analysis are extended to support modular buildings, and the results are discussed in the following section.

6.5. Extension of the Structural MVDs

The MVD extended in this research is targeted toward information exchanges from Architectural BIM to structural D/A model. Virtual Building Laboratory (VBL) proposed the following two MVDs for conventional site-built structures (BLIS (Building Lifecycle Interoperable Software) 2015): VBL-001 (2007) for exchanges from architectural design to structural design, and VBL-002 (2006) for structural design to structural analysis. The structural design model represents the physical model of the building, while the analysis model illustrates the topological representation of the building components suited for Finite Element (FE) analysis. Although these models violated some of the IFC schema’s constraints and a few of their proposed concepts became deprecated, most of the concepts are approved and formed the basis for the structural model representation in the IFC file format. To keep the model views of these
MVDs usable, they are redefined in this research and become updated based on the data structure of the IFC schema and as their deprecated concepts replaced with some valid ones.

Engineering analysis models of buildings, especially structural analysis models, are the equivalent discipline-specific engineering representation of buildings associated with the physical models. This correlation is even more pronounced in multi-story modular buildings due to the stand-alone hierarchical nature of the buildings’ spatial or physical objects. For example, a module is a complete structure that may have its own structural analysis model, which when aggregated with structural analysis models of other components and parts of the building form the model for structural analysis of the whole building. Hence, the two current structural MVDs are extended to one MVD in this research to serve both the structural design and structural analysis representation at the same time. Of course, sub-MVDs could be extracted from the developed MVD for separate architectural design to structural design and structural design to structural analysis information exchanges. Combining these two MVDs makes it possible to have the physical design model of the building correlated with its analytical model in one identical model.

The additional modular-building specific model views and concepts defined for reflecting general characteristics of these buildings lay the foundation for extension of the current MVDs. Additional modifications discussed below are also required to be made by
incorporating the methodology discussed to extend these MVDs to support multi-story modular building projects.

By conducting the first step, some of the high-level model views of these MVDs including “Building” and “Building Story” are updated to capture the special hierarchy of modular buildings. In the current MVDs, the “Building” is contained in the “Site”, while “Site” is contained in the “Project”. This remains the same in the extended MVD, whereas “Building” model view is updated to contain the “Building Part” in parallel with the “Building Storey”, while “Building Storey” is also defined to be contained in the “Building Part”. In addition, the “Building Storey” model view is also updated to contain “Planar Module” and “Volumetric Module”.

To carry out the second step, the “Building Part”, “Planar Module”, “Volumetric Module”, “Generic Module Part”, “Module Wall”, “Module Floor”, “Module Ceiling”, and “Module Column” model views are updated to be integrated with the low-level model views of the current MVDs. Several low-level model views were borrowed form VBL-01 for structural analysis: “Structural Curve Member”, “Structural Surface Member”, “Structural Point Connection”, “Structural Curve Connection”, “Structural Point Action”, “Structural Linear Action”, and “Structural Planar Action”. Since all this these model views are directly or indirectly related to the “Structural Analysis Model”, by assigning a “Structural Analysis Model” concept to each of the “Building Part”, “Planar Module”, and “Volumetric Module”, these low-level model views become integrated with the modular building-specific model views. In addition, several low-level model
views are borrowed from VBL-02 for Structural Design: “Beam”, “Column”, “Ramp”, “Slab”, “Stairs”, “Wall”, and “Building Element Proxy”. Integration of these elements with the added model views is addressed by defining an aggregation-type relationship between an abstract building element concept and the “Planar Module”, “Volumetric Module”, and the model views related to the building parts.

The last step involves extending the modular building-specific model views to capture information required for structural modeling and design. To do so, 14 generic concepts along with their equivalent IFC concepts are defined, while most of these concepts include a set of related single properties. The list of these IFC concepts is shown in Table 6-2. The extension for adding these concepts takes place in the “Planar Module”, “Volumetric Module”, and the model views related to module parts.
Table 6-2. List of the IFC concepts developed to support general aspects of modular buildings

<table>
<thead>
<tr>
<th>Code</th>
<th>Concept Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSU-047</td>
<td>Hoisting Port Nested in Module</td>
</tr>
<tr>
<td>PSU-049</td>
<td>Connection Port Nested in Module</td>
</tr>
<tr>
<td>PSU-070</td>
<td>Port Connection</td>
</tr>
<tr>
<td>PSU-074</td>
<td>Wall Resultant Structural Properties</td>
</tr>
<tr>
<td>PSU-076</td>
<td>In-Plane Lateral Load Capacity</td>
</tr>
<tr>
<td>PSU-078</td>
<td>In-Plane Lateral Stiffness</td>
</tr>
<tr>
<td>PSU-080</td>
<td>Vertical Load Capacity</td>
</tr>
<tr>
<td>PSU-082</td>
<td>Vertical Stiffness</td>
</tr>
<tr>
<td>PSU-088</td>
<td>Floor Resultant Structural Properties</td>
</tr>
<tr>
<td>PSU-090</td>
<td>Floor Occupancy Type</td>
</tr>
<tr>
<td>PSU-092</td>
<td>Floor Diaphragm Rigidity</td>
</tr>
<tr>
<td>PSU-094</td>
<td>Ceiling Resultant Structural Properties</td>
</tr>
<tr>
<td>PSU-096</td>
<td>Ceiling Diaphragm Rigidity</td>
</tr>
<tr>
<td>PSU-098</td>
<td>Structural Model Assignment</td>
</tr>
</tbody>
</table>

MVDs are composed of many model view diagrams, each representing a general variable concept in terms of several leaf and adapter concepts. Each MVD standard contains two separate sets of model views, including generic and IFC model views. Generic model views are made up of generic concepts and specify the information that should be considered in representation of a variable concept such as building element, spatial container or load. On the other hand, the IFC model views are made up of IFC concepts and specifies how the variable concepts defined in the generic model views should be represented in the IFC schema language. Therefore, since these two model view sets are equivalent, the number of model views in these two sets should be equal. The extended MVD developed for multi-story modular buildings contains 13 generic and
13 IFC model views in addition to those borrowed from the current structural MVDs. From these 26 model views, the ones that define “Project”, “Site”, “Building”, “Building-Storey”, and “Structural Analysis Model” are redefined to address their inconsistencies with the IFC schema and replace the deprecated concepts with valid ones. The rest are related to the modular building-specific model views discussed in the previous section. In addition to these model views, seven generic and seven IFC model views are borrowed from VBL-001 that are related to the structural finite elements and different types of loads. Furthermore, the same number of model views is borrowed from VBL-002 to add structural design modeling support to the extended MVDs. All the model views defined in VBL-001 and VBL-002 are listed in Figure 6-6, and the ones that are borrowed for development of the PSU-001 are related to it with arrows. In addition, all the modular building-specific model views along with those redefined from the current MVDs are listed in the right-hand side of this figure and are briefly discussed in the following:
Figure 6-6. Model views of VBL-001, VBL-02, and PSU-001 along with their connections
(a) **Project:** The model view related to this model object carries general information about the modular building project including the project name, owner, unit system used in the project, and the buildings contained in the project. In addition, it accommodates information about project partners including the architect, structural designer, contractor, and manufacturer.

(b) **Building:** The model view related to this model object is made up of information related to the building name, stories and other spatial containers of the building, breakdown of the building into different building parts, related structural analysis model, and placement of the building.

(c) **Building Storey:** The model view related to this model object includes information related to the name of the building story, spaces held in the building story, structural members and building elements contained in the building story, and local coordination of the building story according to the building placement.

(d) **Building Part:** The model view related to this model object provides information related to the name of the building part, type of the building part that could be modular or prefabricated, and the spatial zones that the building part is aggregated from. Since it is an aggregation of spatial zones, which have their own placement, placement information is not required in this model view.

(e) **Volumetric Module:** The model view related to this model object holds information related to volumetric module. Such information includes name and tag of the module, type name resulting from classification of different modules based on similarity, overall dimensions, perimeter gap existing between the external face of the module
and the connection surface, mass and size limitations, transportation route and time, manufacturer and transporter of the module, ports nested in the module that could be used for hoisting and/or connecting the module with other modules or site-built structures, information related to representation of the module in the digital model, placement coordination of the module, information related to the objects that are connected to the module at the defined ports, specification of the assemblies (module parts) aggregated to form the module, building elements to which the module decomposes and are not part of a module part, and structural analysis model related to the internal structural member of the module an its connection ports.

(f) **Planar Module:** The model view related to this model object comprises information related to planar module including name and tag of the module, type classification name, overall dimensions, information about the building system within which the module will be utilized, mass and size limitations, transportation route and time, module supplying actors including manufacturer and transporter, hoisting and connection ports nested in the module, information related to representation and placement of the module, information related to the objects that are connected to the module at the defined ports, building elements that aggregate the module, and the structural analysis model of the module and its ports.

(g) **Generic Module Part:** The model view related to this model object could be used for representation of a module part that is not one of the predefined volumetric sub-assemblies. The comprised information includes the name of the module part, its representation in the digital model, placement of the part according to the local
coordination of the module, and specification of the elements that aggregate the module part.

(h) **Module Wall:** The wall parts of a module could be represented by the model view defined for this model object. In addition to the information contained in the generic module part, this model view includes a property definition set that holds the overall dimension of the wall and resultant structural properties of the module such as vertical and horizontal stiffness and resistance of the wall assembly.

(i) **Module Floor:** The floor parts of a module could be represented by the model view defined for this model object. This includes a property set in addition to the information contained in the generic model view. This property definition set holds the overall dimension of the wall and resultant structural properties of the module such as diaphragm rigidity of the floor and occupancy type at the floor location.

(j) **Module Ceiling:** The model view related to this model object is developed for representation of a module ceiling part. In addition to the information that the generic module part model view holds, it contains a property definition set that includes the overall dimension of the ceiling and information related to its structural property such as the rigidity of ceiling diaphragm.

(k) **Module Column Assembly:** The model view related to this model object could be used for representation of a column part located in a model. A column part composed of the column and any related attachments such as connections, hoisting parts, guides for the installation, etc. Module column part contains all the information included in the generic module part model view.
(1) **Structural Analysis Model:** The model view related to this model object includes information required for structural modeling of a spatial container or a module. It contains name and type of the structural analysis model, structural members that the model holds, loads that are acting on the structural members, results of the analysis, and information related to the nested or nesting structural analysis model, if any.

### 6.6. Summary and Concluding Remarks

This paper, has discussed how current MVD standards developed for conventional construction could be extended to support multi-story modular buildings. Site-built and modular buildings are made up of the same building elements and multi-story modular buildings may contain several site-built parts. Hence, extension of the current MVDs for capturing special information related to modular buildings is identified to be a more reasonable solution as opposed to developing new separate MVDs for these projects. Taking this approach yields an extended MVD that is rich enough to support both site-built and modular constructions. In addition, since several tools have already implemented the currently available MVDs, this approach would speed up implementation of the extended MVDs.

The hierarchy of objects in modular and site-built buildings is different; modular buildings have three more types of objects: building parts, modules, and module parts. To
address these objects along with their related components and attributes, several model views and concepts must be defined to extend the current MVDs for modular construction. Based on the hierarchy of modular buildings in the Product Architecture Model (PAM) and information exchange requirements recognized in the framework of multi-story modular buildings, these model views are recognized to be the following: Building Part, Volumetric Module, Planar Module, Generic Module Part, Module Wall, Module Floor, Module Ceiling, and Module Column Assembly. To extend the MVDs based on the newly defined model views, three required modifications are identified: 1) Modifying the structure of MVD by relating the high-level model views of the current MVD to the ones added, 2) Connecting the added model views to the low-level model views of the current MVD, and 3) Extending the added model views for capturing the information related to the BIM Use targeted by the MVD.

Extension of current structural MVDs (e.g., VBL-001 and VBL-002) has illustrated how additional model views can be employed. The extended MVDs standardize representation of modular buildings’ structural models in the IFC language. It combines both structural design and analysis representations of the building and is developed in such a way that could be used in both modular and site-built projects.

The modular building-specific model views and the presented methodology could be used for extension of other MVDs available for other BIM Uses. Extending other available MVDs along with their implementation would enhance reliability of information exchange in modular construction and facilitate implementation of BIM in
these projects. Moreover, the MVD extension methodology discussed in this paper could be used for addressing interoperability needs of other emerging construction technologies.

6.7. Acknowledgements

The authors wish to thank Richard See from Digital Alchemy for his reviews and devoted time during this research.

6.8. References


CHAPTER 7:  Interpreted Information Exchanges (IIEs)

Interpreted Information Exchange: A Systematic Approach for BIM to Engineering Analysis Information Transformations

Issa J. Ramaji, Ph.D. Candidate¹, Ali M. Memari, Ph.D., P.E.²

¹ Ph.D. Candidate, Department of Architectural Engineering, Penn State University, 104 Engineering Unit A, University Park, PA 16802, E-Mail: ramaji.issa@gmail.com
² Professor, Department of Architectural Engineering and Department of Civil and Environmental Engineering, Penn State University, 222 Sackett Building, University Park, PA 16802, E-Mail: memari@engr.psu.edu

Abstract: Transforming a building information model to an engineering analytical model is tedious and time consuming. With the current state of interoperability, after
importing the building information model, the designer makes extensive modifications to prepare the model for analysis. This issue could be addressed by automating the model transformation process. In this paper, an automated interpretation mechanism referred to as Interpreted Information Exchange (IIE) is presented and compared with the conventional Direct Information Exchange (DIE). Moreover, a methodology is proposed for design and standardization of this type of information exchange to facilitate its implementation. Subsequently, application of IIE for engineering analyses is illustrated by explaining and validating a case study platform developed for interpreting the structural analytical model from the architectural model. Furthermore, based on the experience gained as a result of developing this tool, the considerations that need to be taken in the design of an IIE process are specified. Finally, the paper suggests how IIE can facilitate using Building Information Modeling (BIM) for engineering modeling purposes.

**Keywords:** Building Information Modeling, Interpreted Information Exchange, Engineering Design and Analysis Models, Model View Definition, Interoperability, Industry Foundation Classes.

### 7.1. Introduction

During the past decade, Building Information Modeling (BIM), as a growing technology, has significantly influenced the construction industry (Eastman et al. 2011; McGraw Hill Construction 2014). BIM is a digital representation of the building’s
physical and functional elements that can help better information management of the construction during its lifecycle (NIBS 2012). During the past few years, BIM adoption in the U.S. construction industry has significantly increased due to numerous potential benefits it has for different stages of the projects (AEC Magazine 2013; Fallon and Palmer 2007; Gallaher et al. 2004; McGraw Hill Construction 2012; Solnosky et al. 2014). CIC Research Group (2013) identified 15 primary BIM Uses from planning to operation phases of building projects. Engineering analysis is one of these BIM Uses that enhances the design quality, reduce time and cost of the design, and mitigate complex design activities. Using BIM for generating engineering design/analysis (D/A) models helps eliminating design errors and shortening the time needed for any modifications (Alwisy et al. 2012; Gane and Haymaker 2007; Sacks and Barak 2008). This is also consistent with what Li et al. (2008) mention in that we can possibly reduce redundant design activities by using BIM to visually interact with and run simulations. As CIC Research Group (2013) defines it, engineering analysis BIM Use is a process for determination of the most effective engineering methods using intelligent tools, design specifications, and building information models. Saving time and cost by automating the process and achieving better design, construction quality, and higher performance in buildings are the most significant values of using BIM for more rigorous and reliable engineering analysis (CIC Research Group 2013). As Schinler and Nelson (2008) state, departing from the traditional implemented process for engineering analysis, engineers can shorten the time they spend on modeling, design, and coordination with other disciplines.
It is evident that because of lack of proper interoperability, BIM users cannot take advantages of this BIM Use properly (Becerik-Gerber and Rice 2010; Guzmán and Zhu 2014; Ramaji et al. 2014). Based on the McGraw Hill Construction (2014) report, currently engineering analysis uses of BIM have a very low value/difficulty ratio. Energy and structural analyses, as the ones with the highest frequency index (represents how often BIM is used for a process), have negative value/difficulty ratio, which means it is easier to create the engineering model from scratch rather than generating it from the building information model (McGraw Hill Construction 2014). Different perspectives and methods are used for representation of building elements in building information models compared to discipline-specific engineering D/A models (Zhang and Issa 2013). This difference is one of the main reasons for the inefficiency and difficulties that currently exist in the engineering analysis BIM Use (Liu et al. 2010). Currently, information exchanges in this BIM Use are mostly limited to transforming the physical geometry information related to the elements. Even in geometry transformations, the physical representation of the elements might be different from their analytical representation; an example is small Mechanical-Electrical-Plumbing (MEP) holes in slabs that are usually ignored by the structural engineer. The geometry information might be enough for coordination and clash detections, but is usually not enough for generation of a model ready for analysis (Hassanien Serror et al. 2008; Polter et al. 2014). In engineering analysis, the performance and behavior of the building and its elements are examined. These evaluations are carried out by simulating different possible conditions such as imposing earthquake load or failure of an element due to a fire incident.
Therefore, more additional information is required for transforming a geometry model to an engineering simulation model (Nepal et al. 2013; Qin et al. 2011; Wang et al. 2013). Some parts of this additional information are related to the engineering modeling of the building elements, while others are related to the boundary conditions and loading definitions (Guzmán and Zhu 2014; Zhang et al. 2014). For example, after exporting information from the building information model to the structural D/A model, many enhancements should be made to the model to make it ready for analysis, including the following: definition of mechanical properties of the materials, dead and live loads assignments, definition of diaphragms at different stories, merging the elevation of different structural elements at the same story, connection type assignments, and earthquake load calculations (Nepal et al. 2013; Qin et al. 2011; Wang et al. 2013; Zhang et al. 2014). Another example is generating the energy analysis model from the architectural building information model, within which information such as R-value of the building’s enclosure elements, thermal properties of windows, and specifications of the HVAC system should be inserted to the model to prepare it for analysis (Guzmán and Zhu 2014; Kim et al. 2015).

To address this issue, a mechanism referred to here as Interpreted Information Exchange (IIE) is presented in this paper for automating the additions and modifications required for engineering models imported from the building information models. This mechanism functions in the information exchange process between different BIM and domain-specific tools, within which information needs to be interpreted by the domain’s engineers. As opposed to the approach taken in some BIM authoring tools and also in the
past by some CAD tools such as Intergraph's Micas (Hu 1993), there is no parallel linked geometric and analytical models in this workflow, but the information is being interpreted from the geometric model using some interpretation rule sets and look up tables. Having linked models in one platform does not address the interoperability between different software, and is not aligned with the open nature of BIM. In the following section, IIE is compared with the conventional Direct Information Exchange (DIE), followed by a discussion of different types of the information transformations that could be addressed by IIE. Similar to DIEs, IIEs also need to be standardized before being implemented in the industry, thereby a methodology is presented in the following sections for design and standardization of IIEs. Next, the IIE process is illustrated by going through a case study for automating the basic information interpretations from BIM to structural D/A models. The tool developed in this case study is also validated by using it for interpretation of the basic structural model of a test model to examine effectiveness of the automated IIE concept and the proposed design and standardization methodology. Subsequently, the experience gained through this case study is discussed in terms of several general considerations that need to be taken into account in any automated information interpretation process. Finally, different uses and advantages of adopting IIE are discussed along with its potentials for addressing some of the interoperability issues that currently exist in the construction industry.
7.2. Interpreted Information Exchange

Bazjanac (2008) identified information transformation necessary for a proper exchange of model data from an interoperable platform to a downstream application. Filtering the information included in the model to reduce size, translation of information to the format required in the downstream application, and adding new information to the output model are stated by Bazjanac and Kiviniemi (2007) as the main types of these data transformation. Therefore, the information exchanged between architecture BIM and engineering D/A tools can be divided in two categories: Direct Information Exchange (DIE) and Interpreted Information Exchange (IIE). In DIE, information is exchanged directly from the building information model to the engineering D/A model without any semantic changes; however, in IIE, the information in building information model is transformed for the engineering modeling purposes with some possible semantic changes. For example, exchanging the information of a single column from the architectural model to the structural model is a direct exchange. But breaking down frame elements at points where one element meets another element somewhere other than its end points, or simplification of a load-bearing light-gauge steel wall defined in the architectural model to a structurally equivalent simple area finite element in the structural D/A model are two cases appropriate for interpreted exchange. Another example for interpreted exchange is simplification of an enclosure panel that is made up of several parts such as framing, glass, and insulation to a simplified generic planar element with equivalent thermal resistance in the energy D/A model. Each IIE process may include several of such interpretations depending on the purpose of the information exchange and its level of
For more clarification, the difference between IIE and DIE is illustrated in Figure 7-1. The colored rectangles represent different types of models. The blue rectangle is the architectural building information model that forms the basic geometric model of the building. The light- and dark- green rectangles represent, respectively, the discipline-specific building information model and engineering D/A model. In the interpretation process, these two models should be differentiated from each other. The discipline-specific BIM is the model that includes all building elements that are related to a certain
building system such as structural or mechanical systems, whereas the engineering D/A model is a simplified analytical model of a building for a certain discipline. The orange areas in this figure, marked by “C”, represent the common information between architectural BIM, engineering-discipline BIM, and engineering D/A models. This information could be directly exchanged without any changes. An example for this type of information is the heights of the building stories that are equal in both BIM and D/A models. The shared information between architectural BIM and discipline-specific building information models are all of this type, but this is not the case for information exchanges from architectural models to engineering D/A models. In such information exchanges, some parts of the shared information need to be interpreted before being used. The light blue round area in this figure, which is marked by “I”, represents this type of information in the D/A model that should be interpreted from a subset of the shared information in the building information model. This subset is shown by black round areas and marked by “B” in this figure. An example is the space occupancy information in the architectural model that specifies the usage of different spaces in a building. Having this information and referring to the building code being used, the live load at different parts of the building could be calculated and assigned to related floor areas. In this example, the occupancy information is the subset of the shared information that belongs to the black round area, marked by “B”, and the calculated live load is the interpreted information that belongs to the light blue round area, marked by “I”.
There are two fundamental differences between DIE and IIE. Firstly, in DIE one piece of information about an object is translated to another language semantically. However, in IIE a set of information is translated to another set that may not have the same meaning as its initial meaning, but since it is being represented in a standardized format, it would be semantically equivalent for all engineering D/A tools. Secondly, in IIE some information from external references might need to be loaded into the interpretation engine to be able to interpret the required information and represent it in a more desirable format. For example, if we have a load-bearing Structurally Insulated Panel (SIP) wall in a building, by sending raw information from the architectural model to the structural D/A software, the interpretation engine cannot assign a stiffness or resistance to the wall, because these values are normally obtained from mockup testing. Therefore, some information needs to be loaded from external sources to make the IIE
effective, whereas in DIE all required information is contained in the input building information model.

IIE may be considered one kind of artificial intelligence in the information exchange process, where we need several rule-sets for implementation of the IIE (Bazjanac and Kiviniemi 2007; Bazjanac 2008; Won et al. 2013; Zhang et al. 2013). Each of these rule-sets interprets a certain information unit that is required by the receiver of the model. The exchange process in an IIE could be divided into several exchange Units (eUs), each responsible for delivering a set of related information units. Considering the aforementioned points, there are two types of eUs: interpretation Unit (iU) and direct-exchange Unit (dU). The iUs are the units that follow a certain interpretation process for extraction of the information, whereas the dUs are the exchanges resulting from filtration of the information contained in the input building information model. For example, in the IIE for generation of the structural model for a light-gauged steel wall enclosed in a steel frame, the process of exchanging information on physical elements could be divided into two units: exchange of information related to the perimeter frame, and the information exchange related to the wall itself. Since frame elements are not transformed into another form, the first exchange unit is a dU. But since the light-gauge steel wall is transformed to an equivalent structural shell area, it is an iU. The building information model of this wall and its schematic interpreted D/A model are shown in Figure 7-2. As shown in Figure 7-3, each iU has an input from the building information model, an output to the engineering D/A model, and an optional set of external input information as an interpretation reference. In the above mentioned example, the light gage steel is the input,
the shell area is the interpreted area, and the stiffness and resistance of the wall is the referenced external input.

Generally, the interpreted information may be included in the model in three different forms: assigning properties to the existing building elements, transforming the representation of the elements to another form, and simplification of the assemblies to equivalent simpler forms. Accordingly, iUs could be classified into the following three categories:

- **Enriching iUs**: In these iUs the building elements are not transformed to another form, but detailed engineering information are inserted into the model to make it ready for analysis. Assigning additional engineering properties to different building elements is one application of exchanging iUs. Examples include assigning R-Value to SIP walls or extending material property set of the structural materials to include their modulus of elasticity and strength in the analytical model.

- **Transforming iUs**: These iUs change the representation of a building element into another form. An example is transforming an “IfcBeam” element to an “IfcStructuralCurveMember” for generation of the structural D/A model. This transformation may be carried out on an element or any of its aspects such as the element’s shape representation. This type of iU is usually used in inter-disciplinary information exchange where different representation types are required for different BIM Uses.
• **Simplifying iUs**: These iUs transform a set of physical or spatial elements to a different equivalent set of elements. The iUs that interpret structural connection points between elements, transform space occupancy information to live load, or transform light-gage steel walls to a shell areas are examples of Simplifying iUs. Automating interpretation required in this type of iU is more complicated compared to Enriching and Simplifying iUs.

![Figure 7-2. An example for the IIE](image)

7.3. **A Methodology for Design and Standardization of IIEs**

BuildingSMART presented methodologies for development of Information Delivery Manuals (IDMs) and Model View Definitions (MVDs) as information exchange standards (Eastman et al. 2010; Panushev et al. 2010; Wix and Karlshoej 2010). These methodologies standardize DIEs in different areas of construction industry. The developed standards have significantly enhanced interoperability in the construction
industry and are to achieve seamless semantic information exchange among different construction software (Aram et al. 2013). Due to fundamental differences between IIE and DIE mentioned in the previous section, achieving the same status for IIEs in different engineering modeling areas is much more tedious. In IIE in addition to the exported file, both the external and BIM inputs should be standardized. Therefore, a special method is required to standardize the information flow and interpretation process. To address this need, in this section a methodology is presented for design and standardization of IIEs.

The first step in design of an automated interpretation process consists of studying the concept of the automated activity; then, detailing should be developed based on the designed concepts; and finally, the detailed design should be translated to machine language for implementation (Bazjanac and Kiviniemi 2007). The same process is considered as the core of the presented methodology. As shown in Figure 7-3, this methodology includes three main layers: 1) Concept Layer, 2) Generic Layer, and 3) Industry Foundation Classes (IFC) Layer. Each of these layers addresses one of the above mentioned automated process design steps. This is also consistent with the buildingSMART’s methodology for standardization of the DIEs, which includes: identifying the exchange requirements, developing model views, and binding the model views to the IFC schema (Eastman et al. 2010). The layers of the proposed IIE design and standardization methodology are explained in more detail in the following, and are illustrated by an example.
Figure 7-3. Relationship among the layers in the presented methodology for design and standardization of IIEs

7.3.1. Concept Layer

In this layer, the required information for engineering modeling of the building is divided into several eUs, including both direct-exchange and interpretation units. Each unit defines extraction of an information unit that may be required in a project. Breaking the required information exchange into different units, makes the units repeatable for other IIE design and standardization projects for different engineering BIM Uses. The modular nature of this methodology makes the implementation of the IIEs much easier, as it is in harmony with the current Object Oriented Programming (OOP) language usually used for automating the process. The dUs define simple filtration of the information and specify the information that needs to be queried and passed on to the export file, whereas the iUs specify much more complicated interpretation processes. As illustrated in
Figure 7-4, each iU defined in the concept layer includes an ID, a required input information set, and an output interpreted information set that generally specifies the function of each iU. The iU example named “iU-01” and shown in this figure is related to the structural D/A BIM Use and interprets the design live load that should be imposed on different floor areas based on the space occupancy information included in the input building information model. The whole designed IIE may contain many iUs but contains only one dU. This dU includes all the information units that should be passed on to the export file without any changes. The concept layer focuses mainly on recognition of information units within the engineering D/A model and the potential ways for extraction of such information from the building information model.

Figure 7-4. Declaration of an iU for extraction of the live load from space occupancy in the Concept Layer

7.3.2. Generic Layer

In this layer, the input and output sets of the eUs are expanded in more detail. The detailed input set specifies the information that should be fed into related eUs. The detailed output set specifies the information that is required in the engineering D/A model for proper engineering modeling of the building. Both the input and output sets of this
layer are made up of lists of building concepts, for which there is a rich repository available in different MVD development projects. These concepts could be used as references for defining the input and output sets for developing iUs and dUs. Since this is a generic layer, the input and output concepts could be independent from any standardized MVD concepts. As mentioned earlier, there are some iUs that need to input some information from the external sources but are not included in the input building information model. The external referenced input usually includes some engineering properties, code requirement, or parameters for simplification. The content of the external input for an iU depends on the method and algorithm used for the interpretation. In the Generic layer, the information in the external reference input would have to be itemized in detail and the references from which they should be acquired identified.

The input concepts list, for example “iU-01” shown in Figure 7-5 at its Generic Layer, shows that the required input includes information units related to slabs as load bearing elements, building stories as spaces that contain the slabs, and one of the property sets assigned to spaces that specifies the occupancy type. These concepts are color coded: light-blue color, also marked by “LB” symbol, means that the concept’s information is used for interpretation, but it will not exist in the exporting analytical model; dark-blue color, marked by “DB” symbol, means that the concept’s information is required for interpretation and is going to be passed on to the exporting engineering D/A model after the interpretation process. Since dark-blue concepts are included in the engineering model, they also appear in the output set, although they may be represented in another format. In case the representation changes, their colors are changed to light-orange and
marked with “LO” sign, unless they keep the dark blue color. There are also some dark-orange shaded items in the output set marked by “DO” sign, which are newly created interpreted concepts.

In the “iU-01” example, since the space occupancy information will not exist in the export file after interpretation, it is shaded by light-blue color, whereas the slab and building story are shown in dark-blue as they will continue to exist in the export model. On contrary, the color of the slab and building story is different on the output side of the iU. Since the information related to building stories is not expected to change during the interpretation process, the related concept is shown in dark-blue color. However, since slabs are expected to be transformed into structural shell areas, the slab concept is shown in light-orange color. Also, the live load concept that represents the distributed live load assigned to the structural shell area in the export D/A model is shaded by dark-orange as it is an interpreted information that is not contained in the input building information model.

The color-coding approach helps to differentiate the information units that need to be interpreted from the information units that are expected to be received directly from the input BIM file. It also helps to capture any changes in representation of the elements. In addition to the blue and orange concepts that show the input and output items, the green items appearing at the bottom of the iUs’ ID represent the itemized required information from the external references. As shown in Figure 7-5, the external information address should be clearly specified in this layer and the magnitude of the live
load should be calculated based on Table 1607 of the International Building Code 2012 (ICC 2012).

![Diagram](image)

Figure 7-5. Generic Layer representation of the iU-01

### 7.3.3. IFC Layer

This layer is the implementation layer of the proposed methodology, which interfaces with software developers to provide the information they need for automating the interpretation process. Consistent with the concept list in the Generic Layer, the IFC Layer specifies the IFC instances that should be generated and exported to the output file by the interpretation tool. It also indicates the instances that need to be transferred to the export file without any interpretation. Moreover, binding the input concepts indicates where the required information could be located in the imported BIM IFC file. In this layer, the concepts specified in the Generic Layer are bound to the IFC data schema. This could be done by borrowing concepts from the available MVD concept repository or defining some new concepts for this purpose. Borrowing the Generic Layer’s concepts from the repository eliminates the need for binding the generic concepts to the IFC
schema, as the bound form of these generic concepts are already available in the repository. In addition, it also facilitates the implementation of the iUs, because many of the available MVD concepts are already implemented in the BIM and engineering D/A software packages.

Since iUs are essentially defined based on the receivers modeling context such as common elements used for modeling, methods used for analysis, and the capability of the common domain-specific software, different iUs might be needed to be developed for different types of the buildings. For examples, the iUs used for IIE standardization of modular buildings for a certain type of engineering BIM Use may be different from the IIE standards of conventional site-built constructions or panelized structures (Ramaji and Memari 2015a). The object oriented methodology presented for IIE standardization makes the defined units reusable in design of other IIE standards and in this way integrates them.

To bind example “iU-01” to the IFC schema, Figure 7-6 shows how the space occupancy information could be acquired from the input BIM file. This information unit shown by symbol ❶ is located in an “IfcPropertySingleValue” instance that belongs to a property set named “SpaceProperties”, which is related to an “IfcSpatialStructureElement” element by an “IfcRelDefinesByProperties” instance. Since building story is a sub-type of “IfcSpatialStructureElement”, the binding is applicable to this or any other IFC spatial element. The diagram shown in this figure follows the color-coding approach used for defining the eUs.
Figure 7-6. Binding Space Occupancy concept to the IFC schema

Figure 7-7 illustrates the format that the live load interpreted in the “iU-01” should follow in the export file. In this figure, similar to the binding diagram of the space occupancy that follows the iUs’ color-code, the location of the distributed live load magnitude is shown by symbol ❷. The live load is represented by an “IfcStructuralLoadPlanarForce” object as part of an “IfcStructuralPlanarAction”
instance. This action should be related to “IfcStructuralSurfaceMember” by an “IfcRelConnectsStructuralActivity” instance. The “IfcStructuralSurfaceMember” is the new representation of “IfcSlab” instances from the input architectural BIM file.

Figure 7-7. Binding slab live load to the IFC schema
Having the third layer of the IIE standard, the interpretation process could be automated. The algorithm for translation of the machine interpretation process depends on the language and platform being used. The generic algorithm depicted in Figure 7-8 could be used for automating the interpretation process in the “iU-01” example. In this algorithm, first the building story spatial elements and their related instances should be queried. Next, based on the binding diagrams of the iU, the space occupancy information would be read where symbol ❶ in Figure 7-6 is shown. Then, the extracted occupancy type would be used for calculation of the distributed live load based on Table 1607 of the IBC 2012. Afterward, all slabs contained in the building story would be found and checked to see if they are represented in the “IfcStructuralSurfaceMember”; if not, they should be transformed to this representation format. Finally, the instances related to the interpreted information should be generated based on the diagram shown in Figure 7-7, and the amount of the live load be inserted in the location identified by symbol ❷.

Figure 7-8. Process of interpreting live load from occupancy information
7.4. Case Study and Validation

In this case study, the basic information interpretation required for generating the analytical structural model of a building from its architectural building information model is documented based on the proposed IIE design and standardization methodology, followed by developing a tool for implementation of the IIE and automating the interpretation process. Currently, the architectural models represented in Coordination View Version 2.0 (CV V2.0) are used as the importing information for generating the structural models. Due to differences between physical and analytical representation of building elements, connections between elements are not considered in CV V2.0; hence, the models imported from this model view are further discretized as necessary since most of the elements are not connected. The distance between joints of the connected elements is a function of the connecting elements’ dimensions. In addition to the connectivity issue, since the structural behavior of elements is not addressed in CV V2.0, the mechanical properties of the material required for structural design of the elements are not included in the model. To address these issues, an IIE is designed in this research to input information from the architectural building information model represented in CV V2.0 model view to interpret information required for structural modeling of the building, and to generate its analytical model in the structural topological view. Four iUs along with one dU are required for this interpretation process. These eUs are briefly discussed in the following:
• **Direct-exchange:** This unit is of dU type and transfers the information units that do not need to be interpreted. The information concepts considered in this unit are related to the general building objects including project, site, building, building-stories, and spaces.

• **Linear elements transformation:** This iUs transforms the IFC linear structural members including “IfcBeam”, “IfcColumns”, and “IfcMembers” to their equivalent finite element topological representation using the “IfcStructuralCurveMember” objects. In the topological representation, linear elements are modeled by a line with a cross-section and a material assigned.

• **Planar elements transformation:** In this interpretation unit that is a transforming iU, the IFC planar structural elements, including “IfcWall” and “IfcSlab” along with all their sub-types, are transformed to their topological representation using the “IfcStructuralSurfaceMember” objects. Each of these objects represents the planar elements in the form of a finite area element with a thickness and a material assigned.

• **Material information addition:** This iU enriches the model by adding mechanical properties of materials related to structural members. The iUs obtain the name of the material in the input model, extract their mechanical properties from a built-in table, and write it into the output analytical model. Considering such iU functions, it can be considered an enriching type.

• **Connectivity adjustments:** The element transformation iUs do the transformations based on the coordinates included in the input model without changing the coordinates to connect elements at their connection points. This simplifying iU is designed to
address this issue by importing the topological representation of elements and adjusting their coordinates for integration. In this process, the coordinates of each element’s end points are compared to other elements’ coordinates to see if the distances between elements are less than their original dimensions (i.e., height or width of the section of a linear element or thickness of a slab). If this is the case, the coordinates of elements are modified to represent the connection point. Since this process deals with coordinates of the points representing elements, it would be functional in connecting both linear and planar elements ("IfcStructuralCurveMembers" “IfcStructuralSurfaceMember” objects) at their connection points. The interpretation that takes place as a result of executing such iUs along with transforming iUs is an example of multi-stage interpretations, which is discussed in the following section.

These four iUs perform the essential interpretations required for the target objective, i.e., interpretation of a basic structural analytical model. Several other iUs may be added to this case study to further facilitate the interpretation of structural analytical models from building information models and minimize the post-processing activities after model transformation. Examples of such interpretations include adding loading information from space occupancy or transformation of wall assemblies into equivalent shell elements, as discussed earlier. Since evaluation of automated IIE process feasibility and presentation of design and standardization methodology are the main objectives of this case study, the scope of the developed tool in this research is limited to interpretation of basic structural analytical models.
To validate the developed automated IIE process and functionality of the automated IIE concept, generation of the structural analytical model of a building from its building information model is presented. The model is developed by buildingSMART alliance as a test model for validation of information exchanges related to building information models. The test model is a two-story office building presented in IFC CV V2.0 model view. The structural system of the building consists of steel moment frames with a concrete slab on the first floor. This test model contains the main types of structural building elements: planar elements such as slab, and linear elements such as beams and columns. The building information model of this building is shown in Figure 7-9 (top). In order to evaluate effectiveness of the designed interpretation process, the structural model of the building is generated using the following two different methods and the results compared: 1) generating the structural model using direct import function of a structural analysis software package, SAP2000® used in this case study; and 2) Interpreting the structural model using the developed tool, saving the model in an IFC file, and importing the interpreted file in the same structural analysis software. The structural analysis model generated directly from the IFC file without interpretation (the first method) is shown in Figure 7-9 (bottom). As a structural finite element model, the model shown in this figure is represented in topological view, within which linear and planar elements are shown, respectively, by lines representing elements’ center-lines and areas (shaded in red in this figure) representing elements’ mid-surfaces. As shown in this figure, most of the linear and planar finite elements in the structural analytical model are not integrated at connection points. In addition to connectivity issues in this model, the
elevations of the elements corresponding to the same story are also different. In order to make sure that these problems are not rooted in the import function of SAP2000®, the building information model is also directly imported using Scia Engineer software package (a structural design and analysis tool developed by Nemetschek), which results in the same broken model. To test functionality of the designed IIE process, the structural analytical model of the building is automatically interpreted using the developed tool (the second method). The interpreted model is saved in an IFC file and then imported in SAP2000® as presented in Figure 7-10. This figure illustrates how the automated interpretation mechanism can generate the correct structural model directly from the architectural model. The model in this figure shows the topological representation of the building’s structural system after running all the direct and interpretation units.

Considering current limitations of the available BIM authoring and structural analysis tools, validation of the interpreted structural model is performed by comparing the models before and after the interpretation process and checking whether the elements and materials information are transformed correctly. In other words, in order to validate the interpretation process, the model resulting from direct importing of the building information model into the structural analysis tool is compared with the model resulting from importing the interpreted IFC structural model. An alternative validation approach would be to compare the interpreted structural model with a structural model that may be generated by other software packages in a proprietary file format. Looking into the latter option revealed that none of the current software packages are able to generate a structural model from a building information model unless the model is drawn in that
specific BIM authoring tool or a user fixes the coordinates of points manually to integrate the elements with one another. This shortcoming is also one of the main incentives for development of the tool that is discussed in this section. Some of the BIM authoring software packages like Revit® generate analytical models in parallel with building information models when models are being drawn. Accordingly, in such a software package any model that is not drawn in that specific BIM authoring software will not have an analytical model assigned once imported. Some other tools such as Tekla® provide an environment for preparing an analytical model for a building information model. In such environment, nodes could be manually moved or merged by a user to come up with an integrated model. The analytical model could then be exported to different structural analysis software such as SAP2000®. In addition, in some structural analysis software packages such as Scia Engineer and SAP2000®, which are capable of importing IFC files, there is an option for merging all the points that are closer than a certain distance. Accordingly, the user is usually prompted to specify this distance while a model is being imported. In all of these alternative workflows for generation of the structural analysis model, the export model would be significantly affected by the decisions of the user who is manually drawing or modifying the analytical model. Therefore, due to this shortcoming and also to eliminate user errors during the validation process, the approach based on comparison of the structural models generated with and without interpretation and random investigation of the main interpreted information is chosen as the validation methodology in this case study. In other words, some interpreted information are selected randomly in the interpreted structural model and their values are
compared with the structural model generated directly from the building information model to make sure that no information is missing or transformed incorrectly. It should be noted that the coordinates of elements in the structural model resulted without interpretation are exactly the same as those of the building information model, because the import function of structural D/A software such as SAP2000® translates input information to their native format without any changes.

The interpreted model is evaluated in this research by checking seamless information exchange related to three main items: section properties of the elements, mechanical properties of the materials, and connectivity between elements. The model contains two materials: steel and concrete. Comparing the mechanical properties of these materials such as Module of Elasticity, Poisson Ratio, and Unit Weight in the interpreted model with the input IFC building information model showed that such information is exchanged correctly. In a similar comparison test performed on the section properties of elements in the interpreted model, all the 20 randomly-selected elements had the same cross-section assigned compared to the input model. Furthermore, to test the integrity of elements in the interpreted model, in addition to visual inspection of the connection points, the coordinates of the elements related to seven random connection points were studied in more detail. The result of this investigation for two of the connection points are shown in Table 7-1 and Table 7-2, where the coordinates of the connecting elements and related joints are presented. The connection point represented in Table 7-1 connects two columns, two beams, and one slab together. As this table shows, the elements related to each connection point in the interpreted model share the same coordinates at one of their
end points, which is the connection point. This point along with its coordinates is shown in bold-italic font in the table. As a result of such type of coordinate adjustment, the number of points in the model decreased from 380 to 139. The connection point documented in Table 7-2 is selected from the cases where a beam element is connected to the middle of another beam element. In this case, as coordinates in the table show, the beam with the connection point at its middle is divided into two elements at the connection point, and the other beam element is connected to the break point. As the result of such connectivity adjustments, the number of elements increased from 208 to 226, as many beam elements are divided into two separate beam elements at the connection point. Reduction in the size of exchanging model is another side benefit of the information interpretation methodology as much of the unnecessary information is filtered out throughout the model interpretation process. For example, in this case study, the size of the file reduced from 14.9 MB in the building information model to 212 KB in the structural analytical model.
Figure 7-9. Building information model of the test structure (top) and structural analytical model of the building generated directly from the IFC file without interpretation (bottom)
Figure 7-10. Structural analytical model of the test building interpreted from its building information model.
Table 7-1. Connectivity information of a corner connection point in the case study

<table>
<thead>
<tr>
<th>Frame</th>
<th>Joint I</th>
<th>Joint J</th>
<th>Length (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>155</td>
<td>156</td>
<td>5.63</td>
</tr>
<tr>
<td>F2</td>
<td>157</td>
<td>158</td>
<td>7.38</td>
</tr>
<tr>
<td>F3</td>
<td>49</td>
<td>50</td>
<td>5.27</td>
</tr>
<tr>
<td>F4</td>
<td>50</td>
<td>113</td>
<td>3.66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Area</th>
<th>Joint1</th>
<th>Joint2</th>
<th>Joint3</th>
<th>Joint4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>575</td>
<td>576</td>
<td>577</td>
<td>578</td>
</tr>
</tbody>
</table>

Model Before Interpretation

<table>
<thead>
<tr>
<th>Joint</th>
<th>X (m)</th>
<th>Y(m)</th>
<th>Z(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>0.00</td>
<td>-35.83</td>
<td>-1.00</td>
</tr>
<tr>
<td>113</td>
<td>0.00</td>
<td>-35.83</td>
<td>7.92</td>
</tr>
<tr>
<td>155</td>
<td>5.74</td>
<td>-35.83</td>
<td>3.87</td>
</tr>
<tr>
<td>156</td>
<td>0.11</td>
<td>-35.83</td>
<td>3.87</td>
</tr>
<tr>
<td>157</td>
<td>0.00</td>
<td>-35.69</td>
<td>3.94</td>
</tr>
<tr>
<td>158</td>
<td>0.00</td>
<td>-28.31</td>
<td>3.94</td>
</tr>
<tr>
<td>575</td>
<td>49.74</td>
<td>0.20</td>
<td>4.27</td>
</tr>
<tr>
<td>576</td>
<td>49.74</td>
<td>-36.02</td>
<td>4.27</td>
</tr>
<tr>
<td>577</td>
<td>-0.20</td>
<td>-36.02</td>
<td>4.27</td>
</tr>
<tr>
<td>578</td>
<td>-0.20</td>
<td>0.20</td>
<td>4.27</td>
</tr>
</tbody>
</table>

Model After Interpretation

<table>
<thead>
<tr>
<th>Frame</th>
<th>Joint I</th>
<th>Joint J</th>
<th>Length (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>220</td>
<td>187</td>
<td>5.86</td>
</tr>
<tr>
<td>F2</td>
<td>187</td>
<td>208</td>
<td>7.66</td>
</tr>
<tr>
<td>F3</td>
<td>211</td>
<td>187</td>
<td>5.27</td>
</tr>
<tr>
<td>F4</td>
<td>187</td>
<td>297</td>
<td>3.66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Area</th>
<th>Joint1</th>
<th>Joint2</th>
<th>Joint3</th>
<th>Joint4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Joint</th>
<th>X (m)</th>
<th>Y(m)</th>
<th>Z(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>0.00</td>
<td>-35.83</td>
<td>4.27</td>
</tr>
<tr>
<td>113</td>
<td>0.00</td>
<td>-35.83</td>
<td>4.27</td>
</tr>
<tr>
<td>155</td>
<td>188</td>
<td>49.54</td>
<td>-35.83</td>
</tr>
<tr>
<td>156</td>
<td>208</td>
<td>0.00</td>
<td>-28.17</td>
</tr>
<tr>
<td>157</td>
<td>211</td>
<td>0.00</td>
<td>-35.83</td>
</tr>
<tr>
<td>158</td>
<td>220</td>
<td>5.86</td>
<td>-35.83</td>
</tr>
<tr>
<td>575</td>
<td>248</td>
<td>5.86</td>
<td>-28.17</td>
</tr>
<tr>
<td>576</td>
<td>297</td>
<td>0.00</td>
<td>-35.83</td>
</tr>
<tr>
<td>577</td>
<td>185</td>
<td>49.54</td>
<td>0.00</td>
</tr>
<tr>
<td>578</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 7-2. Connectivity information of a middle connection point in the case study

<table>
<thead>
<tr>
<th>Model Before Interpretation</th>
<th>Model After Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectivity - Frame</td>
<td>Connectivity - Frame</td>
</tr>
<tr>
<td>Frame</td>
<td>Joint I</td>
</tr>
<tr>
<td>F1</td>
<td>181</td>
</tr>
<tr>
<td>F2</td>
<td>371</td>
</tr>
<tr>
<td>F3</td>
<td>309</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Joint Coordinates</th>
<th>Joint Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint</td>
<td>X (m)</td>
</tr>
<tr>
<td>181</td>
<td>28.31</td>
</tr>
<tr>
<td>182</td>
<td>21.58</td>
</tr>
<tr>
<td>371</td>
<td>24.01</td>
</tr>
<tr>
<td>372</td>
<td>24.01</td>
</tr>
</tbody>
</table>

Since some of the iUs input information result from other eUs, the execution order of these eUs is of importance. In the interpretation process, first, the dU is executed, followed by running the transforming iUs to generate the topological representation of the elements. Then, the connectivity adjustment iU is executed to integrate the elements. In the last step, the mechanical properties of the material related to the structural elements are added to the model.

Even if the IIE process designed in this case study is capable of transferring information from the geometric model to the structural analysis model, it cannot interpret the type of connection between elements (i.e. hinged, fixed, etc.). The reason is that the CV V2.0 does not contain any information related to the connections between elements.
Hence, the IIE designed in this case study considers all connection types to be rigid and leaves the modeling of such information to be performed manually by the structural engineering. However, in some cases this information could be interpreted indirectly from other information. As an example, if the type of a linear element is “JOIST.”, it means that this element should be connected to other elements at its ends only by hinged connections; so, a rule could be added to the connectivity iU accordingly to release the moment resistance at the end points of such type of “IfcStructuralCurveMembers” objects.

7.5. Implementation of the IIE Standards

The implementation method for eUs depends on characteristics of the platform used for this purpose; but there are some general considerations that are independent from the platforms being used and are essential for proper implementation of an IIE. Some of the important considerations discussed in this section reflect the experience gained by development of the tool for the case study.

As shown in Figure 7-11, in the automated interpretation workflow that uses the open standard file format, the IIE tool imports the IFC file, extracts and interprets the required information by implementation of an IIE standard, and exports the equivalent engineering analysis model in the IFC language. After importing the IFC file, the interpretation tool finds instances related to the input set of each iU and generates the
output information of all IUs. Next, such output information coupled with the instances directly passed on from the input file form the engineering analysis model in the IFC file format.

Before talking about the implementation of IIE standards as integrated systems made up of several eUs, we should first discuss implementation of individual eUs. In the following, some of the identified necessary characteristics of a properly designed eU are listed:

- Each eU should input a set of IFC instances or objects and export another set. It should be noted that since the one dU that exists in every IIE standard filters the object or instances from the input set to the output set, it essentially follows this rule.
- There should exist only one dU in an IIE standard, whereas there could be several iUs simultaneously. As instances of an IFC file are highly interconnected, having several
dUs may cause multiple transfers of directly exchanged instances or may break some of the relationships between the instances transferred from different dUs.

- Due to the modular nature of the IIE standards, each iU is best designed and programmed as an independent function from other iUs, although due to the Object Oriented Programming (OOP) nature of today’s programming languages, many shared functions could be used in these interpretation functions. Each iU can best be implemented in such a way that has the potential of being the only iU of an IIE if coupled with a proper dU.

- Transforming iUs needs to record the connection between instances related to the main and the transformed representations of elements. For example if a beam in the architectural building information model is to be interpreted as a linear member in the structural D/A model, the connection between the indexes of the related “IfcBeam” instance and the interpreted “IfcStructuralCurveMember” instance should be internally recorded in the program. This is important because there might be several iUs that need the same transformed representation of an element. Structuring the iUs in this way makes it possible to check whether the transformed representation of the element is already generated by another iU; and if this the case, the recorded connection could be used to access the related transformed representation. Then, the accessed representation of the element could be used for relating the interpreted information. This is essential for management of the IIE against duplications.

As discussed in the previous section, for the same type of engineering BIM Use, different IIE standards may be required for different types of buildings. However, since
the same approach would be used throughout the interpretation process of building elements and condition simulations, these IIE standards may share several common iUs. For example, for a special type of building, the building information model could be simplified into the form of a conventional structure; and then from there, the iUs of the conventional building could be used for generation of the engineering D/A model. As a practical application, in the IIE process for generation of the structural D/A model of modular buildings, first the modules could be assumed to be a completely isolated building; then, the analysis model of these modules could be generated using the iUs of conventional site-built structure; and finally all these separate structural models of the modules could be integrated into a single model using a special iU for modular buildings.

Although the IIE standardization methodology presented in this paper is modular, the execution order of these eUs is important as the output and input of different iUs and the dU may have some overlaps. As a result, while a standardized IIE may contain a selective set of eUs, the execution of these eUs should be managed considering duplications and input delivery. In the following, the importance of these considerations is discussed.

- **Duplication**: As discussed in the previous section, the output set of an eU is classified into four categories: directly exchanged information, new representation of the elements, enriching information, and simplified information. Since the one dU that exists in any IIE standard does all the direct exchanges, the implemented iUs should not exchange any information directly. In addition, since the enriching or simplified
information of an iU is its unique output, there should not be any overlap between the interpreted information of different iUs in an IIE standard. However, duplication management of the representation transformation of building elements is not as simple as the other two. To ensure that the changed representation of a building is not generated and exported multiple times, the iUs that conduct the same type of representation transformation should not be executed in parallel. It is better to define an iU for each type of representation change, otherwise the simplest iUs that include the representation changes can best be selected as the representation transformer iU and be executed as the first stage. The interpretation process then could be followed by parallel execution of the rest of the iUs at the second stage. This workflow for exception of the multi-stage interpretations is illustrated in Figure 7-12. The recorded connections between different representations of elements in the input BIM file and the output interpreted file facilitate the search for the transformed representation of elements. This connection record would be used at the second stage of the iUs execution to assign the interpreted information to the transformed element. Since iUs can best be independent from the others, they will be capable of doing all the required representation transformations independently, although they may not be required to use this capability as other iUs might set to execute it. Therefore, at the execution time, the iUs should be checked to see whether or not the required representation transformation has taken place; if not the case, then it must be transformed, otherwise use the already generated information. This process is shown in Figure 7-13.
- **Input Delivery:** Some of the interpretation for engineering analysis should be carried out in a multi-step manner, with each step made up of one or more iUs. Hence, the input for some iUs cannot be queried directly from the building information model and the referenced inputs. These iUs need the outputs of some of the other iUs for their execution. An example is interpretation from a shear wall with multiple middle...
columns to a shell area that is meshed right along the axis of the columns. In this example, first an iU is required to transfer the wall to a shell area ("IfcStructuralSurfaceMember") and several structural linear elements ("IfcStructuralCurveMember"). Next, another iU is required to mesh the wall based on the location of the columns to structurally integrate the meshed shells and the columns. In this case, the interpretation process needs to be broken into several sequential stages, while at each stage a set of iUs is implemented. The design of the sequence should be in such a way that provides proper information to each iU at the right time, unless the iU would miss a part of its essential inputs and may return unfit outputs. It should be noted that although some iUs need the output of other iUs, but it does not violate the independency requirement for the iUs. Multi-stage interpretation is in fact transformation of the model into another model that could be used for a certain type of interpretation. In spite of the fact that multi-stage interpretation increases the number of iUs in the IIE standard, it enhances the flexibility of the IIE and increases reusability of the standardized iUs for design of other IIEs.

As mentioned earlier, there are three types of iUs: Enriching, Transforming, and Simplifying. Although the considerations mentioned in this paper are applicable to all iUs, there are some minor differences in the overall workflow of each iU. Some of these differences are pointed out in the following:
• **Enriching iUs**: These units do not make any changes in available instances, they just add some informative instances and assign them to related instances. The relationship instances that are subtypes of the “IfcRelationship” are used for relating the enriching information to other instances.

• **Simplifying iUs**: These iUs input a set of instances related to certain assemblies and replace them with a new set of instances that represent the assembly in simpler engineering form. Since not all types of building assemblies are defined in the IFC schema, grouping the building elements that aggregate the assembly would be one feasible method for introducing an assembly to the Simplifying iUs. In this way, “IfcGroup” instances define the input set of the elements that for simplification purposes should be fed into the iU.

• **Transforming iUs**: These units replace a certain type of element or entity with another representation form and record the connection between the two instance sets related to both representation forms before and after the interpretation. The instances related to the transformed elements need to be integrated with high level objects of the model such as building or building story. This integration should be carried out using “IfcRelContainedInSpatialStructure” or any other appropriate instances.
7.6. **Potentials and Benefits of Implementing IIEs**

During the past decade, numerous MVDs have been developed to standardize information exchanges for different BIM Uses. Each of these MVDs standardizes the most desirable and reliable digital representation of building models in IFC language for a certain use. The number of these MVDs is increasing as BIM is being adopted more and more by the construction industry and its different sectors. These MVDs are not useful unless implemented by software developers. In this case, the users can generate the building information model in their desirable view format that could be one of these MVDs. Unfortunately, currently only one of these MVD standards (CV V2.0) is widely adopted by the BIM software. Implementation of all the developed MVDs is very expensive considering the increasing number of MVDs and their version changes. Since the majority of BIM authoring tools have implemented only the above mentioned MVD, other BIM tools that need to import BIM files as input have implemented the same MVD accordingly, although in many cases it is not their most desirable representation format. For example, the majority of structural D/A software packages can import or export the coordination view, but most cannot export the structural view. As a result of such framework, the geometry of some elements can be exchanged, while other important information such as the loadings and responses of the structure cannot be transferred.

The diagram shown on the left hand side of Figure 7-14 illustrates the framework currently used in the construction industry. In this figure, A, B, and C are different BIM tools that are connected together by several MVDs. It is anticipated that all BIM
authoring tools will be able to export and import models in all widely-adopted MVDs to make it possible for other BIM tools to choose their most desirable one. In addition to lack of implementation certificates, this is another reason most of the engineering D/A tools have implemented CV V2.0 instead of their own domain-specific MVDs. An alternative option for implementation framework of MVDs is having a Master-MVD standard (M-MVD) that is implemented by all BIM tools and contains enough information for extraction of models in other MVDs. In such framework that is shown in Figure 7-14 (right), IIE can play an important role in transformation of the models in M-MVD to other alternative MVDs. Since CV V2.0 has been widely implemented by software developers and is a generic discipline-independent model view, it has the potential for being used as the M-MVD. However, to use it for interpretation of models in other MVDs, it needs to be enriched in some aspects to provide adequate information required for interpretation. As previously explained in the Case Study and Validation section, the type of connection is an example of the information missing in the CV V2.0 for use as a M-MVD. Since, the CV V2.0 is not designed for this purpose, there are many other basic discipline-specific information missing in this MVD. Hence, to use it as a M-MVD, many additions are required to be made to this MVD, although as discussed in the case study, the current state of this MVD is rich enough to automatically interpret a significant amount of information required for engineering modeling and increase efficiency of such information exchanges. In addition, the capability of exporting custom user-defined MVDs, which are considered in major BIM authoring tools such as Revit® or ArchiCAD®, could be very helpful for including the required information in the input
models that CV V2.0 needs for use as a M-MVD. The tool developed in this research as a case study shows that this transformation is absolutely feasible, although considering some changes in CV V2.0 would make this transformation much easier, richer, and faster. The tool that Jiang et al. (2012) developed for generation of energy analysis models in Open Studio directly from building information models in IFC format is another example for using the Coordination View for extraction of analytical models related to other disciplines. Although this tool, which is developed on the BIMserver™ platform, converts building information models to the native Open Studio file format, it shows that the Coordination View is capable of carrying a considerable amount of domain-specific information that could be used for generation of the basic energy analysis model. However, further enrichment of this MVD would significantly increase the information interpretation process. This proposed framework has many benefits, some of which are listed in the following:

- The models generated by interpretation from the M-MVD could be much richer than the analytical models generated directly from the building information model. The interpretation can automatically add some additional information or make some modification through the interpretation process, and as a result, it can minimize the changes that the user of the file has to make to prepare the model for the intended use.
- This framework significantly shortens the number of MVDs needed to be implemented by software developers. Testing and licensing implemented MVDs is one of the most important needs to ensure the exchange models are reliable and seamless. The proposed model significantly decreases the number of engines that
generate models in different MVDs. It limits the export engines to a few (mostly one or two) per each BIM tool and one interpretation engine per each MVD standard.

- This framework makes it possible to implement both the DIEs and IIEs. The only difference between these two in this framework is that the DIE transformation of the models in M-MVD would contain only transformation of iUs, whereas an IIE may contain all three types of iUs.

- This framework makes it possible to shift the BIM manipulation/data-extraction tools from add-ons of BIM authoring software to much more independent tools that can import M-MVD IFC files from all the BIM software. These tools could be developed as IIEs and be used to combine with the MVD transformation DIEs.

- In this framework, the newly developed MVDs could be implemented and adopted by the industry much faster and easier. Since the BIM authoring software packages are not required to implement the newly developed MVDs, its adoption needs only a tool that transforms the models in the M-MVD to models in the newly developed MVD and a software package that could import and use these transformed models. In other words, this framework shifts the responsibility of adoption and implementation of different MVDs from the general-purpose BIM authoring tools to discipline-specific software packages that use these MVDs.
In such a framework, the engineering D/A software packages would be capable of exchanging information with each other and with BIM authoring software. Figure 7-15 shows the connections between BIM authoring and engineering D/A tools using only two sets of MVDs. To provide the analytical model for the D/A tool, the BIM tool exports the model in the M-MVD format. Then, the interpretation engine transforms the model to a desired discipline-specific analytical MVD format and makes the required additions and modifications. As shown in this figure, the connections between different engineering D/A tools are made using the discipline specific analytical MVDs. Therefore, in this framework, the BIM authoring tools need to be able to export and import IFC models in
the M-MVDs format, while the engineering analysis tools should be capable of importing and exporting models in the related discipline specific analytical MVD and at least be able to export models in the M-MVD format.

Figure 7-15. The connections between BIM authoring and engineering D/A tools in the proposed framework

In the current MVD implementation framework, models are exported in a MVD format to be directly imported in an engineering D/A tool. As a result, majority of the engineering D/A tools do not perform any model processing at the time of importing models, but only transform information from IFC files to their native format without any changes. In such framework, designers need to repeat several model modifications and enrichments each time a new model is imported. Examples are assigning live loads in structural models and adding thermal properties of materials in energy analysis models. The proposed MVD framework, which considers information interpretation throughout information exchanges, makes it possible to automate some of these post-importing model processing.
Since both input and output of IIE processes are IFC files, they could be implemented in any tool that supports importing or exporting IFC files. They could even be implemented in stand-alone third-party applications that import IFC models in a certain MVD and interpret another model, saved again in IFC format but potentially in a different MVD, which is suitable to be imported in another tool. The modular nature of the proposed IIE development methodology makes the iUs designed in an IIE reusable for other software developers, although different developers may choose a different algorithm for automating the interoperation process designed in an iU. In addition, the IFC layers of iUs specify IFC concepts that need to be included in the input files to be able to perform a certain interpretation process. These would act along with IDMs to identify information units that need to be considered in development/updating of MVDs in order to make them support interpretation process in the information exchanges they are developed for.

One of the unique characteristics of IFC data schema, which adds to complexity of IFC model exchanges, is that an identical information unit can be represented in many different ways in different MVDs. Addressing this complexity is one of the main incentives of performing this research that led to proposing the IIE concept. Therefore, implementation of the IIE concept for other common Architecture, Engineering, and Construction (AEC) standard file formats that could be represented in only one model view would not result in much benefit. However, the IIE process can still be used for such file format to automate post-importing model processing.
7.7. Summary and Concluding Remarks

This paper has presented a new concept referred to as Interpreted Information Exchange (IIE) to facilitate application of BIM for engineering analysis purposes. IIE could be used to automate transformation of building information models to engineering analysis models, and in this way can address the difficulties associated with this type of information exchange. In contrast to the conventional Direct Information Exchange (DIE) that semantically delivers information, the IIE approach first builds some interpreted information on top of the information extracted from the origin, then semantically transfers it to the destination. Three types of transformation are identified to be the main interpretation required for this type of information exchange: enriching, simplification, and change in representation.

The IIE concept is presented in this paper as a solution to lack of standardized discipline-specific MVDs that need to be created by software developers in the industry. In the current state of the implemented information exchange standards in the construction industry, most of the available commonly implemented MVDs are related to model coordination, while several disciple-specific MVDs are already available that are not implemented yet. Without implementation of disciple-specific MVDs, information exchanges from BIM authoring tools to engineering D/A tools are very inefficient and error-prone. The IIE makes it possible to use the standards that are already widely implemented by the major AEC software developers for interpretation of required information for different disciple-specific uses, as oppose to requiring all BIM authoring tools to implement several domain-specific standards, which would be very expensive to
Initially implement, validate, and maintain. In other words, implementation of the IIE concept makes it possible to create IFC file in several engineering domain-specific standardized MVDs, which are not yet implemented by BIM authoring tools, from the few currently widely implemented MVDs such as CV V2.0. Such approach could shift the responsibility of exporting engineering domain-specific MVDs from the developers of BIM authoring software packages to the developers of engineering D/A analysis tools, who are actually the ones that need and benefit from implementation of domain-specific MVDs. It also facilitates participation of third party software developers to use the export model of BIM authoring software and connect them with domain-specific applications by development of stand-alone, server-based, or add-on tools.

A methodology is presented in this paper for design and standardization of the IIEs in different engineering areas. This methodology is consistent with the IDM/MVDs standardization methodology developed by buildingSMART. In this methodology, the whole interpretation process in an IIE would be divided into several exchange units (eUs), each responsible for exchanging one or a set of related information units. The modular nature of the methodology makes the units of an IIE standard reusable in others, if applicable. Using the proposed methodology make it possible to achieve the capability of recognizing, extracting and exporting the required information from the building information model to the corresponding engineering D/A model.

The IIE concept along with the presented design and standardization methodology is validated in this paper through a case study on development of a tool for interpretation of the structural model from the architectural building information model. The experience
gained has been discussed in the form of a series of requirements and recommendations. Duplication of interpreted instances in the output file and the right time delivery of information to different eUs are identified to be two of the most important issues that should be managed during implementation of an IIE. Development of different IIEs along with their implementation in the construction industry enhances interoperability between BIM and engineering software packages, increases richness of the engineering D/A models generated from the building information models, decreases efforts required for preparation of imported building information models for analysis, increases reliability of the exported IFC files by mitigating testing and licensing of the IFC file exporters, and helps for faster and easier adoption and implementation of the newly developed MVDs.

7.8. References


CHAPTER 8: Platform Developed for Implementation of IIEs

Implementation of the Interpreted Information Exchange from A Software Development Point of View

Issa J. Ramaji, Ph.D. Candidate\textsuperscript{1}, Ali M. Memari, Ph.D., P.E.\textsuperscript{2}

\textsuperscript{1} Ph.D. Candidate, Department of Architectural Engineering, Penn State University, 104 Engineering Unit A, University Park, PA 16802, E-Mail: ramaji.issa@gmail.com
\textsuperscript{2} Professor, Department of Architectural Engineering and Department of Civil and Environmental Engineering, Penn State University, 222 Sackett Building, University Park, PA 16802, E-Mail: memari@engr.psu.edu

Abstract: Engineering design is one of the most in-demand Building Information Modeling (BIM) uses that is needed to generate engineering analytical models from BIM physical models. Due to efforts required to modify and prepare the imported model for analysis, the difficulty-to-benefit ratio is low in this BIM use, thereby, some designers prefer to create the analytical model from scratch. These preparations are more geared toward modifying the model based on the designer’s interpretation of the BIM model and including additional engineering information. Automating the interpretation and model transformation process can significantly facilitate the information exchange related to this BIM use. A platform is developed and presented in this paper that contains the fundamental functionalities required for automating Interpreted Information Exchanges
(IIEs) process. The platform allows input of IFC BIM models and export of models transformed by the interpretation engines, which are the extensions made to automate an interpretation process.

Keywords: Building Information Modeling, Interpreted Information Exchange, Engineering Analysis, Model View Definition, Industry Foundation Classes.

8.1. Introduction

Building information modeling (BIM) emerged almost two decades ago as a solution for information management of building construction projects. Over the past decade, BIM adoption in the industry has significantly increased as its numerous potential benefits have been achieved by improvement of the supporting technologies such as tools and platforms (AEC Magazine 2013; Fallon and Palmer 2007; Gallaher et al. 2004; McGraw Hill Construction 2012). The rich information contained in BIM models could be used for several purposes. CIC Research Group (2013) identified 15 different BIM Uses from early stages of design to the operation and maintenance phases. Automating the engineering modeling process for creation of high quality and accurate analytical simulation is one of the benefits of engineering design BIM Use that can eliminate design errors and reduce design cost and time (Alwisy et al. 2012; Gane and Haymaker 2007; Sacks and Barak 2008; Sudarsan et al. 2005). Structural design/analysis
and energy performance analysis are two examples for this BIM Use that are being used frequently in the industry. Although some BIM authoring software are capable of engineering analysis to some extent, they are not usually as sophisticated as the stand-alone engineering analysis software products. Hence, many efforts are made to develop links between different BIM software products and engineering analysis tools (Becerik-Gerber and Rice 2010; Guzmán and Zhu 2014). These efforts are mostly focused on information translation from BIM physical models to analytical models as the key process for incorporating this BIM Use. Usually, these translators and information extractors could be found in three different forms: 1) entirely separate from base and self-sustaining, 2) a sub-routine within existing BIM authoring or engineering discipline-specific software such as Revit or SAP2000, or 3) using a sever-based platform like BIMserver for data management along with query functions such as BIM Query Language (BIMQL) for extraction and manipulation of the required information. An example of this information translator is the web-based integration tool developed by Zhang et al. (2014) for bi-directional information exchanges between IFC BIM files and several common proprietary structural design file formats. Another example is the tool developed by Guzmán and Zhu (2014) to interpret energy performance analysis models directly from BIM models. Still another example is the use of Java programming language for partial information extraction from IFC files by Zhang and Issa (2013). Won et al. (2013) developed a tool for similar purpose while it uses an algorithm independent from the IFC schema data structure. Jiang et al. (2012) created an add-on for the BIMserver platform to import architectural BIM model in IFC format and export the
energy analysis model in native format of Energy Plus software. Other examples are the tools and platforms developed by Hassanien Serror et al. (2008), Wang et al. (2013), Nepal et al. (2012), Polter et al. (2014), Qin et al. (2011), and Liu et al. (2010) for different information transformation purposes.

According to Bazjanac (2008), to achieve an efficient information exchange for engineering analysis BIM Use, the information should be transformed and enriched before being used. Although Industry Foundation Classes (IFC) is the open file format for BIM information exchange, the current way of using this language for engineering analysis BIM Use is error-prone and ad hoc. Some of the current common issues include the following: discontinuity between connected elements, lack of required information related to material and engineering properties of elements, inaccurate representation of elements, and missed information due to interoperability errors. Model transformations required for engineering design BIM Use include filtering, translating, and adding new information to the model (Bazjanac and Kiviniemi 2007). Manual information transformation is tedious and in most cases inefficient, so many designers prefer to create the model from scratch instead of using the BIM model. To address this issue, this study has developed a mechanism for documentation and automation of the interpretation process called Interpreted Information Exchange (IIE). In this process, the BIM model in IFC format is converted to the interpreted equivalent analytical model again in the IFC language, while having the enriching information added. In such a workflow where both input and output information are represented in IFC format, an open interpretation system
would result, which could interact with a large variety of BIM authoring and engineering
tools.

One can easily open an IFC file in any text editor and simply copy and paste the needed
lines into another file; and although the files are human readable and there is even room
for instance manipulation, this is still not enough to make the resultant file useful. The
file needs to consider interconnectivity between instances and the schema’s
constrains/rule sets, which requires extraction, manipulation, and interpretation of BIM
information be carried out in a controlled environment. To do so, a platform is
devolved and presented in this paper for extraction of information from BIM models
and implementation of the IIEs. This platform offers functionalities required for reading
an IFC file, extracting required information, interpreting desired information, and
generating an output IFC file for semantic exchange of the interpreted information. In
other words, the platform provides several programing functions essentially required by
a tool developer for programming an automated IIE process to be used for different
engineering purposes. In such a framework, the platform is the core while these tools
would be extensions or add-ons, called interpretation engines, which input information
from the parsed instances of the input file and generate instance objects related to the
interpreted information. The C++ language is used for development of this platform as
it is one of the most powerful programming languages and is compatible with different
computer operation systems.
In the following section, first the IFC language along with the structure of related files and schemas are explained as they are essential in the proposed platform. Next, the IIE concept and its documentation methodology are briefly discussed, followed by introduction of the overall workflow and individual modules of the developed platform, each devoted to a certain functionality. This paper also discusses the integration and the framework for communications among different platform modules, as they go through their assigned processes.

8.2. Industry Foundation Classes

With improvements in manufacturing industry, the products have become more complex with a clear need for more organized product information model. Product modeling is a key element for implementation of advanced technologies in manufacturing (Zhao and Liu 2008). As products employ such technologies, more experts and engineers need to be involved in the projects to address the need for resilient and robust information exchanges. In the 1980s, International Standard Organization (ISO) started an effort to address this need by developing a standardized product data model supplemented with a STandard for the Exchange of the Product model data (STEP) (Spiby 1992). This product information model is developed in Express language (Pratt 2001; Zhao and Liu 2008). By introducing STEP in 1992, it became the main approach for information modeling in a
wide range of industries such as mechanical and electrical systems, ship building process, and furniture manufacturing to name few (Pratt 2005). In 1994, International Alliance for Interoperability (IAI), later changed the name to buildingSmart International, started to use STEP and Express language to develop a standard non-proprietary information model for the whole lifecycle of building construction (Spiby 1992). The result of this efforts is Industry Foundation Classes (IFC) file format.

All definitions of information units developed for building industry along with relevant relationships are written in IFC schema file using Express language. According to Eastman (1999), schema is a unit that defines Universe of Discourse (UoD) and declares objects associated with their dependencies and purposes. UoD is defined by Halpin and Morgan (2010) as the application area from which the information model is being developed. IFC files use the entities defined in schema for declaring project objects and related information. In other words, IFC files represent project-specific information in the data structure defined in the IFC schema. Both IFC files and schemas are stored in text-based file formats, respectively, with “ifc” and “exp file extensions. IFC schema gives meaning to the IFC file, without which IFC files will be some meaningless lines of text.

Two types of entities are defined in the IFC schema: Object and Links. Objects can be either physical or non-physical components or properties such as beams, walls, project’s owner, time, and dimensions used in the building industry. Since Express is not a fully object-oriented language, STEP lacks links that connects objects together. This
problem is solved in Express by defining some entities that play the role of links (also known as Relations), which are the entities that connect Object entities together. An example is IfcRelAggregates that is used for defining an object composed of many other objects such as ramp that is made of railings, flight, and slab. Another example is IfcRelAssigns that is used for assigning two objects together like assigning a client to a project.

IFC schema is made up of several blocks, each defining one entity. In an entity definition block, five important types of definitions could be found including super-type, sub-types, attributes, and rules. As mentioned before, STEP uses the object-oriented concept, thereby, each entity could be a sub-type of another entity, and in this case inherits all attributes of its super-type. At the beginning of the block, super-type and all sub-types of the entity, if there is any, are declared. Then, immediately after sub-type and super-type declarations, specific attributes of the entity that are not inherited from its super-type are defined. The attributes of an entity in STEP could be classified in three categories that are introduced in the following:

- **Explicit:** These attributes are information units that characterize the properties of the entity. An explicit attribute could be a single value like an integer number, a set of values with the same type, or a reference to other Object entities.

- **Derived:** This type of attribute is not loaded as data like Explicit attributes, but derived from Explicit attributes of that entity.
• Inverse: These attributes are like connectors that could be used to relate two or more Object entities together using Link entities.

The last part in the block is rule definition that is used in the schema to define constrains. Rules are defined in the form of equations and are shown after the “WHERE” line in the entity definition block. The value of the rules could be true, which means it is following the rule; false, that means it is violating the constraint; and unknown, which means some attributes are missing.

8.3. Interpreted Information Exchange

Incorporating BIM for engineering modeling purposes is very tedious and time consuming, unless the required interpretation be automated in the process of information exchange. A methodology is developed in this study for documentation and standardization of these interpretations using IFC language. In this methodology, firstly the whole interpreted information exchange is divided in several exchange Units (eUs). Two types of eUs were identified to be included in each automated IIE process: directly-exchanged Units (dUs), and interpreted Units (iUs). The dUs are responsible for transferring information from the input model to the output model without any changes,
whereas iUs transform a set of information units to another set. Three types of information interpretation were identified to be required for transformation of physical models to analytical models: enrichment, representation transformation, and simplification. In the enrichment process, additional information required for engineering simulation of the building are added to the model; adding mechanical properties of the structural materials to the model based on the materials’ names is an example of this type of transformation. Representation transformation may be carried out in an IIE to deliver information in a more desirable format; an example is transforming the extruded area representation of a linear element to its equivalent topological representation. Simplification that is generally the most complicated type of interpretation consists of simplifying a complex building component or any other complex information unit to its simplified analytical from; interpretation of the equivalent structural shell area element from a cold-formed steel wall in BIM model is an example of simplifying transformation. According to the developed methodology, the following five steps are needed in order to design an IIE for automated generation of an engineering analytical model directly from BIM models:

- **Categorizing the information units into direct-exchange and interpretation units:**

  The first step is focused mainly on recognition of the required information in the analytical model that could be exchanged directly from the BIM model, as well as the ones that are not included in the BIM model and need to be interpreted. In other words,
this step divides the information units contained in the engineering model in two categories: directly-exchanged and interpreted information.

• **Defining eUs and mapping them to the related units in the BIM model:** This step classifies all the recognized information units from the previous step into several eUs and map the interpreted information units to the related information in the BIM model. In other words, the information in the BIM model upon which the interpreted information could be extracted are specified. For better management of the IIE, the eUs should either transfer interpreted (iUs) or directly-exchanged (dUs) information units. It is preferred to have only one dU in each IIE, whereas several iUs may be considered in an IIE, each carrying out one specific interpretation process. The result of this step is specification of the inputs and outputs for each eU.

• **Binding the iUs to the IFC file format data structure:** Binding the input and output of the eUs to the IFC schema is the main focus of this step. The outcome of this step specifies the IFC file instances that should be queried from the input file, the instances that should be directly passed to the output file, and the instances that should be created based on representing the interpreted information. Binding these eUs to IFC language makes it compatible with large variety of BIM tools, as IFC is an open standard file format for BIM models.

• **Design a translation mechanism for interpretation:** This step designs the algorithm for interpreting the outputs of iUs from their inputs. In other words, this step specifies how the input and output of each iU are connected. As interpretation may need input
from external references other than the input file, the source of these references should also be specified in this step. An example is the thermal resistance of a layer of insulation that is required for energy performance evaluation analysis of a building; this information could be obtained from the product specification table published by the manufacturer.

- **Testing the designed IIE:** The last step is testing the designed IIE by means of a case study to make sure that the defined interpretation units along with the interpretation algorithm are designed properly. To investigate the interoperability of the platform, it is preferred to run the case study for information exchange between different combinations of BIM authoring and engineering analysis tools, as each export IFC files slightly different than others.

Once the first four steps of the IIE designed, the platform presented in this paper could be used for testing or development of a tool for automating the interpretation process. The following sub-section reviews the overall workflow of the proposed platform and the relationship between its different modules.
8.3.1. Interpretation Workflow

Industry Foundation Classes (IFC) file format is a digital text-based data structure for modeling building information. Files in this format are made of many instances, each containing an information unit or defining a relationship between units. Each information unit or relationship is defined to be an object in the IFC data structure, and each instance constructs one of these objects, which are defined in the IFC schema as entities.

An interoperable interpretation platform imports IFC files as input information; furthermore, the interpretation platform needs to be able to convert the instances contained in the input IFC files to the related IFC objects. Since export file format is also IFC, the platform should be capable of transforming interpreting objects into IFC instances to be written in the output file. As the workflow of the interpretation platform is illustrated in Figure 7-1, in addition to the functionalities required for reading and writing IFC file, the platform should also be capable of querying information from the objects generated from input IFC file as well as generating IFC objects related to the interpreted information. The querying functions of the platform feed the required information to both iUs and dU. In brief, reading IFC file to equivalent IFC objects, querying required information from the read IFC objects, generating new IFC objects, and writing the IFC objects into the output file are the four main needed functionalities for an interoperable platform. These four functionalities could be coupled with any interpretation engine for automating the interpretation process. The interpretation engines use information provided by the query functions along with several internally defined rule-sets to generate...
interpreting information. This is followed by passing the information through the IFC object generator functions to generate the interpreted IFC objects. The last step would be generation of the output file by writing the directly-passed and interpreted IFC objects in the form of IFC instances. In the following sections, the way each of these functionalities are added to the platform is discussed.

Figure 8-1. Workflow of the interpretation platform
8.4. Importing the IFC File

Reading IFC file involves converting text-based instances included in an IFC file to the related digital IFC objects. An example is converting an “IfcBeam” instance shown in Figure 8-2 to an object that contains several attributes as shown at the bottom of this figure. IFC files do not include any information about the meaning of the values listed in the contained instances. Therefore, parsing an IFC file has two steps: 1) reading instances from the IFC file and 2) assigning meaning to the read values. In the following, each of these steps are discussed in more details.

Figure 8-2. An example for the parses IFC object
8.4.1. Reading IFC File Instances

As shown in Figure 8-3, each instance contains three main parts including index, concept name, and item list. Each instance starts with an index, followed by a concept name and ends with a list of items. All indexes start with the “#” symbol and are the unique identification number of each instance in the IFC file; the concept name is the name of the entity from IFC schema that the instance is constructing; and the items are the attributes contained in the instance. Items included in an instance are separated by the “,” symbol in the list. Item lists include only the values of items while the name and meaning of these items could be inferred from the related schema.

![Figure 8-3. Anatomy of IFC instances](image)

In majority of available platforms for reading IFC files, a class is defined for each of the entities defined in the IFC schema. These classes are predefined in the library of these platforms and a copy of each is constructed per related instances contained in the imported IFC file. This approach makes the platform highly dependent on the schema it is designed for, and makes maintenance of the platform difficult for capturing schema’s
version changes. A new approach is taken in this research for development of the IFC parsing module. In this approach, one class is defined to support all instances with different concept names instead of defining one class for each.

As depicted in Figure 8-4, the class defined for modeling IFC instances includes six variables and two functions. The “index” variable stores the index of the instance. The “indexNum” stores the index without the “#” symbol in a numerical format to be used for sorting the instance list. Sorting instance list based on a numerical variable significantly speeds up the querying process compared to searching based on the string-type index. The “concept” variable stores the concept name of the entity being constructed in the instance. The “items” variable stores the value of the instance’s items. These items may be a single value or a list of sub-items. For example, the list of building elements contained in a building story is written as one item in an “IfcRelContainedInSpatialStructure” instance. This type of item is named list-item in this research. Since each of these sub-items may be independently queried, the “itemList” variable is defined to store the sub-items individually. The “raw” is the last variable that keeps the raw un-parsed text-based value of the instance. This variable is used in the first and last steps of reading and writing stages of the platform’s workflow. The two functions included in this class are defined for parsing information to the variables. The “Instance_Analysis” function parses information from the raw form of the input instance, while the “listItem_Analysis” function breaks down the list-item variables into several sub-items and stores them in the “listItems” array.
8.4.2. Reading IFC Schema Entities

IFC schema defines several entities related to the concepts being used in the building construction industry. In the data schema of the IFC files, both relationship and concepts are considered to be an entity. Each entity is separately defined in an entity block in the IFC schema along with its inheritance relationships. Reading an IFC schema mainly consist of parsing information contained in these blocks. The anatomy of an entity block is shown in Figure 8-5.
Similar to the instance parsing module, the same single-class approach is taken for modeling the schema’s entities. As shown in Figure 8-6, this class includes seven variables. The “concept” variable stores the name of the concept while the “supertype” and the “subtype” variables keep information related to the concept name of the parent (super-type) and children (subtypes) of the entity. The “abstract” variable is a logical variable that can be assigned true or false. This variable specifies whether or not the concept is an abstract object. The “items” variable stores entity’s list of attributes. These attributes lists contain both the specific attributes of the concept and the attributes inherited from its super-type. These items are the ones whose value appear in IFC instances. Next variable is the “invItems” that stores Inverse attributes of the entity. These attributes specify types of relationships the concept may have with other concepts and are used for checking the import file against schema. The last variable is the “block”
that is read in the first step of the schema importing process and keeps the raw text-based definition of the schema. The “SchEntity_Analysis” function is used for parsing information from the block variable to other entity’s variables.

```cpp
class SchemaEntity {
public:
    string concept;
    bool abstract;
    string supertype;
    vector<string> subtype;
    vector<Item> items;
    vector<InvItem> invitems;
    vector<string> block;

    // SchEntity_Analysis(vector<string> block) {...}
};
```

Figure 8-6. Class definition for modeling IFC schema entities

Concept names are the connections between the schema entities and the IFC file instances. In other words, each instance is an ally of the IFC entities that has the same concept name. As a result, their “items” variables are allies of one another. Hence, the name and meaning of each attribute of the instance could be inferred from the items of the related concept entity. This method of parsing information makes the platform independent from the version of the schema and the IFC file. As far as the version of the IFC file is consistent with the imported schema and the IFC file does not violate the schema’s constraints, the platform can parse the information.
8.5. Checking the Input File

Reading, writing and querying functions of the proposed platform are independent from the schema, whereas instance generator functions and interpretation engines are closely dependent on the IFC schema. On the other hand, the output IFC file is a mix of instances generated based on the interpreted information and the instances passed directly from the input file. Since interpreted instances are generated based on the schema, instances of the input file should observe the schema to have a homogeneous output file that is consistence with the schema.

Malfunctioning of interpretation engines is another problem that can be caused due to inconsistency of the input file with the schema. Places where certain information could be found in the input instances are based on a series of assumption in the interpretation engine for the. For example, the length of a beam element in an “IfcBeam” instance could be extracted from the “Representation” item that is the seventh item in the item list. Based on the schema, “IfcProductRepresentation” is the concept that should be cited in this item. Hence, an interpretation engine that needs to extract the length of an “IfcBeam” object may be designed to find the information in an “IfcProductRepresentation” instance or any of its subtypes. In this case, if an IFC concept that is not of the same type or any of its sub-types appears in this item, the engine cannot find the desired information, and as a result, it may misinterpret the targeted information.

To guarantee uniformity in the output files generated by this platform, a module is included in this platform to check the input file against schema before running
interpretation engines. Several possible violations that the input IFC file may have are identified, and a check function is accordingly designed for each. In the following, these potential inconsistencies are discussed along with the workflow of checking functions:

- **Validity of the concept:** This checks the existence of the instance’s concept in the version of the IFC schema to which the file is related. This check could be done simply by searching for the concept name in the entity list of the imported schema. The check is passed if an entity with the same concept name exists in the schema.

- **Length of the item list:** This check is needed to see if the number of items included in the instance is consistent with the schema. This check counts the number of attributes specified in the related schema entity and compares it with the length of the instance’s item list.

- **Item availability:** The purpose of this check is to see if all items that are not optional have values in the instance. To do so, the check function goes through all items mentioned in the related schema entity to determine if it is optional; if it is not, the function checks the value of the related item in the instance. If the item value is not equal to “$”, it means that the item is not blank and the check is passed.

- **Item cardinality:** This inspects all items of the instance, finds its list-items, and checks the cardinality of their sub-items. In other words, this function identifies the item-lists of the instance based on the schema and checks if their values are enclosed in prentices
as it is a requirement for IFC files. This function then tests the number of the sub-items contained is checked against schema.

- **Referred instance:** This check is intended to see if the cited instance in an item observes the constraints of the schema. To do so, the check function goes over the instance item to determine if the cited instances match the IFC concept that is specified in the IFC schema for that specific item. If there is no match, the query function finds all sub-types of the referred concept in the schema and checks the referred instance’s concept with this sub-type list. If the instance concept is available in this list, it means that the check is passed and the item is following the schema’s constraints in referring to other instances. If the item is a list-item, this check function would be executed for each of the sub-items separately.

These are the essential checks all IFC files should pass. The designed checking module in this platform queries the constraints directly from the schema being used for generation of the input IFC file. Hence, this module is independent from the version of the input IFC file. In addition to the constraints defined in the schema, there are some constraints that are specific to the Model View Definition (MVDs) followed by the input file. These constrains are more related to the representation of elements and relationships between different instances. Hence, the interpretation engines should be designed to test if the constraints of the used MVDs are observed in the input file, or at least filter the instances that violate these constraints.
8.6. Information Querying

Information querying in the imported file is one of the fundamental needs of any information platform. Query functions of the developed platform that are the links between input files and interpretation engines search the parsed information instead of the raw instances of the IFC file, and this significantly speeds up querying process. Generally, the simplest form of information query in an IFC platform would be searching for a specific instance. Instance sorting and searching take place based on the numerical format of the instances’ index. The reading functions are designed to sort the imported instances at the time of parsing input file. Since the objects are read sequentially, the Insertion Sort method is used for sorting the parsed instances, where combining reading and sorting processes by insertion of the read instances at their right sorted places speed up the sorting process. Searching for certain types of instances is another type of query. Some of the interpretation engines need to access a certain type of object. For example, all instances declaring columns contained in a certain story may be queried by an interpretation engine. In this type of query, the “concept” variable of the instances are inspected to see whether or not they match the searched instance type.

In majority of the instances, definition of the concepts is carried out by referring to other instances. Therefore, the relationships among different instances are key elements for extraction of the required information. Generally, the relationships among IFC objects could be classified in two categories: essential dependency and relative dependency as discussed in the following.
• **Essential dependency:** Essential dependents of an instance are part of the information required for declaring an instance. These instances are referred to at the item list part of an instance, where referring takes place by mentioning the index of the referred instance. For example, in the IFC object shown in Figure 8-7, the instance refers to instance “#158” for specifying representation of element. In this example, instance “#158” is an essential dependent for the “#162”. Essential dependents of an instance could be categorized in two groups: immediate essential dependents and indirect essential dependents. Immediate dependents of an instance are the ones that are cited in its item list. In some cases the immediate essential dependents of an instance refer to other instances for their definition. Referring to other instances continues until none of the cited instances contains any essential dependent. All of these instances that are essential dependents of the instance and do not appear in the instance are called indirect essential dependents of the instance.

• **Relative dependency:** As mentioned in the previous section, there are some concepts in the IFC schema called Link that are to relate several objects to one another. In fact, they are the relationships in the object oriented data structure of the BIM model. There are different types of Link IFC concept defining different types of relationships, but they are all sub-types of the “IfcRelationship” concept. The names of all these concepts start with “IfcRel”, thereby, instances related to these concepts could be queried by checking the first six characters of the instance “concept” variable. Assigning a property to an object, assigning several elements to the assembly they are aggregating, and assigning a containment relationship between the elements located in a space to the
containing space are a few examples for application of these concepts. These relationships are the attributes that are referred to as the Inverse attributes of the IFC concepts in the schema. The instances that are of this kind and relate an instance to other instances are called the relative dependent of that instance. These dependents do not appear in the item list of an instance, but they cite the instance in their item list. In the example shown in Figure 8-7, instance “#194” is a relative dependent for the instances “#162” and “#189”, while they are both essential dependents of the instance “#194”.

Figure 8-7. Relative and essential dependents of an instance.

According to the above explanation, the five main types of query functions designed in this platform are as follows:
• **Finding an instance based on the index:** This function reads the index of an instance, transforms the index to a numerical format and searches the transformed index in the sorted instance object list.

• **Finding immediate essential dependent of an instance:** This function starts with an instance index, finds the instance in the list using the instance finder function, and then makes a list of all its immediate dependents by going through the instance’s item list.

• **Finding essential dependent of an instance:** This function too starts with the index of an instance and runs the immediate dependent finder function recursively on the instance and its immediate essential dependents to find all its immediate and indirect essential dependents.

• **Finding relative dependent of an instance:** This function searches for relative dependents of an instance by using the instance’s index. It goes through the item list of all IFC relationship objects and checks for any relations of the input instance to other objects. This function returns a list of relative dependents of the instance.

• **Finding instances with a certain concept name:** This function starts with a concept name and a list of IFC objects and searches all the objects to see if any of their “concept” variable is equal to the input concept name. This concept could be combined with any of the dependent finder functions for finding instances of a certain type in the dependent list of an object. For example, it could be used to check if a certain piece of information such as a certain property set of representation type exist for an object.
In addition to the query functions required for data extraction from the input IFC file, some query functions are also defined for querying information from the IFC schema. In the following, their main four functions are discussed.

- **Finding schema entity related to an instance**: This function starts with the concept name of an instance and finds the related entity in the IFC schema. This is carried out by comparing the input concept name with the value of the “concept” variable of the schema entities.

- **Finding sub-types of a schema concept**: This function starts with a concept name and finds all its sub-types and sub-types of the found sub-types. The process continues to the point where none of the sub-types has any lower sub-type. This function is used to check whether a concept is sub-type of another concept at any level.

- **Finding attributes of a concept**: Each schema block defines only the attributes that are specific to the entity being defined, while the sub-types of the entity inherit these attributes. This function is designed to make a list of attributes that are either specific to the input concept or inherited form the concept’s super-type.

- **Finding location of an attribute in a concept**: Since the instance class defined in this platform stores the instances’ items regardless of their meaning, the capability of searching for a certain attribute is added to the platform. This function uses the instance’s concept name and the name of the attribute that is considering; it then returns the location of the item in the instance. The locations are addressed by order of the item in the array of the item list.
8.7. Generating New IFC Objects

Information interpreted by the interpretation engines should be written in the IFC language to make the output file importable by a wide range of BIM software. Although the data structure of the interpretation engine may be different from the IFC schema, it should be transformed into the IFC data structure before being written as the instances of the IFC file. As interoperation engines feed interpreted information to the IFC instance generator functions that are designed for certain MVDs, the information should be transformed to the data structure of the MVDs before calling these functions. Usually all the information about an object cannot be written in only one instance, and as a result, a group of instances are required to write an information unit in the output file. Therefore, the process of generating instances for representation of an information unit could be divided into the following two steps: 1) generating individual raw instances, and 2) assigning values to the generated instances and linking them together. These two steps are discussed in the following and their connection is illustrated in Figure 8-8.

8.7.1. Generating Raw Instances

The first step in generating an instance is constructing an instance object. An index should be assigned to the newly constructed instance to give it an identity and make it possible to refer to the instance. As the index of the object should be unique, and considering the fact that some instances of the input file would be passed directly to the
output file, we cannot start the indexes from “#1”. This problem is addressed in this platform by finding the largest index of the input file and starting from there for generating an index for the instance. Assigning the concept name to the instance is the next step. Since the input of the instance generator function is the concept name, the function just has to copy the input concept name into the “concept” variable.

The next step would be generating an array of blank variables to be filled by the items of the instance. The number of items may be different for each concept and could be found from the schema. To find the length of the array, the instance generator function goes through the schema and counts the attributes the concept has or inherits from its super-type concept. These item variables would stay blank until being filled based on the interpreted information and connections of the instance with other instances. Since the blank items are shown by the “$” symbol in the IFC files, this symbol would be inserted for all these blank variables. This is for the case where no information is generated for an optional item during the interpretation process.

All the concepts that are sub-types of the “IfcRoot” concept need to have a Globally Unique Identifier (GUID). The GUIDs that IFC files use are not the standard 32 hexadecimal digits. Since development of the IFC language started in 1990s when the computer’s small memory was a real issue for computational tasks, IFC language was designed to use a shorter GUID. To assign GUID to the related item variable, this platform first generates a standard 32-character standard GUID and then encodes it to a
Since the instances generated in this step refer to the schema for acquiring the required information for generation of instance, this function does not need to be maintained for the IFC schema changes. In addition, since in this step no relationships to other instances are assigned to the generated instances, this step is also independent from the MVDs that are used in interpretation engines for structuring the export information. The process of generating instances in this step along with its connection with the next step is depicted in Figure 8-8.

8.7.2. Linking and Value Assignment

In this step, the values of the items would be assigned based on the relationships of the instances or results of the interpretations. The relationships are presented by citing instance index of the related instances in the item variables. Since IFC is a redundant file format, instances could be related to one another in many different ways. Therefore, the standardized structure of the MVD that is targeted by the interpretation units should be used for relating the generated instances. As a result, unlike the first step, this step is not model view-independent and functions that are defined in this step are specific to the MVD they are designed for. In addition, since the meaning and order of the items used
for defining connection are important in these functions, these are not independent from
the version of the IFC and should be maintained for the version changes of the schema.

Since the essential dependents of each instance are required for its definition, the
dependent instances should be constructed prior to the definition of the instance.
Therefore, instance generation related to an information unit should be started from the
lowest level of the essential dependents and continue to the upper levels. The values of
the items listed in these instances may be specified by referring to other instances or
direct insertion of the value obtained from the related interpretation engine. Since
interpretation engines may not follow the data model of the IFC schema for modeling the
extracted information, the imported information may need to be processed before being
inserted into the item variables. Furthermore, such information should be consistent with
the structure of the observed MVD and provide enough information for generating
instances and assigning values to their non-optional items.

There are some instances that are referred to by many other instances related to
different information units. An example is the “IfcOwnerHistory” instance along with all
its relative and essential dependents that specify the authors’ information. In this
example, since many information units may be created by an author, making a separate
“IfcOwnerHistory” instance for each of these information units significantly increases the
size of the export file. Default “IfcLocalPlacement” and
“IfcGeometricRepresenationContext” instances are some of the other instances that are
widely used by different instances. These types of instances are called public instances in
this research, while others are called private instances. Since public instances are required at the time of generating private instances related to the interpreted information units, they have to be generated before transformation of the interpreted information to IFC instances. “IfcUnit” along with all its essential dependents that define units of different numerical values mentioned in instance items are another example of public instances that should be generated before private instances. Figure 8-8 illustrates the connection of this step with the raw instance generation step, and shows the process of structuring the information by linking instances.
8.8. Exporting the Interpreted IFC File

Writing IFC instance objects into the IFC file is the last step of the IIE process. Before writing the instances into the file, first the header of the file should be generated. The header block that is made up of nine lines specifies some general information about the file including the version of the used STEP language ISO standard, file name, file description, and the IFC schema version used for constructing instances. Seven out of these nine lines appear before instances, while the other two appear at the end of the file after the last instance. The last header line before the instance and the first line at the end of the file are, respectively, indicators of the beginning and end of the file instance section. These header lines are constant standard expressions and are, respectively, equal to “ENDSEC;” and “DATA;”.

The next step in writing the IFC file is writing the instances that should be passed from the input file directly to the output file. These instances are related to the directly-exchanged Unit (dU). Definition of an instance is impossible without defining its essential dependents. Therefore, the essential dependents finder query function could be used for finding instances in the input file that should be passed to the output file along with instances pointed out in the dU. Since relative dependencies of dU’s instances are not of concern, it is not required to pass the instances that are their relative dependents. Instances passed from the input files along with the newly generated instances form the output file, hence the version of the input and output file should be identical. Unless, the
instances related to dU should be parsed into the format of the new version and then be written into new instances accordingly.

The file writing process continues with writing the public instances, followed by writing the private instances. To write an instance, the instance objects should be transformed into the standard text based format. The transformation process is in reverse order of the instance parsing process. First, the sub-items should be attached to one another to make the equivalent list items enclosed between parentheses with commas between sub-items. Next, the items should be attached to one another to make the item part of the instance. By adding the concept name and instance’s index to the beginning of the string, the raw text-based of the instance would be obtained. In the last step, these raw strings are written in the IFC file.

The process of generating the IFC file from the instance objects and input file is illustrated in Figure 8-9.

![Process breakdown of transforming an information unit into IFC instances](image)

Figure 8-9. Process breakdown of transforming an information unit into IFC instances
8.9. **Summary and Concluding Remarks**

Automating the interpretations required for transforming the BIM models to their equivalent analytical models can significantly enhance efficiency of this information exchange and as a result increase interests in using BIM for engineering design uses. Execution of the Interpreted information Exchange (IIE) needs supporting frameworks and technologies such as tools and platforms. To address this need, this paper presented a platform created for implementation of the IIE concept and made development of interpretation tools. This platform is made up of five modules, each responsible for performing a certain functionality: reading IFC file, checking the input file, information querying, generating new IFC objects, exporting IFC file. These five modules could provide enough tools for the developers to couple them with the interpretation engines to develop a tool for implementation of an automated IIE. The reading module imports the input IFC file and related schema, and then parses the read files into the native version-independent objects of the platform. Checking module compares the imported input file against the related schema to see if the file observes the constraints and rule-sets defined in the schema. Querying functions are responsible for searching and of extraction of the required information from the input file and related schema. Instance generator functions transform the interpreted information units to the IFC instance format, while the writing file functions write the newly generated instances to the output IFC file along with the instances that should be passed to the output file directly from input files. This platform could be coupled with any interpretation engine to automate an IIE process. The novel
approach used for parsing the IFC instances and the schema’s entities made this platform independent of the version of the input IFC file and the schema used. In addition, the workflow of this platform makes the developed tools highly interpretable, as it uses the BIM open standard IFC file format for both importing and exporting the models.

8.10. References


CHAPTER 9: Interpretation of Structural Analysis Model from Coordination Model

Interpretation of Structural Analytical Model from BIM in Coordination View

Issa J. Ramaji, Ph.D. Candidate¹, Ali M. Memari, Ph.D., P.E.²

¹ Ph.D. Candidate, Department of Architectural Engineering, Penn State University, 104 Engineering Unit B, University Park, PA 16802, E-Mail: ramaji.issa@gmail.com
² Professor, Department of Architectural Engineering and Department of Civil and Environmental Engineering, Penn State University, 222 Sackett Building, University Park, PA 16802, E-Mail: memari@engr.psu.edu

Abstract: Structural design/analysis is one of the most needed uses of Building Information Modeling (BIM). Transforming a building information model to an engineering analytical model is tedious and time-consuming. In addition to geometry transformation, extensive modifications and interpretations are required to make the transformed model ready for analysis. Currently, interoperability level between BIM authoring tools and structural design/analysis tools is relatively low, making some designers prefer to generate analytical models from scratch instead of importing
information form building information models. In this paper, fundamental interpretations required for generating structural analysis models directly from building information models are discussed. Accordingly, a mechanism is developed for transformation of building information models in coordination view to their equivalent structural models in structural view. The mechanism is designed to import and export models in Industry Foundation Classes (IFC) files as the BIM open standard file format. In addition, a tool is developed to automate Interpreted Information Exchange (IIE) documented in the proposed mechanism and is validated by implementing a case study.

**Keywords:** Building Information Modeling, Interpreted Information Exchange, Structural Analysis, Model View Definition, Industry Foundation Classes.

### 9.1. Introduction

Emerging Building Information Modeling (BIM) has significantly altered the way construction projects are conducted (Eastman et al. 2011; Jung and Joo 2011; McGraw Hill Construction 2012). BIM could be incorporated for different purposes throughout various stages of buildings’ lifecycle, engineering analysis being one of them (AEC Magazine 2013; Fallon and Palmer 2007; Gallaher et al. 2004; McGraw Hill Construction 2012; Ramaji and Memari 2015a; Solnosky et al. 2014). According to CIC Research Group (CIC (Computer Integrated Construction) Research Group 2013),
工程分析 BIM 使用是一个过程，用于确定最有效的工程方法，使用智能工具、设计规格和建筑信息模型。通过自动化过程，这已经从严格的和可靠的分析中结果，以及在建筑物中实现更好的质量和更高的性能，是使用 BIM 进行工程分析的最显著潜力值 (Alwisy et al. 2012; Cerovsek 2011; Gane and Haymaker 2007; Sacks and Barak 2008)。根据 Schinler and Nelson (Schinler and Nelson 2008)，从传统的实施过程开始进行工程分析，工程师可以缩短他们用于建模、设计和与其他学科协调的时间。为了能够在工程分析中使用建筑信息模型，需要 BIM 制作工具和工程设计/分析 (D/A) 软件之间的高可操作性 (ATC (Applied Technology Council) 2013)。在过去十年中，许多人已经努力为不同工程目的提供冗余和无缝的基础设施和平台，以从建筑信息模型中提取信息并用于工程 D/A 工具 (Bazjanac 2008; Sanguinetti et al. 2012)。这些努力可以分为两类：框架和工具。在第一类中，努力主要集中在信息交换过程和信息在整个项目阶段之间流动的要求的识别和不同学科之间交换信息的数字表示的标准化方面，例如信息传递手册 (IDMs) 和模型视图定义 (MVDs) (Laakso and Kiviniemi 2012; Venugopal et al. 2012b)。
et al. 2011; Nawari 2012; NIBS (National Institute of Building Sciences) 2012; Panushev et al. 2010; Ramaji et al. 2014). In the second category, efforts are focused on development of tools to support information exchange processes. These tools are developed in different forms such as stand-alone software, add-ons, or plug-ins for server-based platforms such as BIMserver (Jiang et al. 2012).

IDMs and MVDs developed by Virtual Building Laboratory (VBL) for architectural design to structural design information exchange and that of structural design to structural analysis exchange are examples of efforts made in the first category. These IDMs and MVDs are developed to standardize information exchange from architectural building information models to structural building information models with focus on the building geometry and spatial elements, and also exchange from structural design building information model to analytical structural models with emphasis on geometry of elements along with their structural information such as assigned material, cross sections, loads, load cases, etc. (BLIS (Building Lifecycle Interoperable Software) 2015). Another example is IDM and MVD developed by U.S. General Services Administration (GSA) for conceptual design to building energy analysis. This IDM/MVD is used for extracting required architectural information to form the basis of energy analysis. The focus of this IDM/MVD is mostly on space geometry and identifiers, space boundaries, construction types, material and material layers, and building elements that generate second-level space boundaries (BLIS (Building Lifecycle Interoperable Software) 2015).
There are several examples for efforts in the second category that were made to facilitate information exchange related to different types of engineering analysis BIM Use. Liu et al. (Liu et al. 2010) developed an integration tool to improve the IFC project file and export the structural model to PKPM structural analysis tool file format. The algorithm used for developing the tool contains four main steps, consisting of reading structural related building element such as beams, columns, braces, and walls; identifying relationship between elements; determining connections between elements; and analyzing the intersection relationship. The tool was developed by mixed language use of FORTRAN and C++, where classes and functions were written in C++ and the Main part that uses the functions and classes were developed in FORTRAN. Hassanien Serror et al. (Hassanien Serror et al. 2008) came up with an infrastructure, called Shared Computer-Aided Structural Design (sCAsD), for earthquake simulation of a structure using IFC file format within DOSE integrated construction management software environment. The whole process of information extractions and data processing in this tool could be divided into four steps consisting of defining structural dynamic actions and boundary conditions, defining entities and their relationship for the FEM model, assigning measurement entities from the IFC Resource layer to the structural analysis model, and integrating structural analysis and architectural models. Zhang and Issa (Zhang and Issa 2013) developed a tool in Java language to partially extract information from IFC files. This tool benefits from IFC format ontology and an ontology-augmented model index. In other words, the extracted data that follows the IFC ontology is slightly manipulated to be consistence with the augmented ontology defined by the tool developer. The algorithm
that this tool uses for data extraction has four main steps consisting of development of an augmented ontology model, finding targeted entities and seeking upward to find container entities, seeking downward to find referenced or related entities, and reassembling of the extracted entities and information. This methodology has the potential of being used for information extraction regardless of any ontologies. Won et al. (Won et al. 2013) used a similar algorithm for extracting the required information independently from the data structure defined in IFC schema. Wang et al. (Wang et al. 2013) developed a tool to extract information required for structural D/A from IFC file of structural BIM. The extracted information in this tool contains five main parts consisting of information related to unit system, materials information, members section properties, information related to member joints, and loading information. Qin et al. (Qin et al. 2011) used an xml-based unified FEM format and programmed with C++ language for bidirectional conversion of information from IFC file format and translating it to different proprietary structural analysis file formats. To support the proposed method, the authors mention the following two main advantages of using XML-based unified FEM as a link between IFC file and proprietary file formats: 1) since it is independent from IFC schema and proprietary file formats of the FEM software, it is easy to upgrade and maintain, 2) it is open, compact, clear, and as a result suitable for making bidirectional interfaces with numerous FEM software packages. Polter et al. (Polter et al. 2014) presented an under-development integrated cloud-based structural analysis platform that uses IFC file of the structural analysis model for some minor design iterations such as thickness of a wall. The rationale for this research is to develop a web service that makes advanced structural
analyses that need very high-speed computers affordable for companies and research facilities. The workflow of using this system includes the following six steps: 1) preparation and uploading structural FEM model IFC file, 2) defining desired variation parameters in a XML-based format, 3) providing further relevant information on the web GUI environment, 4) creating variant model by Model Generator unit of the platform based on the input IFC file and variant parameters, 5) analyzing variant models, and 6) sending results to the Data Manager unit for comparing and making it ascribable through the web GUI. Zhang et al. (Zhang et al. 2014) developed a web-based platform for bidirectional conversion of the architectural IFC model to proprietary structural modeling software file formats such as e2k and s2k. The core element of the approach taken in the aforementioned research is development of an IFC-based unified information model for data stored on the cloud space. This information model specifies the information necessary to be extracted for generating structural model of the building. An algorithm was developed in the aforementioned research to extract each piece of information specified in the unified information model from architectural BIM.

Structural design and analysis is one of the most used types of the engineering analysis BIM Use. In such Use, designers use building information models for generation of structural analytical models instead of making analytical models from the scratch. This would significantly shorten the engineering modeling time and increase accuracy of the analytical models as a result of better coordination between the analytical structural and building information models. According to the McGraw Hill Construction (McGraw Hill Construction 2014), after energy performance analysis, structural analysis has the lowest
value/difficulty ratio among different engineering analysis BIM Uses, while it has the highest frequency index (represents how often BIM is used for a process). This demonstrates that interoperable information exchange for structural analysis is so much in demand and the level of interoperability is not high enough to make this BIM use efficient. Hence, there is a great potential to benefit from employing BIM for structural design purposes only if the interoperability level between different BIM authoring tools and structural D/A software packages can be increased. Currently, BIM authoring tools cannot export structural analytical models in non-proprietary file format. As a result, many attempts have been made to develop tools, as discussed earlier, to connect them to some of the most common structural D/A software packages. To address this issue by using open standard file format, a mechanism is presented in this paper for interpretation of structural D/A models directly from architectural models in IFC format. This mechanism is developed using the methodology for documentation of Interpreted Information Exchange (IIE), which is developed by the authors as other part of this research. A tool is developed based on the presented mechanism to automate model interpretation. Since Coordination View Version 2.0 (CV V2.0) is the most commonly implemented MVD in the BIM authoring tools, both the mechanism and tools are designed to import architectural model in this format. Contrary to the discussed tools that are available for generation of structural models from architectural models, this tool does not rely on any proprietary file format for either importing or exporting information as it uses open standard file format for both purposes. The output format is designed to be IFC structural view as it is the most appropriate format to represent structural information.
this paper, first differences between model representations in architectural coordination view and structural view are discussed. Next, the methodology developed for documentation of the IIEs is briefly explained. Subsequently, the interpretation process is discussed by breaking down the whole mechanism into several units, followed by describing the units in more details. At last, an overview of the case study carried out for validating the mechanism and testing functionality of the tool is presented.

9.2. Coordination View versus Structural Analysis View

Coordination View is developed for coordination of building information models between architectural, structural, and mechanical disciplines. Coordination View Version 2.0 (CV V2.0) is the latest and most implemented MVDs in the construction industry. This model view is developed by buildingSMART International and currently is its only official MVDs being regularly maintained.

Virtual Building Laboratory (VBL) developed a structural analysis view in 2007 to support information exchange from structural building information models to structural analysis models. Development of such MVDs was coordinated by BLIS Consortium that was an ally of buildingSMART International at the time. Unfortunately, this model view that is called VBL-001 was not maintained after release. Therefore, a few of its concepts are deprecated and are not consistent with the new IFC schema. However, the construction software industry is using a slightly modified version of this model view that
is consistent with the schema. This modified model view is called Structural Analysis View in this research.

There are fundamental differences between representation perspectives taken in the Coordination View compared to Structural Analysis View. The approach in Coordination View is to represent elements by specifying boundaries of spaces they occupy; whereas in Structural Analysis View the topological finite element representation of structural elements and their connections are concerned. Figure 9-1 shows an identical model in these two model views. As shown in this figure, Coordination View represents elements as extruded areas, whereas in Structural Analysis View elements are shown by simple lines for linear elements and simple areas for planar elements. Structural Analysis View assigns cross section and thickness information, respectively, to linear and planar elements to specify their dimensions in more detail.

![Left) Model in Coordination View](image1)  ![Right) Model in Structural Analysis View](image2)

Figure 9-1. Two different representations of an identical model
The developed tool is designed to support prismatic form of the five main building elements including beams, columns, braces, walls, and slabs. These elements are represented, respectively, by “IfcBeam”, “IfcColumn”, “IfcMember”, “IfcWall”, and “IfcSlab” in Coordination View. Although Coordination View categorizes linear building elements based on their functionality into “IfcBeam”, “IfcColumn”, and “IfcMember” element types, their data structures are identical. “IfcMember” is defined to represent all elements that are neither horizontal as beams nor vertical as columns. Hence, since braces are inclined, they are represented by “IfcMember” instances in the coordination model view. Unlike, in Structural Analysis View, all of these linear elements are represented by only one IFC instance concept that is “IfcStructuralCurveMember”. Similarly, data structure of planar elements including “IfcWall” and “IfcSlab” are identical in Coordination View, while Structural Analysis View represent both of these concepts by only one instance concept named “IfcStructuralSurfaceMember”. Therefore, building elements of interest for this platform could be categorized in two groups: linear and planar elements. Coordination View represents both linear and planar elements as volumetric objects that are positioned with a certain orientation, whereas Structural Analysis View represents these elements as a series of connected Nodes. In Structural Analysis View model view, linear elements are represented with two nodes, while planar elements have more than two nodes.

Coordination View is designed to specify the general information of building elements such as location and borders of building element that may be required for
coordination purposes among different project disciplines. In such an approach, connectivity relationships among different elements are not of concern, thereby, no connectivity information is included in models generated according to this model view and it cannot be inferred from such model if two elements are connected. In Coordination View, there is no overlap among elements at the connection node of the two elements as the elements are represented in such a way that the end section face or the end side face of one element abuts the end side face (e.g., flange) of the other element. On the contrary, since Structural Analysis View uses vertices to specify boundaries of an element, connectivity of elements could be obtained from nodes that belong to more than one element. This difference in representation of connection nodes is illustrated in Figure 9-2.

Figure 9-2. Two different representation of a connection point
9.3. Interpreted Information Exchange

During the past decade, many model views have been standardized for different BIM Uses. These standards are useless unless they are implemented by software developers. Among available MVDs, only Coordination View is widely implemented by software developers, while some of these software packages can only import or export models in this view. Therefore, utilizing this model view as the basis for creation of other model views through Interpreted Information Exchange (IIE) would eliminate the need for implementation of all MVDs by BIM authoring tools, and as a result, this approach can significantly enhance interoperability in BIM information exchange. IIE may have more benefits for engineering analysis BIM Use, where several information transformations are required to make the imported model ready for analysis. Part of these transformations are related to interpreting engineering models of building assemblies and components as different perspectives may exist for different discipline-specific engineering representation of the building (Bazjanac and Kiviniemi 2007). Besides this type of transformation, there are some engineering information that needs to be included in analytical models in addition to geometry and general material information; examples are mechanical properties of materials or magnitude and distribution of wind and seismic loads. Hence, designers may need to make significant efforts to transform the building information model into an analytical model. These efforts could be automated through an IIE process to save modeling time and enhance accuracy and reliability of the model.

In other parts of the research not discussed here, a methodology was developed for documentation and automation of interpretations required through the process of
using BIM for engineering analysis (Ramaji and Memari 2015b). It is consistent with the IDM development methodology presented by buildingSMART International and could be used for design of different IIE specifications. This methodology is used in this paper for documentation of IIE designed for interpretation of basic structural analytical models from architectural building information models. IIEs that are designed using this methodology would have three layers: Concept, Generic, and IFC. In the Concept layer, the whole transformation process is divided into different exchange Units (eUs). In the Generic layer, the information units that the defined eUs should export to the output model are specified. In addition, input information units that are required as inputs for generation of outputs are also specifies in this layer. The last layer is related to binding the defined eUs to the IFC data structure. In other words, this layer specifies instances that need to be imported from the input file as well as the instance structure that should be used for representing the output information.

Since part of the input model could be exchanged without interpretation, the eUs set of IIE includes one direct-exchange Unit (dU), which specifies information units that should be directly forwarded to the output model without any changes. In addition to the dU, the eUs set may contain several interpretation Units (iUs), each responsible for one interpretation task. Based on the type of interpretation, the iUs could be classified into three categories: enriching, transforming, and simplifying. In the enriching iUs, information related to engineering analysis of the building would be added to the model; an example is live load assignment to the structural model in the information exchange from the architectural building information model. In representation of transforming iUs,
representation of an element would be changed to the most desirable format for receiver of information; an example being transforming extruded area representation of column to its structural finite element representation. In the simplifying iUs, an assembly of elements is transformed to a simplified analytically equivalent element. Simplifying a light-gauge steel wall to a structurally equivalent shell element in the structural model is an example for this type of transformation.

9.4. Interpretation Process

One of IIE’s applications is interpretation of structural analysis models from coordination models. Although the current version of Coordination View (V2.0) is not designed to include all the information required for interpretation of structural analysis models, it is still rich enough to be used for generation of basic structural models. In this section, an IIE specification is presented for automating the model transformation from building information models to their equivalent structural analytical models. The interpretation process is documented using the methodology presented in the previous section. This IIE is made up of one direct-exchange Unit (dU) and four interpretation Units (iUs) shown in their Generic layers in Figure 9-3. The dU lists information units from the input file that should be forwarded to the output file without any changes; whereas the iUs specify interpretations required in IIE and describe the way these
interpretations have to be handled. These units are discussed as separate bullet items in the following.

Figure 9-3. Units defined in the coordination model-to-structural D/A model IIE
• **dU**: As shown in Figure 9-3, the dU forwards project information to the output file. Information related to the unit system used in the model is one essential relative of the project object. Therefore, this information would also be forwarded to the output file along with the project object. In addition to the project and unit system information, the dU forwards information related to spatial element from site to building stories and generic spaces. Spatial aggregation relationships are also exchanged in this dU along with the spatial object Information.

• **iU-01**: This information unit translates linear building elements to their equivalent structural finite elements. Input objects are “IfcBeam”, “IfcColumn”, and “IfcMember” instances that represent geometry of these elements for coordination purposes. “IfcStructuralCurveMember” instances are output of this unit that represent the input linear element in the form of lines each connecting two vertex points with an assigned cross sections.

• **iU-02**: This iU translates building planar elements including walls and slabs to equivalent structural area elements. This unit inputs “IfcWall”, “IfcSlab”, or any instances related to sub-types of these concepts, and returns their corresponding “IfcStructuralSurfaceMember” instances. “IfcStructuralSurfaceMember” instances contain an attribute for specifying the thickness of area elements. Therefore, unlike linear structural elements, area elements do not need to have any cross section assignments in their topological representation.
• **iU-03:** In this interpretation unit, mechanical properties of materials declared in the input file are added to the output file. Since Coordination View is designed to provide information required for coordination, models in this view do not contain any information related to mechanical properties of materials. To add such information, first materials related to load bearing elements of the input file are identified. Then, mechanical properties of these materials required for structural design of related elements are extracted from the built-in library defined in the interpretation engine. Next, information is assigned to related “IfcMaterial” instances in the output file.

• **iU-04:** This last iU unit to run adjusts coordinates of building elements to resolve discontinuity among interpreted structural elements. This is carried out by merging points related to same connections and breaking down elements at their conjunction with other elements. These two adjustments integrate structural members for proper load transfer among load-bearing elements.

Since some of the defined units are fed to other units, the interpretation should be executed in an ordered manner. The overall workflow of structural model interpretation is illustrated in Figure 9-4. As shown in this figure, the process starts with forwarding instances related to project definition and unit systems used to the output file. It follows by forwarding spatial aggregation information, i.e., instances related to sites contained in the project, buildings aggregate sites, building stories aggregate buildings, and spaces aggregate buildings or building stories.
The IIE process continues by searching for building elements contained in building stories or spaces identified in the input file. Since IIE is designed to extract elements’ information from their swept solid representation, element instances found are checked for existence of their “IfcExtrudedSolid” instances or any of its sub-types in the input file. If the swept solid representation of an element is included in the input file, they would be transformed to “IfcStructuralCurveMember” or “IfcStructuralSurfaceMember” objects based on the type of input elements. The transformation process is carried out by iU-01 and iU-02, which are used, respectively, for transformation of linear and planar elements. The methodology used in this process is discussed in more detail in the following sections.
The adjustment of coordinates is the next step in the interpretation process. Final coordinates of structural elements are extracted by execution of several iUs including iU-01, iU-02, and iU-04. Hence, interpretation of coordinates is a multi-stage task. As a result, input to iU-04 is not only from the input file, but also from the outputs of iU-01 and iU-02 units. In this step, end or corner coordinates of transformed building elements are examined for association to a certain connection point, whose coordinates in that case would be modified to be identical with those of the connection point. Hence, all elements attached to a connection point would share one identical point in the output model. This shared point is located at one end of linear elements or at one corner of planar elements. In some cases, one linear element may be connected to the middle of another linear element. In terms of finite elements for such a model, if the linear element is not broken down at the connection point, these two elements cannot transfer their loads to each other. Therefore, breaking linear elements at their connection points is another adjustment that should be conducted for coordinates. Both of above-mentioned types of coordinate adjustments are carried out by the iU-04 process. The methodology used for coordinate adjustments is explained in more detail in the following sections.

The last stage has to do with enriching the output file by adding mechanical properties of materials assigned to load bearing elements. In this sages, which is carried out by iU-03 process, first, elements’ material assignment relationships are identified from the input file. Next, mechanical properties of these materials are extracted from the interpretation engine’s built-in library. Then, the enriched material information are written into IFC instances and are sent to the output file.
9.5. Representation Transformation and Material Information

Enrichment

In the designed IIE process, both interpretation units iU-02 and iU-03 are of the transforming iU type, where representation of considered objects is transformed to another form. These two iUs transform the swept solid representation of linear and planar building elements to their equivalent topological representation. The topological representation of linear elements consists of connecting two vertex points together with a line that contains a cross section property, while for planar elements boundaries of elements are defined using several edges forming a closed loop. In cases where edges are not curved, they are defined by connecting two vertex points. Hence, the core task of these iUs is interpretation of vertex points related to elements contained in the input file. Since input elements of these units are represented in swept solid format, the cross section of linear elements and the thickness of planar elements could be directly forwarded to their topological representation without any interpretation. The algorithm used for extraction of elements’ vertex point is discussed in the following.

Since Coordination View positions each element by one point and one orientation direction, the first step in the transformation process is to determine coordinates of the insertion point and the unit-vector of the insertion direction. Coordination View specifies coordinates of the insertion point of elements according to local coordinate system of spatial elements that contain elements. The spatial element itself may be located within another spatial element. For example, a building element may be located in a building
story, while the building story is located in a building, the building is located at a site, and
the site is designated for a project. All these spatial elements may have their own local
coordinate systems. Similar to building elements, each of these local coordinate systems
are defined by one insertion point and one insertion direction. The directions of local
coordinate systems are defined by specifying the directions of the x- and z- axes of the
local coordinate system according to the coordinate system of the immediate parent
object. Hence, one rotation matrix and one insertion point could be calculated per each
defined local coordination system.

The coordinates of an element’s insertion point in its parent spatial element
coordinate system is calculated by multiplying the local coordinates of the insertion point
to the rotation matrix of the local coordinate system, followed by adding the resulting
coordinates to the insertion point of the element in local coordinate system. To find the
coordinates of the element’s insertion point in the project’s global coordinate system, this
transformation is recursively executed to the point where the coordinate system of the
project is reached as the local coordinate system. The instance specifies that the local
coordinates of an element’s insertion points is located in the address shown by symbol
\(1\) in Figure 9-5. All addresses shown in this and similar figures are based on the data
structure of IFC2x3 and IFC4 schemas. In this figure, instance concepts are shown in
oval shapes and are in bold fonts, while the concept attributes are shown in rectangles
attached to their related concepts.
Figure 9-5. Swept solid representation of linear structural element in Structural View
9.5.1. Acquiring Information about Linear Elements

Having the insertion point of a linear element along with its length and direction, coordinates of vertex points required for its topological representation could be calculated. In swept area representation, the length of a linear element is specified as the depth of extruded area that is related to “IFcBeam”, “IfcColumn”, and “IfcMember” objects, which are extracted from the address shown by symbol ⑤ in Figure 9-5. In addition, the local directions of elements are found in the address number ④ depicted in Figure 9-5.

The cross section of a linear element in Coordination View is specified in the address shown by symbol ② in Figure 9-5. The default location of insertion point is the center-point of the cross section area. The offset of the cross section from its center-point is found in address ③ in the above mentioned figure. Since centerlines of linear elements should be shown in their topological representation form, coordinates of extracted vertex points are modified according to the cross section offset from its center-point.

9.5.2. Representation Transformation of Linear Elements

More than two vertex points are required to define topological representation of planar elements. These vertex points are area element corners. Coordinates of one of these vertex points could be extracted from the insertion point, which along with their related offset distances could be extracted from the same address path shown for linear elements. Depending on shape, Coordination View may present the swept area of these
planar elements using two main types of profile objects: “IfcRectangleProfileDef” and “IfcArbitraryClosedProfileDef”. Extraction of coordinates from these two types of profile instances are discussed separately in the following.

“IfcRectangleProfile” instances are used in Coordination View for defining dimensions of rectangular planar elements. The insertion point of planar elements is the center of the rectangle and the offset is measured from this point. Therefore, the insertion points are modified accordingly to find the first corner of the rectangles. This is done by subtracting half of related rectangle’s dimensions from the coordinates of the insertion point and adding the offset vector to the resultant coordinates. Having dimensions of a rectangle along with its orientation and coordinates of one of its corners, coordinates of other three points of the element could be extracted. The dimensional data of these elements are extracted from the path shown by ⑥ and ⑦ symbols in Figure 9-5.

The planar element could be a wall or a slab. The “IfcRectangleProfileDef” objects assume the extrusion direction to be parallel to the Z-axis, unless another direction is specified. Therefore, if the planar element is a wall, the rectangular area represents the element’s cross section, which is represented by one edge line in the Structural View, and therefore, only two of the points extracted from a rectangular section could be used for topological representation the related element. The other two points are used for finding the thickness of the wall by measuring their distance from the points used for defining the area’s edge. Having the height of the wall and knowing that the wall’s height direction is toward the local Z-axis or any other custom direction, the
coordinates of the other two lines required for defining the finite element area could be calculated. The height of the wall could be found in the same address used for finding the length of linear elements.

If the planar element represents a slab, the corners’ coordinates of its rectangular cross section could be directly used for defining its topological representation. In this case, the thickness of area elements is extracted in the same path used for finding the wall height and the length of linear elements. For non-rectangular planar elements, Coordination View uses “IfcArbitraryClosedProfileDef” instance object to specify the element shape. This object defines boundaries of an area element. If the input area is a polygon, boundaries are made up of a series of points in the order they are connected together. These points are listed in the point set defined in the address shown by symbol in Figure 9-5. The coordinates of the points belonging to this set are used directly to define the vertex points required for topological representation of the transforming element. The thickness of the planar element could be found in the same path used for finding the length of linear elements.

9.5.3. Representation of the Transformed Elements

The extracted coordinates related to building elements should be written into the output IFC file in the Structural Model View format to make it accessible by structural D/A software. As shown in Figure 9-6, generation of topological representation of linear structural elements is carried out by defining an edge object made up of two vertex
points, and inserting the interpreted coordinates in paths marked by ② and ③ symbols. In addition, the coordinates of the element’s local position is placed in the path marked by symbol ①.

Figure 9-6. Topological representation of linear structural element in Structural View

The format used in Structural Model View for topological representation of planar elements is shown in Figure 9-7. In this view, a planar element is defined by a surface element while the surface element is defined by its boundaries. The boundary is
composed of several edges that form a closed loop. Similar to the linear element, each linear edge is made up of two vertex points. The defined edges are placed in the path marked by symbol ③ in Figure 9-7. The local positions of planar elements are specified in the same path used for representation of linear elements, which is marked by symbol ①. In addition, as shown in this figure by symbol ②, the thickness of the planar element is one of the attributes of the “IfcStructuralSurfaceMember” instance concept. Therefore, the extracted thickness data are assigned to this attribute in the related objects.

Figure 9-7. Topological representation of planar structural element in e Structural View
In Coordination View, the element’s material information is specified by assigning an “IfcMaterial” instance to related element’s instances. The mechanical properties interpreted from the material’s name are assigned to “IfcMaterial” instances to make it accessible by the structural D/A tool importing the output file. Due to changes made in the IFC schema from version 2x3 to 4, modeling these properties in the form of IFC instances may vary based on the IFC schema version used for generation of the output file. As shown in this figure, there are two ways in IFC 2x3 files to define mechanical properties of steel and concrete materials, one is specific to these two types materials (Left) and the other is a generic general purpose view for defining properties of any types of materials (Right). In the first method, the material is related to one of the “IfcMechanicalConcreteMaterialProperties” or “IfcMechanicalSteelMaterialProperties” instances that contain a list of attributes for mechanical properties. In the second method, several property sets are assigned to the “IfcMaterial” instance using the “IfcExtendedMaterialProperties” instance object. Each of these property sets specifies one mechanical property of the related material. In both of these methods the material instance is related to elements using “IfcRelAssociatesMaterial” instances.
Figure 9-8. The view used for material and section properties in output files observing IFC 2x3

In IFC 4 schema, assignments of the cross section and material are combined into one relationship concept. In IFC 2x3 files, cross sections of elements are assigned to linear elements simply by relating one of the sub-types of the “IfcProfileDef” concept to related elements using an “IfcRelAssociatesProfileProperties” instance, whereas in IFC 4, both the cross section and the material information are assigned to related elements simultaneously by “IfcRelAssociatesMaterial” IFC objects. As shown in Figure 9-9, definition of these two information units takes place at “IfcProfileMaterial” instance.

a) View specific to steel and concrete  
b) General purpose material definition view
objects. In Version 4 of the schema, a profile section along with the related material information can be defined in an “IfcMaterialProfileUsageSetUsage” instance object. This type of object refers to “IfcProfileDef” and “IfcMaterial” instance for specifying, respectively, the cross section and material information. As depicted in this figure, additional information about structural properties of a material are listed in an “IfcMaterialProperties” instance where the “IfcMaterial” instance is related to several “IfcProperty” instance objects. Specifying the profile and material information is not optional in the “IfcProfileMaterial” instances. Hence, in the case of a planar element, the rectangular cross section of the element is stated as the cross section in the “IfcProfileMaterial” instance.
Figure 9-9. The view used for material and section properties in output files observing IFC 4
9.6. Connectivity Adjustment

As discussed in previous sections, models in Coordination View are unusable for structural design purposes without interpretation. Models in this view lack integrity between building elements, although it could be addressed by modifying elements’ coordinates through an IIE process. The connectivity adjustment designed in this tool includes two main parts being executed in sequence: merging points at connection points and breaking linear elements at their internal connection points. To recognize connections from the information included in the input coordination model in the automated IIE process, a special point class is defined to store information required for interpretation of connections. Definition of this class named by Merge Point along with its declaration in the C++ programming language is shown in Figure 9-10. A merge-point is made up of five attributes: 1) coordinates of the point, 2) index of the “IfcCartesianPoint” instance generated based on the point in the output file, 3) indexes of elements from the input file that are connected at the merge-point, 4) indexes of interpreted structural members in the output file connected in the merge-point, and 5) the distance over which the merge point is sensitive to for recognizing the connections. Attributes of a merge-point may change during connectivity adjustments. Keeping track of the index of elements connected to the merge-point makes it possible to access and modify the related output file instances in such cases.
9.6.1. Connectivity adjustment of Connecting Points

Elements in Coordination View are not connected at their centerline but at the point or surface their body meet each other. Therefore, as the overall cross section dimensions of the connecting elements increase, the distance between their connecting points also increases. Hence, if two points are located at a distance smaller or equal to overall cross section dimensions of related elements, they could be considered as connected elements and be merged together. Therefore, for elements whose coordinates are not adjusted yet, the coverage distance of points are considered to be equal to the overall cross section dimension of the related element. In linear elements, this is equal to the maximum width or height of the cross section, while in planar elements it is equal to the thickness of the element. Once two points are merged together, a merge-point is generated based on the adjusted coordinates of merging points. The coverage distance of the generated merge point is equal to the maximum of the coverage distance of the points.
merged. Since connectivity adjustments take place in the iU-4, which is being executed after transforming elements to their topological view, instances of output elements are already generated at the time of adjustments. Therefore, if coordinates of one end of connecting elements are modified, instance objects related to these elements should be modified accordingly. List of connecting elements stored in the fourth variable of merge-point objects could be used in this case to access related instance objects that need to be modified.

When two points are merging together, many different approaches may be taken for finding the coordinates of the generated merge-point. One may use the average of the connecting points as merge-point coordinates, while another option is to randomly choose the coordinates of one of the connecting points. One of the problems associated with these two approaches is that vertical columns may be inclined after transformation. Having several elevations for points on the floor of a certain story is another problem that could happen based on choosing these approaches. Hence, a system is needed to be designed to prevent distortion in the transformed model.

This system is designed by ordering transformation of different types of building elements and prioritizing importance of keeping their coordinates unchanged. In such a system, no new coordinates are defined as connecting points are merged to the coordinates that belong to the element with higher priority. To ensure vertical elements remain vertical, the process starts with vertical building elements, including columns and walls. If a wall or column is connected, size of the column determines length of the wall; therefore the coordinates of columns are considered to have higher priority.
Connectivity adjustment of horizontal elements including beams and slabs is the next step in this process. For the same reason mentioned for columns and walls, priority of the beams is considered to be at a higher level. The last step is connectivity adjustment of inclined elements that are represented by “IfcMember” instances in Coordination View. These elements have the lowest priority in the designed system as they may be connected to the conjunction point of vertical or horizontal elements. Therefore, their coordinates are adjusted after coordinates of vertical and horizontal elements are adjusted.

During the adjustment process, connectivity of all transformed building elements is investigated by checking whether any merge-points lay within coverage distance of the element’s points. If this is the case, the coordinates of the related point are replaced by the identified merge-point, unless a new merge-point is generated based on the point’s coordinates. Therefore, at the end of the adjustment process, all elements are defined by merge-points. Figure 9-11 illustrates two points being checked against one merge-point. In this figure, the merge-point is shown by a back dot, while its coverage area is represented by an orange circle. P1 and P2 are points whose coordinates are being adjusted. Since point P1 lies on the coverage area of merge-point MP, it would be merged to point MP. On the other hand, even if the coverage areas of point P2 and the merge-point MP have overlap, point P2 would not be merged to point MP because none of these points lie on the coverage area of the other one. In this case, a new merge-point would be defined based on the coordinates of point P2.
9.6.2. Connectivity adjustment of Point-to-Line Connections

Since only coordinates are considered in Coordination View, in this model view, building elements are not broken down at points they have to transfer loads to other elements. Therefore, after adjusting the coordinates of elements, connection points at the middle of linear elements should be identified and the geometry of related elements modified accordingly. These connections are found by examining the distance of merge-points from linear elements as shown in Figure 9-12. A connection point exists at the
middle of the linear element only if the measured distance is smaller than the coverage distance of the linear element or merge-point. The coverage distance of a linear element is the maximum overall dimension of its cross section. Unlike connectivity adjustment of the connecting point, the coordinates of the merge-point are modified here if a connection is recognized. The new coordinates of the merge-point lie on the linear element where the end of the connecting element meets the linear element. As shown in Figure 9-12, the location of this point is the closest point of the linear element to the connecting point (point D). The coordinates of such point are calculated according to the coordinates of one end of the linear element and its distance from the closest point (distance AD). This distance can be calculated by projecting the line that connects the connecting point to the end of the element (line AC) on the linear element. Once location of the closest point is found, it will be checked to see if the point lies between two ends of the linear element. If this is not the case, it means that elements are not connected, although they may meet each other if extended. This condition may happen when two elements are closely placed near each other or if two elements are placed in the same direction and are separated by another element. An example for the latter condition is two beams located on two different sides of a column in the same direction.
Once connection along with new coordinates of the merge-point are recognized, the coordinates specified in the instance generated based on the connecting merge-point should be modified accordingly. The index of this instance is recorded in the “relatedInstance” variable of the merge-point object. Changing coordinates of this instance leads to changing the end coordinates of elements connected to the merge-point. The next step is breaking the linear element at its closest point to the connecting point where the modified merge-point is located. Each of the resulting linear elements is connected to one end point of the un-divided linear elements (point A or B) at one side and is connected to the modified connecting merge-point (pint D) at the other side. After the connectivity adjustment, connecting elements are common at one merge-point, and as a result, they would be able to transfer load between each other at this point.
9.7. Validation

The interpretation mechanism designed for interpretation of the structural model is automated by development of a tool, which is then used to implement a case study to test functionality of the designed interpretation mechanism and supporting tool. This tool has five main modules, each responsible for executing a certain task: reading module, information query module, interpretation engine, IFC object generation model, and exporting module. The reading module reads and parses information to IFC objects, while querying function designed in querying module conducts search within acquired information. The interpretation module uses querying functions to extract information required for interpretation and also uses IFC object generation modules to write interpreted information into IFC object. At last, exporting module translates IFC objects resulting from interpretation and direct-exchange Units to IFC file instances, adds proper header to the file, and generates the output for interpreted IFC file.

To perform the case study, a two-story office building test model was used that was developed by buildingSMART alliance and represented in IFC Coordination View V2.0. The structural system of this building is steel moment frame with concrete slab on steel joist floor system. The building information model of this building along with the structural analytical model, which is generated from the building information model without interpretation, are shown in Figure 9-13. This model contains both types of main structural building elements including linear elements such as beams and columns, and planar elements such as slabs. As illustrated in this figure, majority of finite linear and area elements are not integrated at connection points, while each of the elements at floor
level has a different elevation. On the contrary, as shown in Figure 9-14, the structural model generated through interpretation process is a fully integrated and structurally correct model. As shown in this figure, all elements are connected to each other at their ends/corners, and beams that are connected to one another at their middles are divided at connection points. Comparing analytical models generated from the building information model with and without interpretation, the number of elements increased from 208 to 226, while the number of points decreased from 380 to 139. Breaking beam elements into multiple elements at their connection points with other elements is the reason for increase in the number of elements. In addition, the size of the model in Coordination View is reduced from 14.9 MB to 212 KB in Structural View.

Joists contained in the building are standard K-type joists that are represented by the boundary face of their elements in Coordination View, hence their structural properties and details cannot be extracted from the model in this coordination view. The analytical model shown in Figure 9-13 is generated without interpretation and can only show approximate placement of joists without a proper section property assigned. Since IIE presented in this paper interprets elements from their extruded area representation, the interpreted structural model in the case study does not contain floor joists. Of course, many designers prefer not to include joists in the model as such elements are designed based on their specification tables; inclusion of such elements in the structural model make the model significantly large without having much added value. However, another iU could be added to the design IIE to support information exchange related to joist
elements by interpretation of their structural properties from their name and a built-in library.

Figure 9-13. Test Model in Coordination View (Top) and related structural model generated without interpretation (Bottom)
Figure 9-14. Structural model interpreted from building information model

9.8. Summary and Concluding Remarks

This paper has presented a methodology for interpretation of structural models directly from the building information models in coordination view. In addition, a tool has been developed and presented in this paper for automating model transformation process defined in the presented methodology, which is also validated in the presented material. This mechanism and supporting tool can input the model in Coordination View V.2 and transform it to a structurally equivalent structural mode. Incorporating Industry Foundation Classes (IFC) as the open standard file format for BIM information exchange for both input and exporting information makes interpretation mechanism and supporting
tool highly interoperable. The methodology for Interpreted Information Exchange (IIE) documentation, which was developed by authors as another part of the research not presented here, has been used for design of the presented mechanism. Three main model transformations are identified and required for generation of basic structural analytical models from building information model: representation of model transformation, material information enrichment, and connectivity adjustment. The first one is related to changing the shape of representation of the elements to their equivalent topological representation. Enriching the model with additional mechanical properties of the structural material according to their name in the input model is the second transformation. The third one is related to merging end/corner points of elements at their connection points with other structural elements. The designed interpretation mechanism is made up of one direct-exchange and four interpretation units. The first interpretation units is responsible for converting swept solid representation of linear elements to their topological representation; the second interpretation unit does a similar transformation for planar elements; the third interpretation unit enriches the output model by adding the materials’ mechanical properties required for structural design of elements; and the forth units integrates the transformed elements by adjusting their coordinates.

The designed mechanism along with supporting tool developed for automating model transformation process is an examples of implementation of IIE concept. Implantation of this automated interpretation process showed how interoperability between BIM and design software could be enhanced by interpretation of the required information from building information models in a general-purpose model view such
Coordination View. In addition, it illustrated how implementation of IIE concept could reduce modification efforts required for preparing the exchanged model for intended purposes. Taking the same approach, other mechanisms along with their supporting tools could be developed for automating model transformations required for other engineering analytical modeling directly from building information models.

9.9. References


CHAPTER 10:  Design of an IIE for Structural Design of Modular Buildings

10.1. Introduction

As for any relatively new structural systems, unknown behavior and lack of sufficient specific tools, design guidelines, and code requirements are main reasons that make designers or contractors reluctant toward adopting multi-story modular building systems in their projects. In order to enhance reliability of this system and achieve its potential as an alternative for some site-built construction, challenging issues for certain structural aspects have to be carefully addressed. Proper analytical modeling of the structural system is the first step needed for evaluation of their structural performance. Unfortunately, current structural finite element modeling tools are not designed or configured to support modular construction, and if used will result in inefficient analytical modeling of these buildings. The reason for this inefficiency has to do with the difference between structural system of conventional site-built construction (e.g., made up of beams, columns, slabs, walls) and how modular units are assembled in factory and
a modular building erected by stacking and attaching the modular units in a pin-jointed fashion, which leads to modeling the latter challenging.

To address the aforementioned needs and issues, a mechanism is designed in this paper for simplifying the structural models of modular buildings into an equivalent modeling format that is used for structural analytical modeling of site-built construction. This conversion makes the current finite element modeling tools efficiently usable for analysis and design of modular buildings. The simplification mechanism is documented as an Interpreted Information Exchange (IIE) process that is designed for transformation of the BIM model of a building’s modules to the analytical model of the building. Similar to the tool developed in this research and discussed in Chapter 7, one can use a BIM information extraction platform, developed in this study and discussed in Chapter 6, to automate this IIE process. In the following sections, the process of creating analytical structural model of modular buildings is discussed along with reviewing the interpretation Units (iUs) required throughout this process.

10.2. Structural Modeling and Design Complexities

Due to the modular nature of components and connections of modular buildings, structural modeling and analysis of these buildings are more complicated compared to the conventional site-built constructions. Some of theses complexities are discussed in the following.
In structural modeling of conventional site-built buildings, one usually assumes that the centerline of all members at a given floor level are located in a single plane, that is the centerline of story level. However, in some types of modular buildings such as 3D systems, as shown in Figure 10-1, the bottom members of the upper modules do not lay in the same plane as the top members of the lower modules. In other words, there is an elevation difference between the floor of a given story and the ceiling of the story below. This difference is more pronounced when the designer wants to model the interface of the podium and the modules in hybrid-3D system. There are two approaches to address this challenge in the structural model. One being reflecting this difference by considering two different closely spaced planes for floor and ceiling of these separate modules and integrating upper and lower joints of these elements by linking their joints at sufficient number of points. An example of the finite element model to represent the diaphragm in such systems is depicted in Figure 10-1. The other solution is combining the two planes into one plane of elements with equivalent structural section and material properties. For example, in post bearing systems, if the cross section and layout of the floor and ceiling elements are identical, these two layers could be simplified in a layer with the same layout and doubling the cross sectional properties of the elements. The second approach is taken here for interpretation of the structural analytical model of these buildings from the BIM model.
Figure 10-1. An example of connection of the floor finite element model for an upper module to the ceiling level of the lower module.

When a multi-bay multi-story building is loaded in the lateral direction, the columns or walls in adjacent bays tend to slip with respect to one another as shown in Figure 8-2 (left) if sufficient shear transfer mechanism is not provided to resist the vertical shear at the vertical interface between modules. The connections of horizontal and vertical elements in conventional site-built construction of for example frames are generally designed for combination of gravity and lateral loads, thus creating the necessary load path continuity between horizontal and vertical elements, with behavior as shown in Figure 8-2 (right). Due to discontinuity between modules in modular building construction, the modules must be connected to one another in the vertical direction to resist vertical shear forces (caused by lateral loads) between modules. Figure 10-2 shows the difference in deflection modes between the two systems when the adjacent vertical load carrying systems are not tied to one another for vertical shear transfer and when they
are. In the left hand side figure, the adjacent modules slightly slide vertically with respect to one another because of the vertical flexibility of attachments between adjacent components, or lack of attachments, while the figure on the right shows the vertical shear transfer through rigid connections between such components.

Figure 10-2. Vertical Shear transfer between stacked units: left side) no vertical shear transfer or partial transfer due to flexibility of connections; right side) coupled at vertical joints though rigid connections

In addition to the specific challenges that exist for structural analytical modeling of multi-story modular buildings, there are some issues and shortcomings that are associated with design of these modular structures. As an example, in code-based seismic design of these buildings, one needs to determine the value of response modification
factor (R). As for any new type of structural system, it is a challenge to determine a reliable R-factor value unless an extensive analytical study or experimental testing is carried out. A further challenge in determination of R factor is that the value of the factor would vary for different types of modular building systems that have different lateral load resisting systems. Evaluating safety and robustness of the structures is another important challenge that designers must take into consideration in the design of these buildings (Lawson et al, 2008). There are numerous situations that could lead to failure of some modules; examples are faults in the factory where modules are fabricated and assembled, damages during transportation and erection, failures resulting from combustion or explosion, plain collisions, and terrorist attacks. Accordingly, the designer should design the building redundant enough to prevent repetitive failure or progressive collapse of the structure for cases in which some of modules may fail. In addition, designers should consider residual forces imposed to structural members by adding some internal forces to the module elements or alternatively increase the factors of safety.

10.3. Design Workflow Using IIE

The process of generating structural analytical model could be divided in three stages: Preparation, Interpretation, and Integration. In the preparation stage, BIM models of the modules are developed. Next, the structural analytical model of the building’s modular part is interpreted by merging the structural models of each pair of adjacent
module walls or each pair of adjacent module floor and ceiling. Figure 10-3 shows an example of for interpreted structural finite element model of a post-bearing model. In this figure, the structural walls elements are simplified by shell elements, while non-structural elements are eliminated and size of the opening is modified to eliminate parts of the wall element that do not contribute to the structural systems. Finally, at stage three, the structural model of the building’s site-built parts, such as shear walls and podiums, is added to the model of modular parts.

Figure 10-3. Structural D/A model generation of a module

Depending on the type of model transformations needed to be executed in the second stage, the interpretation process could be divided in three sequential steps: Transformation, Refinement, and Assembling, while different sets of the iUs need to be implemented in each of these steps. Figure 10-4 shows the sequence of the model
generation stages along with their relationships with the interpretation steps. In the following, these stages are explained in more detail:

**Transformation Stage:** According to the PAM, modules have four main subassemblies: walls, floor, ceiling, columns, and voids. In the transformation stage, the structural walls are transformed to simple shells (if there is any), the floors are transformed to a simple joist-beam-shell assembly, the ceiling is simplified to a beam-joist assembly, the columns and braces are extracted without any changes, and the voids are left for the Refinement stage. In other words, a simple raw structural model of the module is generated at this stage and the rest of the components are eliminated through a filtration.

**Refinement Stage:** This stage adds the required features to the raw structural model in preparation for analysis. This stage may include some modifications such as insertion of voids in shell elements and mesh generations, or may contain some assignments such as dead and live loads assignment and defining diaphragms at the floor level. Both the output of the Transformation stage and the architectural BIM IFC files are used for interpretations in this stage. From this stage on, the architectural BIM IFC file is not used anymore. An example output of this model is shown in Figure 10-3.

**Assembling Stage:** In this stage, modules are merged together at their interfaces. The adjacent walls of attached modules should be merged together to have one single elevation for each story and only one row of structural elements at each interface of adjacent modules. Modules may interact with each other horizontally and vertically. Modules in a same story interact vertically, while modules stacked on top of each other
interact horizontally. Hence, two iUs are defined to merge the modules in these two types of interaction surfaces.

<table>
<thead>
<tr>
<th>Modular Buildings Structural D/A Model Preparation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
</tr>
<tr>
<td>Prepare of architectural BIM model of the modules</td>
</tr>
</tbody>
</table>

Figure 10-4. Process map of the Modular Buildings Structural D/A Model Preparation using the proposed iEs

10.4. Required Interpretation

To perform information transformations required for interpretation of structural analytical models of multi-story modular buildings defined in the previous section, 13 iUs are defined in this research, which are listed in Table 10-1. The execution order of these iUs throughout different steps of the Interpretation Stage is listed in Figure 10-4. The methodology used for definition of the iUs is the method proposed in the previous chapter for design of IIE processes. To document the IIE process for generation of the structural analytical model of a building’s modular part, initially the required
interpretations are recognized in the first layer called the Concept Layer and documented in different interpretation Units (iUs). Next, physical and spatial elements of the modules along with their attributes and parameters, which are required for the interpretation, are recognized in the BIM physical model. Then the same is done for the analytical model. By this means, information from these two data sets are mapped together in different iUs in the second layer called Generic Layer. The proposed iUs could be programmed to automate model transformation in the process of interpreting structural analysis model of modular buildings from their BIM model. To do so, the platform developed for implementation of the IIE could be employed along with the presented iUs after being bounded to the IFC data schema. In the following, the developed iUs are shown in the Generic Layer and are briefly discussed individually:
Table 10-1: Defined Unis for Multi-Story modular buildings in Concept level

<table>
<thead>
<tr>
<th>Unit ID</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>dU-01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>iU-01</td>
<td>Wall Components</td>
<td>Wall Shell</td>
</tr>
<tr>
<td>iU-02</td>
<td>Wall Shell</td>
<td>Meshed Wall Shell</td>
</tr>
<tr>
<td>iU-03</td>
<td>Weights of Wall Components</td>
<td>Wall Dead Load</td>
</tr>
<tr>
<td>iU-04</td>
<td>Floor Components</td>
<td>Simplified Floor</td>
</tr>
<tr>
<td>iU-05</td>
<td>Floor Shell</td>
<td>Meshed Floor Shell</td>
</tr>
<tr>
<td>iU-06</td>
<td>Weights of Floor Components</td>
<td>Floor Dead Load</td>
</tr>
<tr>
<td>iU-07</td>
<td>Module Occupancy</td>
<td>Floor Live Load</td>
</tr>
<tr>
<td>iU-08</td>
<td>Rigid Floor</td>
<td>Floor Diaphragm</td>
</tr>
<tr>
<td>iU-09</td>
<td>Ceiling Components</td>
<td>Simplified Ceiling</td>
</tr>
<tr>
<td>iU-10</td>
<td>Ceiling Components Weight</td>
<td>Ceiling Dead Load</td>
</tr>
<tr>
<td>iU-11</td>
<td>Wall Voids</td>
<td>Wall Shell Voids</td>
</tr>
<tr>
<td>iU-12</td>
<td>Horizontally Assembled Modules</td>
<td>Merged Modules</td>
</tr>
<tr>
<td>iU-13</td>
<td>Vertically Assembled Modules</td>
<td>Merged Modules</td>
</tr>
</tbody>
</table>

**iU-01:** This interpretation unit converts a module wall into an equivalent structural finite element area element. In other words, it changes representation of a module’s wall assembly, but not that of the braces placed in the wall. This iU in its Generic Layer is shown in Figure 10-5.
**iU-01**: This iU generates the mesh on shell areas at the location of the middle columns located within the module’s wall. This type of meshing makes the load transfer possible between columns and wall shells. The Generic Layer of this unit is illustrated in Figure 10-7.
**iU-03:** In this unit shown in Figure 10-7, the weight of wall components are calculated and assigned to shell area elements representing the wall. This iU considers wall elements ignored through execution of the iU-01.
**iU-04**: This iUs simplifies a module floor assembly to its finite element representation made up of middle and perimeter beams, joists simplified by linear finite elements, and a floor shell representing the concrete slab or steel deck of the module. This iU is represented in its Generic Layer in Figure 10-8.

![Figure 10-8. The iU-04 in Generic Layer](Image)

**iU-05**: This iU meshes a floor shell area along the floor assembly’s beams and joints to make the load transfer possible between different elements of the floor assembly. This iU in its Generic Layer is shown in Figure 10-9.
**iU-05:** In this iU, the dead loads of the components of a floor assembly are calculated and assigned to the shell area representing the floor. Figure 10-10 illustrates this unit in its Generic Layer.

**iU-06:** In this iU, the dead loads of the components of a floor assembly are calculated and assigned to the shell area representing the floor. Figure 10-10 illustrates this unit in its Generic Layer.
**iU-07:** This iU that is shown in Figure 10-11 transforms a floor’s occupancy information to live load that should be assigned to the shell area representing the floor. The magnitude of the live load is calculated based on the applicable building code.

![Figure 10-11. The iU-07 in Generic Layer](image)

**iU-08:** In this iU, a diaphragm is assigned to the joints related to shell areas and linear finite elements representing a module’s floor assembly. This unit in its Generic Layer is shown in Figure 10-12.

![Figure 10-12. The iU-08 in Generic Layer](image)
iU-09: This iUs shown in Figure 10-8 simplifies a ceiling assembly to its finite element representation where beams and joists (if there is any) are represented by linear finite elements.

Figure 10-13. The iU-09 in Generic Layer

iU-10: This iU assigns dead loads resulting from non-structural elements of ceiling assembly to its beams and joists. The Generic Layer of this interpretation unit is shown in Figure 10-14.
iU-11: This iU modifies the meshed shell area elements representing a wall assembly based on the location of the openings placed on the wall. This modification includes meshing the areas along the edges of the openings and eliminating areas placed at the location of these openings. This iU in its Generic Layer is shown in Figure 10-14.
iU-12: This interpretation unit merges two adjacent modules vertically. In other words, it inputs the interpreted structural models of the sides of two adjacent modules and interprets their merged equivalent structural model. Each module side may contain several elements from wall, ceiling, floor, and column parts of the modules. This iU is represented in its Generic Layer in Figure 10-16.
Figure 10-16. The iU-12 in Generic Layer
iU-13: This unit merges horizontally two modules stacked on top of one another. In other words, it inputs the interpreted structural models of the two modules, connects their walls and columns, and interprets the equivalent structural model of the integrated floor and ceiling assemblies. The Generic Layer of this iU is shown in Figure 10-16.
Figure 10-17. The iU-13 in Generic Layer
To better illustrate the designed workflow, the building information model and structural analytical model of a two-story modular building at different stages of the process are shown in Figure 10-18 and Figure 10-19.

Figure 10-18. Building information model (top) and interpreted structural model (bottom) of a module of the building
Figure 10-19. Building information model (top) and interpreted structural model (bottom) of the modular building
10.5. Summary

In this chapter, a mechanism has been defined for transformation of BIM models of multi-story modular buildings into their equivalent structural analytical models. The methodology presented in the previous chapter for development of Interpreted Information Exchanges (IIEs) was used to document the interpretation process. The interpretation process presented in this chapter could be categorized into three sequential steps: Transformation, Refinement, and Assembling. A series of interpretation Units (iUs) are defined for each of these steps to document transformation required for execution of these steps.

Using the proposed IIE, structural D/A model preparation of a modular building includes three stages: Preparation, Interpretation, and Integration. In the Preparation stage, separate IFC BIM models of the building along with structural model of the site-built parts are prepared. After implementations of the iUs in Transformation, Refinement, and Assembling steps of Interpretation Stage, structural elements related to the site-built components would be added to the structural model of the assembled modules in Integration stage. By completion of this stage, structural model of the building would be ready for analysis and design in the last stage.
CHAPTER 11: Summary and Conclusions

In this research, an integrated framework was developed for information management of multi-story modular buildings, while the focus was on structural aspect of such buildings. This framework addresses information modeling, information exchanges, and digital representation of modular buildings. In addition to the outcomes of this research, which directly serve modular building projects, the information management methodology and tools developed in this research can be also deployed for other type of construction in the building industry. In the following, the main outcomes and major contributions of this research are discussed separately.

11.1. Outcomes

The key deliverable of this study is a specialized modular object-oriented Building Information Modeling (BIM) framework that is unique in its formulation and allows it to be adopted by modular companies and building information management tools. The function of the framework is such that it will allow users to adopt transformative and novel industrial design and information management approaches in
multi-story residential, commercial, and mixed use buildings that can deploy modular construction. The research presented has several important outcomes: 1) Multi-story modular buildings’ Product Architecture Model (PAM), 2) An Information Delivery Manual (IDM) and a Model View Definitions (MVDs) for modular construction, 3) A tool and approach for interpretation of analytical structural model from a building information model, and 4) A mechanism for interpretation of the analytical structural model of multi-story modular buildings directly from their building information models. The outcomes of this research are discussed individually in more detail in the following sub-sections.

11.1.1. Product Architecture Model

The PAM is an essential component for industrial design of such buildings and fills the gap that exists between product-based nature of information models in production industry and project-based nature of information modeling frameworks in construction industry. As for any other industrial product, modular buildings are composed of assemblies, subassemblies, and elements. The PAM contains the hierarchy of product assemblies along with associated attributes and properties of each component. In the PAM, objects and their attributes that are categorized based on Levels of Detail (LoDs) are mapped to their functionalities and limitations and contain constraints and relations between different entities in an organized way. The PAM is designed to be generic to address different systems of multi-story modular buildings. In addition, the
PAM is designed flexible to be expandable for capturing more information in certain areas. For example, the PAM presented in this study is expanded to include the structural aspect, which makes it usable for structural design purposes in addition to the general information it contains for design coordination among different disciplines.

The generic and expandable nature of the PAM makes it an open information model and provides a shared information platform that could be used and updated by industry practitioners and scholars who work on different aspects of modular building. The hierarchy of developed PAM is consistent with the way a building and corresponding modules are broken down into smaller sub-assemblies in manufacturing plants. In addition to hierarchy, many LoDs are defined in the PAM to provide a common language for requesting a model with a certain level of development and a scale for measuring it. To enhance flexibility of the information model for supporting new products and assemblies, objects that belong below a certain level of detail are defined in abstract form. The PAM addresses the modular building industry’s need for an information model as a base for product design and product information management. Similar to the Manufacturing Industry, the PAM could be used for development of product lifecycle and information management frameworks to increase productivity in modular building projects.
11.1.2. Information Delivery Manual and Model View Definitions for Modular Buildings

The IDM developed in this research provides the modular building industry with an information framework, which identify workflow of design activities in multi-story modular building projects and specify information flow between these activities. To address the product-oriented design approach in multi-story modular building projects, the bSI’s methodology for development of IDMs is extended to capture the hierarchy of building components and their related attributes. The bSI’s methodology is not designed to capture the relationship between different information units considered in the framework. Hence, this method is extended by adding a new component (the PAM) to the information framework. Since the PAM encapsulates all physical components of different modular building systems along with their attributes, it acts as a repository for the considered information units and contains the relationship and rule-sets that should be followed in the framework. As a result of extending the bSI’s methodology, the comprehensive anatomy of the modular building is observed in the developed information framework. Integration of the PAM with the information framework provides a holistic information management framework that clearly specifies the information units of the product’s components along with the project stage in which they should be designed.

The developed information framework considers the whole design process of these projects and is extended to capture more detailed information for structural design
tasks and information exchanges. The generic nature of the developed information model and the PAM makes it possible to extend the framework to support any other BIM uses in more details. In addition, since encapsulated information in the PAM are labeled based on the level of detail, the Level Of Development (LOD) specification of the defined exchange requirements is already embodied in the developed information framework, and if required, it could be utilized for defining a custom-defined project-specific exchange requirement. The information framework is made up of two main parts including process map and Exchange Requirement (ER) specification. The process map specifies an overview of activities and related information exchanges. Supplementary information about each of these activities and exchanging information are provided in the Task Unit (TU) and Exchange Model (EM) definitions, which are documented in Appendix A. The ER specification, which is included in Appendix B, is the link between the information exchange standard and the PAM. It clearly specifies the subset of information units from the PAM that should be included in each EM.

In addition to facilitating execution of BIM in multi-story modular buildings, the proposed information framework lay the basis for standardization of digital representation of BIM models in order to address interoperability. This dissertation has discussed how current MVD standards developed for conventional construction could be extended to support multi-story modular buildings. Site-built and modular buildings are made up of the same building elements, and multi-story modular buildings may contain several site-built parts. Hence, extension of the current MVDs for capturing special information related to modular buildings is identified to be a more reasonable solution as
opposed to developing new separate MVDs for these projects. Taking this approach yields an extended MVD that is rich enough to support both site-built and modular constructions. In addition, since several tools have already implemented the currently available MVDs, this approach would speed up implementation of the extended MVDs. The hierarchy of objects in modular and site-built buildings is different; modular buildings have three more types of objects: building parts, modules, and module parts. To address these objects along with their related components and attributes, several model views and concepts must be defined to extend the current MVDs for modular construction. Based on the hierarchy of modular buildings in the PAM and information exchange requirements recognized in the developed IDM, these model views are recognized to be the following: Building Part, Volumetric Module, Planar Module, Generic Module Part, Module Wall, Module Floor, Module Ceiling, and Module Column Assembly. To extend the MVDs based on the newly defined model views, three required modifications are identified: 1) Modifying the structure of MVD by relating the high-level model views of the current MVD to the ones added, 2) Connecting the added model views to the low-level model views of the current MVD, and 3) Extending the added model views for capturing the information related to the BIM Use targeted by the MVD. Extension of current structural MVDs (e.g., VBL-001 and VBL-002) has illustrated how additional model views can be employed. The extended MVDs standardize representation of modular buildings’ structural models in the IFC language. It combines both structural design and analysis representations of the building and is developed in such a way that could be used in both modular and site-built projects.
11.1.3. A Tool and Approach for Interpretation of Structural Analysis Model from Building Information Model

A methodology for interpretation of structural models directly from building information models in coordination view is presented in this dissertation. The methodology is used for development of a tool in this study for automating model transformation process. This mechanism and supporting tool can input the model in Coordination View V.2 and transform it to a structurally equivalent structural mode. Incorporating Industry Foundation Classes (IFC) as the open standard file format for BIM information exchange for both importing and exporting information makes the interpretation mechanism and supporting tool highly interoperable. The methodology for the Interpreted Information Exchange (IIE) documentation has been used for design of the presented mechanism. In this methodology, the whole interpretation process in an IIE would be divided into several exchange units (eUs), each responsible for exchanging one or a set of related information units. The modular nature of the methodology makes the units of an IIE standard reusable in others if applicable. Use the proposed methodology makes it possible to achieve the capability of recognizing, extracting and exporting the required information from the building information model to the corresponding engineering D/A model.

Three main model transformations are identified and required for generation of basic structural analytical models from building information model: representation of
model transformation, material information enrichment, and connectivity adjustment. The first one is related to changing the shape of element representation to their equivalent topological representation. Enriching the model with additional mechanical properties of the structural material according to their name in the input model is the second transformation. The third one is related to merging end/corner points of elements at their connection points with other structural elements. The designed interpretation mechanism is made up of one direct-exchange and four interpretation units. The first interpretation unit is responsible for converting swept solid representation of linear elements to their topological representation; the second interpretation unit does a similar transformation for planar elements; the third interpretation unit enriches the output model by adding the materials’ mechanical properties required for structural design of elements; and the forth unit integrates the transformed elements by adjusting their coordinates. The designed mechanism along with supporting tool developed for automating model transformation process is an example of implementation of the IIE concept.

11.1.4. A Mechanism for Interpretation of Structural Analysis Models from Building Information Models

Using the proposed IIE concepts and based on the MVD and IDM developed for multi-story modular building, a mechanism has been developed for transformation of BIM models of multi-story modular buildings into their equivalent structural analytical models. In this process, generation of the structural D/A model of a modular building
includes three stages: Preparation, Interpretation, and Integration. In the Preparation stage, separate IFC BIM models of the building along with structural model of the site-built parts are prepared. The interpretation process presented in this chapter could be categorized into three sequential steps: Transformation, Refinement, and Assembling. A series of interpretation Units (iUs) are defined for each of these steps to document transformation required for execution of these steps. After implementations of the iUs, structural elements related to site-built components would be added to the structural model of the assembled modules in Integration stage. By completion of this stage, structural model of the building would be ready for analysis and design in the last stage.

11.2. Research Contributions

This research has resulted in the following two main contributions: 1) development of a new information exchange mechanism to facilitate engineering analysis BIM uses; and 2) development of a new methodology for standardization of BIM information exchanges. These outcomes are further discussed in the following subsections.
11.2.1. Proposing a New Information Exchange Mechanism for Facilitate Engineering Analysis BIM Uses

The IIE concept is presented in this research as a solution to the lack of standardized discipline-specific MVDs that need to be created by software developers in the industry. In the current state of implemented information exchange standards in the construction industry, most of the available commonly implemented MVDs are related to model coordination, while several discipline-specific MVDs are already available that are not implemented yet. Without implementation of discipline-specific MVDs, information exchanges from BIM authoring tools to engineering D/A tools are very inefficient and error-prone. The IIE makes it possible to use the standards that are already widely implemented by major AEC software developers for interpretation of required information for different discipline-specific uses, as oppose to requiring all BIM authoring tools to implement several domain-specific standards, which would be very expensive to initially implement, validate, and maintain. In other words, implementation of the IIE concept makes it possible to create IFC file in several engineering domain-specific standardized MVDs, which are not yet implemented by BIM authoring tools, from the few currently widely implemented MVDs such as CV V2.0. Such approach could shift the responsibility of exporting engineering domain-specific MVDs from the developers of BIM authoring software packages to the developers of engineering D/A analysis tools, who are actually the ones that need and benefit from implementation of domain-specific MVDs. It also facilitates participation of third party software developers to use the export
model of BIM authoring software and connect them with domain-specific applications by
development of stand-alone, server-based, or add-on tools.

11.2.2. Proposing a New Methodology for Standardization of BIM
Information Exchanges

This research has resulted in a proposed modified version of the buildingSMART
International methodology for standardization of information exchanges, within which
the current methodology is extended by adding a new component named PAM. In the
current methodology, every information exchange standard is made up of two main
components including an IDM and a MVD such that there is no shared component
between different pairs of standards, and as a result they are not integrated. In the
extended methodology presented in this research, a PAM is used as a basis for defining
the IDM. Since MVDs are defined according to the IDM information requirement, the
PAM would be the basis for all components of the standards that are developed using the
proposed methodology. Using the PAM as an identical basis for development of different
information exchange standards related to different BIM uses would result in an
integrated set of standards serving the building industry. The difference between the
current and the proposed information exchange standardization methodology is shown in
Figure 11-1. In addition, the architecture of the PAM allows defining LOD concisely in a
human readable format by referring to information units of the PAM, while in the
currently available LOD standards, definition of such levels is not concise as they are highly descriptive.

Figure 11-1. Information exchange standards: using the proposed methodology (top) and using the current methodology (down)
11.3. Broader Impacts

The proposed IDM will be a good resource for companies interested to start using this technology. The industry generally knows the benefits of building modularization, but because of lack of in-house knowledge, they are not confident enough to start using this system. The proposed IDM can help modular construction firms to better understand what processes they have to follow and which design criteria they have to consider at each step of the project. Also, the proposed PAM would be a resource for designers; they can use this model to find the right options for different elements of the modules and design buildings more efficiently. The PAM could also be used for any product design and optimization purposes such as product family design or implementation of lean design concept in modular construction projects. These will provide opportunities at a large scale to reduce waste and save the environment through sustainability not achievable through conventional construction.

Besides the benefits that this research project has for new adopters of this technology, it has more benefits for companies that have already adopted this technology and have experience in this field. In the following, some of these benefits are listed:

- The developed IDM, which specify information requirements for different disciplines, will help users to optimize the design process in such a way that less clashes and rework occur and will help information to be delivered to the right discipline at the right time. This model is also sharing of information and
experience of companies in this field, which helps them to take advantage of this knowledge-sharing to improve the performance of their products.

- The developed IDM and MVD will address interoperability issues of modular building industry. The IDM and MVD are developed based on the National BIM Standard – United States® (NBIMS-US™); accordingly, their adoption and implementation by software developers will bring about high interoperability between BIM software as well as between BIM and analysis software in modular building projects. As a result, less effort has to be devoted for multiple modeling of the same building for each software they use. Furthermore, using an identical model for different purposes significantly decreases clashes and makes it possible to have an integrated information model of the building.

- Implementation of the interpretation mechanism developed for structural design of modular buildings in a software like the one developed in this research for building information model-to-structural model transformation will dramatically decrease structural design cost and required time in modular building projects. As a result of using the Industry Foundation Classes (IFC), which is an open file format for building information models, such software can import the model from a large variety of BIM software and export the simplified structural model. The use of such software will help save modeling time, decrease analysis and design time because of the simplification, decrease the coordination time between
structural engineers and other disciplines, and minimize design errors which are caused by modeling and interoperability flaws.

This research study also led to the development of a tool for automated transformation of building information models to analytical structural model. Although this tool was developed as a proof-of-concept for implementation of the Interpreted Information Exchange (IIE) concept, the resulting tool can be very useful for design companies for facilitating difficulties associated with structural engineering use of BIM. This tool is developed as part of a platform that is developed in this research for information extraction from IFC files. In addition to automating the mechanism designed in this research for interpretation of the analytical structural model of modular buildings from their building information model, the developed platform can also be used for automating any other BIM-related information interpretation process; other example areas for such application include energy modeling, construction, manufacturing, and transportation.
11.4. Potential Future Researches

There is potential for conducting extensive follow-up research based on the outcomes of this study. Some of the potential future studies are briefly discussed in the following:

- Extending the information management framework – the framework developed for modular construction to capture more detailed information can be extended for other BIM uses besides the structural modeling/design. For example, the framework could be extended to better serve the MEP design of modular buildings. It could also be extended to capture information exchanges related to fabrication or facility management of modular buildings.

- Developing a product architecture model for site-built constructions – such a PAM can be developed and used for re-representing the current information exchange frameworks. The same PAM that was developed in this research for modular building projects could be developed and used as a basis for standardizing information requirements and Level of Developments.

- Applying the PAM to implement the product family design concept – modular building projects can benefit from employing the PAM in product family design and evaluating such benefits through case studies.

- Implementing the IIE concept in other engineering uses of BIM – This can be the case in areas such as energy modeling and lighting analysis. In such follow-up research, an interpretation mechanism needs to be designed for each of the uses.
There may be some shortcomings in current IFC schema and the MVDs available in the industry. Part of the follow-up research would be identifying these shortcomings and suggesting solutions.

- Identifying additional required improvement in Coordination View – this is deemed necessary in order to make it a M-MVD for engineering analysis BIM uses. In this follow-up research, first the requirements of the most common engineering BIM uses needs to be identified; then, it should be followed by proposing required extensions for Coordination View to buildingSMART International.
APPENDIX A: Task Unit and Exchange Model Table Sets
There are 17 Task Unit (TU) and 13 Exchange Model (EM) tables developed for the process map proposed in Chapter 3.

A.1. Task Unit Table Set

Table A-1: Task Unit table set

<table>
<thead>
<tr>
<th>[TU.A.0] Conceptual Design of Building</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Stage</strong></td>
</tr>
<tr>
<td><strong>Discipline(s)</strong></td>
</tr>
<tr>
<td><strong>Input</strong></td>
</tr>
<tr>
<td><strong>Output</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>[TU.C.1] Construction Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Stage</strong></td>
</tr>
<tr>
<td><strong>Discipline(s)</strong></td>
</tr>
<tr>
<td><strong>Input</strong></td>
</tr>
<tr>
<td><strong>Output</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
</tbody>
</table>
general properties of the modules such as size and weight. In addition, storage of the modules in the construction site and their installation are reviewed and limitations are recognized in this task.

[TU.M.1] Manufacturing Evaluation

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Criteria Definition Phase (31-30 00 00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Manufacturing Services (33-41 09 21)</td>
</tr>
<tr>
<td>Input</td>
<td>EM.AC.1 (Preliminary Architectural Concept Model)</td>
</tr>
<tr>
<td></td>
<td>EM.CM.1 (Construction Criteria)</td>
</tr>
<tr>
<td>Output</td>
<td>EM.MA.1 (Manufacturing Criteria)</td>
</tr>
<tr>
<td>Description</td>
<td>Manufacturer reviews the defined project, and by considering the present status of the manufacturing facilities and available resources recognizes the limitations for the prefabricated parts of the buildings. In addition, the transportation of the modules from the manufacturing site to the construction site is reviewed and limitations are recognized in this task.</td>
</tr>
</tbody>
</table>

[TU.A.2] Preliminary Design of Building Architecture

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Preliminary Design Phase (31-20 10 21)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Architecture (33-21 11 00)</td>
</tr>
<tr>
<td>Input</td>
<td>EM.MA.1 (Manufacturing Criteria)</td>
</tr>
<tr>
<td></td>
<td>EM.CA.1 (Construction Criteria)</td>
</tr>
<tr>
<td></td>
<td>EM.MA.2 (MEP Criteria)</td>
</tr>
<tr>
<td></td>
<td>EM.SA.2 (Structural Concept Model)</td>
</tr>
<tr>
<td>Output</td>
<td>EM.AS.2 (Architectural Concept Model)</td>
</tr>
<tr>
<td>Description</td>
<td>The objective of this task is to prepare the architectural model to a certain level that the structural engineer can start evaluating different structural systems. The building is broken down into separate modules in this level. In addition, in this task, the architect has to prepare the preliminary layout of the required core that is going to be used as a corridor, elevator shaft, staircases, etc.</td>
</tr>
</tbody>
</table>

| [TU.S.2] Preliminary Design of Building Structure |
| --- | --- |
| **Project Stage** | Preliminary Design Phase (31-20 10 21) |
| **Discipline(s)** | Structural Engineering (33-21 31 14) |
| **Input** | EM.AS.2 (Architectural Concept Model) |
| **Output** | EM.SA.2 (Structural Concept Model) |
| **Description** | In this task, the structural engineer reviews the architectural conceptual design, estimates design loads, recognizes the most appropriate structural system for the building and modules, and investigates if the building needs a secondary load resisting system. |

| [TU.S.2.1] Define General Structural System of the building |
| --- | --- |
| **Project Stage** | Preliminary Design Phase (31-20 10 21) |
| **Discipline(s)** | Structural Engineering (33-21 31 14) |
| **Parent Task** | TU.S.2 (Preliminary Design of Building Structure) |
| **Description** | In this task, the structural engineer reviews the architectural conceptual design, estimates design loads, and investigates |
different structural systems for the building to decide the existence of the secondary load resisting system for lateral loads, type of the foundation, and requests probable changes in the layout or other general architectural properties of the building.

**[TU.S.2.2] Define Structural System of the modules**

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Preliminary Design Phase (31-20 10 21)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td>Parent Task</td>
<td>TU.S.2 (Preliminary Design of Building Structure)</td>
</tr>
<tr>
<td>Description</td>
<td>Based on the defined general structural system of the building in task TU.S.2.1, the internal structural system of the modules’ floor, ceiling, walls, bracings, etc. is defined in this task.</td>
</tr>
</tbody>
</table>

**[TU.S.2.3] Author Structural Model**

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Preliminary Design Phase (31-20 10 21)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td>Parent Task</td>
<td>TU.S.2 (Preliminary Design of Building Structure)</td>
</tr>
<tr>
<td>Description</td>
<td>The first structural building information model would be developed based on the Architectural Concept Model and the defined structural systems in tasks TU.S.2.1 and TU.S.2.1. This model includes some information about the general structural elements of the modules and site-built structures.</td>
</tr>
</tbody>
</table>
### [TU.M.2] MEP Preliminary Design

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Preliminary Design Phase (31-20 10 21)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Electrical Engineering (33-21 31 21)</td>
</tr>
<tr>
<td></td>
<td>Plumbing Engineering (33-21 31 17 11)</td>
</tr>
<tr>
<td></td>
<td>Mechanical Engineering (33-21 31 17)</td>
</tr>
<tr>
<td>Input</td>
<td>EM.AE.2 (Architectural Concept Model for MEP Engineers)</td>
</tr>
<tr>
<td>Output</td>
<td>EM.EA.2 (MEP Concept Model)</td>
</tr>
<tr>
<td>Description</td>
<td>The primary goal of this task is the preliminary MEP design and to recognize the location, path, and required space of the equipment.</td>
</tr>
</tbody>
</table>

### [TU.A.3] Architectural Design Development

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Design Development Phase (31-20 20 00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Architecture (33-21 11 00)</td>
</tr>
<tr>
<td>Input</td>
<td>EM.SA.3 (Structural Design Development)</td>
</tr>
<tr>
<td></td>
<td>EM.EA.3 (MEP Model)</td>
</tr>
<tr>
<td>Output</td>
<td>EM.AS.3 (Architectural Design Development)</td>
</tr>
<tr>
<td></td>
<td>EM.AE.3 (Architectural Model for MEP Engineers)</td>
</tr>
<tr>
<td>Description</td>
<td>The objective of this task is to develop the architectural model to a certain level that the structural engineer can start modeling and design of the building’s structural members. In this task unit, the architect may revise the design based on the feedbacks from structural designer.</td>
</tr>
</tbody>
</table>

### [TU.S.3] Structural Design Development

| Project Stage       | Design Development Phase (31-20 20 00) |
### Discipline(s)
| Discipline(s)          | Structural Engineering (33-21 31 14) |

### Input
| Input                  | EM.AS.3 (Architectural Design Development) |

### Output
| Output                 | EM.SA.3 (Structural Design Development) |

### Description
In this task, the structural engineer performs detail analysis of the structure and completes the structural design of the building to the detail design level and production of structural manufacturing and construction drawings. In this task unit, the structural engineer may revise the design based on the feedback from architecture.

---

### [TU.S.3.1] Export information from Structural Model to Structural D/A Software

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Design Development Phase (31-20 20 00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td>Parent Task</td>
<td>TU.S.3 (Preliminary Design of Building Structure)</td>
</tr>
<tr>
<td>Description</td>
<td>In this task, after some preparation, the Structural Model is used for generating the Structural D/A Model. After importing the model, it would be modified if required to make it ready for structural design and analysis.</td>
</tr>
</tbody>
</table>

---

### [TU.S.3.2] Perform Structural Analyses and Design

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Design Development Phase (31-20 20 00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td>Parent Task</td>
<td>TU.S.3 (Preliminary Design of Building Structure)</td>
</tr>
<tr>
<td>Description</td>
<td>Design of the general structural members of the building is the main goal of the building. At the end of this task, general information about the main structural members of the building,</td>
</tr>
</tbody>
</table>
including sizes of the beams, columns, bracings, foundations, shear walls and required reinforcement at different part of the structure will be specified.

[TU.S.3.3] Detail Design of the Structure

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Design Development Phase (31-20 20 00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td>Parent Task</td>
<td>TU.S.3 (Preliminary Design of Building Structure)</td>
</tr>
<tr>
<td>Description</td>
<td>The details of the structural members such as the connection details, steel member splices, and concrete reinforcement patterns are designed and specified in this task.</td>
</tr>
</tbody>
</table>

[TU.S.3.4] Prepare/Update Structural Model

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Design Development Phase (31-20 20 00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td>Parent Task</td>
<td>TU.S.3 (Preliminary Design of Building Structure)</td>
</tr>
<tr>
<td>Description</td>
<td>Using the structural design data resulting from tasks TU.S.3.2 and TU.S.3.3, in this task, the Structural Concept Model is modified and updated to prepare the Structural Model.</td>
</tr>
</tbody>
</table>

[TU.M.3] MEP Design Development

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Design Development Phase (31-20 20 00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Electrical Engineering (33-21 31 21 )</td>
</tr>
<tr>
<td></td>
<td>Plumbing Engineering (33-21 31 17 11)</td>
</tr>
<tr>
<td></td>
<td>Mechanical Engineering (33-21 31 17)</td>
</tr>
</tbody>
</table>
### EM.AE.3 (Architectural Model for MEP Engineers)

**Input**  
EM.AE.3 (Architectural Model for MEP Engineers)

**Output**  
EM.EA.3 (MEP Model)

**Description**  
The main objectives of this task are design and modeling of MEP equipment of the building.

### [TU.M.4] Manufacturing Documents Preparation

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Fabrication Drawing Preparation Phase (31-25 10 21)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Manufacturing Services (33-41 09 21)</td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td>EM.SM.4 (Manufacturing Document Update)</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Document preparation for manufacturing of prefabricated parts of the building along with their interfaces with site-built structures is the main objective of this task.</td>
</tr>
</tbody>
</table>

### [TU.C.4] Construction Documents Preparation

<table>
<thead>
<tr>
<th>Project Stage</th>
<th>Construction Documents Preparation Phase (31-25 10 00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline(s)</td>
<td>Construction Management (33-25 16 00)</td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td>EM.SC.4 (Construction Document Update)</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Document preparation for construction of site-built parts of the building along with their interfaces with prefabricated structures is the main objective of this task.</td>
</tr>
</tbody>
</table>


A.2. Exchange Model Table Set

Table A-2: Exchange Model table set

[EM.AC/M.1] Preliminary Architectural Concept Model

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Criteria Definition Phase (31-30 00 00)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Architecture (33-21 11 00)</td>
</tr>
<tr>
<td></td>
<td>Construction Management (33-25 16 00)</td>
</tr>
<tr>
<td></td>
<td>Manufacturing Services (33-41 09 21)</td>
</tr>
<tr>
<td>Description</td>
<td>This exchange model contains the preliminary architectural conceptual design. It includes general information about the project such as site location, building layout, and number of stories that the Construction Manager and Manufacturer need to identify their preferences and limitations in the project.</td>
</tr>
</tbody>
</table>

[EM.CM.1] Construction Criteria

<table>
<thead>
<tr>
<th>Type</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Criteria Definition Phase (31-30 00 00)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Construction Management (33-25 16 00)</td>
</tr>
<tr>
<td></td>
<td>Manufacturing Services (33-41 09 21)</td>
</tr>
<tr>
<td>Description</td>
<td>This exchange model contains information about the limitations and preferences of the construction group. These criteria are related to the limitations of construction site, available equipment, and skilled workers. Satisfying these criteria will reduce the cost and time, and will increase the</td>
</tr>
</tbody>
</table>
quality of construction.

[EM.MA.1] Manufacturing Criteria

<table>
<thead>
<tr>
<th>Type</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Criteria Definition Phase (31-30 00 00)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Manufacturing Services (33-41 09 21) Architecture (33-21 11 00)</td>
</tr>
<tr>
<td>Description</td>
<td>This exchange model contains information about the limitations and preferences of the manufacturing group. These criteria can be related to the dimension, weight, and material selection. Satisfying these criteria will reduce the cost and time, and will increase the quality of manufacturing and transportation.</td>
</tr>
</tbody>
</table>

[EM.AS.2] Architectural Concept Model

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Preliminary Design Phase (31-20 10 21)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Architecture (33-21 11 00) Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td>Description</td>
<td>The objective of this exchange model is to provide the structural engineer with conceptual design of the building, desired breakdown of the building into modules, and preliminary layout of the core and other site-built structures for structural conceptual design. This exchange model can be sent iteratively to come up with an agreement between actors.</td>
</tr>
</tbody>
</table>

[EM.SA.2] Structural Concept Model
<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Preliminary Design Phase (31-20 10 21)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td></td>
<td>Architecture (33-21 11 00)</td>
</tr>
<tr>
<td>Description</td>
<td>Reviewing the architectural conceptual design and sending feedback to the architect are the main objectives of this information exchange. Also, this exchange model contains information such as the structural limitations of the received architectural model, suggested structural system, possible required secondary structural system, suggestion for modification of the core and other site-built structures, and estimated location and size of the structural elements. This exchange model can be sent iteratively to come up with an agreement between actors.</td>
</tr>
</tbody>
</table>

[EM.AE.2] Architectural Concept Model for MEP Engineers

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Preliminary Design Phase (31-20 10 21)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Architecture (33-21 11 00)</td>
</tr>
<tr>
<td></td>
<td>Electrical Engineering (33-21 31 21)</td>
</tr>
<tr>
<td></td>
<td>Plumbing Engineering (33-21 31 17 11)</td>
</tr>
<tr>
<td></td>
<td>Mechanical Engineering (33-21 31 17)</td>
</tr>
<tr>
<td>Description</td>
<td>The objective of this exchange model is to send architectural conceptual design, building’s layout and breakdown, spaces and occupancies information, and the building core layout to the MEP engineers.</td>
</tr>
</tbody>
</table>
### [EM.EA.2] MEP Concept Model

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Preliminary Design Phase (31-20 10 21)</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>This exchange model contains MEP engineers’ requirements, including the location, path, and a rough estimation of the required size of the ducts and spaces, along with the preferred material for the building components that may affect their design.</td>
<td></td>
</tr>
</tbody>
</table>

### [EM.AS.3] Architectural Design Development

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Design Development Phase (31-20 20 00)</td>
<td></td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Architecture (33-21 11 00)</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td>Description</td>
<td>In this exchange model, the architect passes the developed architectural design of the building to the structural engineer for design development of the building’s structure. In this module, the architect has considered the suggestions and limitations of the structural engineer. This exchange model can be sent iteratively if there are still structural conflicts and feedbacks on the architectural model. Sending the revised architectural model is continued until no structural conflicts remain. In this exchange model, the architect has to inform the</td>
<td></td>
</tr>
</tbody>
</table>
structural engineer about the related limitations of the other disciplines.

### [EM.SS.3] Structural D/A Model

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Design Development Phase (31-20 20 00)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td>Description</td>
<td>In this information exchange, the structural engineer generates</td>
</tr>
<tr>
<td></td>
<td>structural D/A model by exporting information from the structural</td>
</tr>
<tr>
<td></td>
<td>model. Depending on the model and the structure, the structural</td>
</tr>
<tr>
<td></td>
<td>engineer may have to prepare model for being exported and modify</td>
</tr>
<tr>
<td></td>
<td>imported information in the structural D/software to add/modify</td>
</tr>
<tr>
<td></td>
<td>some information to make it ready for analysis and design.</td>
</tr>
</tbody>
</table>

### [EM.SA.3] Structural Design Development

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Design Development Phase (31-20 20 00)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td></td>
<td>Architecture (33-21 11 00)</td>
</tr>
<tr>
<td>Description</td>
<td>In this information exchange, the structural engineer passes</td>
</tr>
<tr>
<td></td>
<td>the designed structural model of building to the architect for</td>
</tr>
<tr>
<td></td>
<td>review. This exchange model can be sent iteratively if there are</td>
</tr>
<tr>
<td></td>
<td>still conflicts between the revised structural design and</td>
</tr>
<tr>
<td></td>
<td>architectural model of the building. Sending the revised</td>
</tr>
<tr>
<td></td>
<td>architectural model is continued until no conflicts and</td>
</tr>
</tbody>
</table>
feedback remains.

**[EM.AE.3] Architectural Design Development for MEP Engineers**

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Design Development Phase (31-20 20 00)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Architecture (33-21 11 00)</td>
</tr>
<tr>
<td></td>
<td>Electrical Engineering (33-21 31 21 )</td>
</tr>
<tr>
<td></td>
<td>Plumbing Engineering (33-21 31 17 11)</td>
</tr>
<tr>
<td></td>
<td>Mechanical Engineering (33-21 31 17)</td>
</tr>
<tr>
<td>Description</td>
<td>In this exchange model, the architect passes the designed architectural model of the building to MEP engineers for their design development activities. In this model, the architect has considered the suggestions and limitations of the MEP engineer. This architectural model includes occupancies, exact size and location of the available spaces for MEP equipment, size and material properties of the components that are important for MEP design of the building, etc.</td>
</tr>
</tbody>
</table>

**[EM.EA.3] MEP Model**

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Design Development Phase (31-20 20 00)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Electrical Engineering (33-21 31 21 )</td>
</tr>
<tr>
<td></td>
<td>Plumbing Engineering (33-21 31 17 11)</td>
</tr>
<tr>
<td></td>
<td>Mechanical Engineering (33-21 31 17)</td>
</tr>
<tr>
<td></td>
<td>Architecture (33-21 11 00)</td>
</tr>
</tbody>
</table>
| Description | In this exchange model, the MEP engineers send the MEP
model of the building to the architect to inform him/her about the exact location, path and size of the required spaces and ducts along with the specifications of the insulation, windows, and other related component and materials.


<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Detailed Design Phase (31-20 20 11)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td></td>
<td>Manufacturing Services (33-41 09 21)</td>
</tr>
<tr>
<td>Description</td>
<td>This module contains fabrication model that the manufacturer needs for fabrication of the building’s structure, including geometry, acceptable tolerances, material information, connection details, designed concrete mix information, etc.</td>
</tr>
</tbody>
</table>

[EM.SC.4] Construction Document Update

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Stage</td>
<td>Detailed Design Phase (31-20 20 11)</td>
</tr>
<tr>
<td>Exchange Disciplines</td>
<td>Structural Engineering (33-21 31 14)</td>
</tr>
<tr>
<td></td>
<td>Construction Management (33-25 16 00)</td>
</tr>
<tr>
<td>Description</td>
<td>This exchange model passes the structural model of the site-built part of the building, like foundations and shear walls, if any, to the construction manager. In addition, it contains the required information for installation of the prefabricated arts including connection details of the modules together, module’s hoisting method, connection of the module to the</td>
</tr>
</tbody>
</table>
site-built parts, and acceptable tolerances.
APPENDIX B: Exchange Requirements Specification
## Exchange Modules Specification

<table>
<thead>
<tr>
<th>Modular Buildings</th>
<th>EMAC/M.1</th>
<th>EMLC/M.1</th>
<th>EMMA.1</th>
<th>EMAS.2</th>
<th>EMAS.3</th>
<th>EMAE.2</th>
<th>EMLE.2</th>
<th>EMAS.3</th>
<th>EMLA.3</th>
<th>EMLA.3</th>
<th>ELE.M.4</th>
<th>EMLC.4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No. of Stories</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Stories Height</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Layout</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Orientation</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Earthquake/Wind Design Loads</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Modular Assemblies</strong></td>
<td>G</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Volumetric Modules</strong></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Weight Limit</strong></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dimensions</strong></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Volumetric Limitations</strong></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Dimensions</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Side Gaps</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Floor</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Occupancy</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Live Load</strong></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Perimeter Beam</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Connections</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Rigidity</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Bolted Connection</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Welded Connection</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cross Section</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Allowable Tolerances</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Middle Beam</strong></td>
<td>G</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Connections</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Rigidity</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Bolted Connection</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Welded Connection</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cross Section</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Allowable Tolerances</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Joists</strong></td>
<td>G</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Connections</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Rigidity</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>EMLACM.1</td>
<td>EMLCM.1</td>
<td>EMMA.1</td>
<td>EMAS.2</td>
<td>EMSA.2</td>
<td>EMLAE.2</td>
<td>EMLEA.2</td>
<td>EMLAS.3</td>
<td>EMSS.3</td>
<td>EMLSA.3</td>
<td>EMLAE.3</td>
<td>EMLSM.4</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------</td>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>Bolted Connection</td>
<td>G</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Welded Connection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross Section</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spacing</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allowable Tolerances</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slab</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thickness</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concrete</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strength</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reinforcement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cover</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grade</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diameter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steel Deck</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross Section</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection Info.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allowable Tolerances</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sheathing (OSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thickness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insulation</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R-Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thickness</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Walls</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Load Capacity</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stiffness</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Light Gauge Steel</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resistance</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stiffness</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross Sections</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection Info.</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allowable Tolerances</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wall Board</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Exchange Modules Specification

<table>
<thead>
<tr>
<th></th>
<th>EM.LAC/M.1</th>
<th>EM.LCM.1</th>
<th>EM.MA.1</th>
<th>EM.LAS.2</th>
<th>EM.LSA.2</th>
<th>EM.MAE.2</th>
<th>EM.LEA.2</th>
<th>EM.LAS.3</th>
<th>EM.LSA.3</th>
<th>EM.MAE.3</th>
<th>EM.LEA.3</th>
<th>EM.SM.4</th>
<th>EM.LSC.4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resistance</strong></td>
<td>G</td>
<td></td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Stiffness</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td></td>
<td>G</td>
<td></td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Thickness</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Connection Info.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Allowable Tolerances</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Structural Panel</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>R-Value</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Perm</strong></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Resistance</strong></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Stiffness</strong></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Layers</strong></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Perimeter Connection</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Connection Info.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Allowable Tolerances</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Insulation</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>R-Value</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
</tr>
<tr>
<td><strong>Perm</strong></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Thickness</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Steel Studs</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Resistance</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Stiffness</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Connections</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Rigidity</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bolted Connection</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Welded Connection</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cross Section</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Spacing</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Allowable Tolerances</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Non-Structural Studs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Spacing</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dimension</strong></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Braces</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Connections</strong></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module Type</td>
<td>Rigidity</td>
<td>Bolted Connection</td>
<td>Welded Connection</td>
<td>Cross Section</td>
<td>Allowable Tolerances</td>
<td>Ceiling</td>
<td>Perimeter Beam</td>
<td>Material</td>
<td>Connections</td>
<td>Rigidity</td>
<td>Bolted Connection</td>
<td>Welded Connection</td>
<td>Cross Section</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>---------------</td>
<td>---------------------</td>
<td>---------</td>
<td>----------------</td>
<td>----------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Perimeter Beam</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Material</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Connections</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Joists</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Material</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Connections</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Spacing</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Insulation</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Thickness</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>G</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Type</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>Insulation</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Thickness</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>G</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
### Exchange Modules Specification

<table>
<thead>
<tr>
<th></th>
<th>EM.LA.C/M.1</th>
<th>EM.LC.M.1</th>
<th>EM.MA.1</th>
<th>EM.LS.A.2</th>
<th>EM.LA.E.2</th>
<th>EM.LA.E.3</th>
<th>EM.LS.A.3</th>
<th>EM.LA.E.3</th>
<th>EM.LM.S.4</th>
<th>EM.LS.C.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corner Posts</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connections</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Rigidity</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bolted Connection</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Welded Connection</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross Section</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allowable Tolerances</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beam Offset at Top</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Beam Offset at Bottom</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Middle Posts</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connections</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rigidity</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bolted Connection</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Welded Connection</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross Section</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Allowable Tolerances</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beam Offset at Top</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Beam Offset at Bottom</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Voids</td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Doors</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>R-Value</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Height</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>R-Value</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Height</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Planar Modules</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Weight Limit</td>
<td>G</td>
<td>M</td>
<td>T</td>
<td></td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dimension Limits</td>
<td>G</td>
<td>M</td>
<td>T</td>
<td></td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manufacturing Limits</td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Dimensions</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Hoisting Info</td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facades</td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Module Type</td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Structure</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection to Main Struc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rigidity</td>
<td></td>
<td>G</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bolted Connection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Welded Connection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Site-Built Structure</td>
<td></td>
<td>G</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Foundation</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layout</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Structure</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection to Modules</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rigidity</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bolted Connection</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Welded Connection</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural Core</td>
<td></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Layout</td>
<td></td>
<td>G</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Position</td>
<td></td>
<td>G</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>2D Braced Frame</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Main Structure</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection to Modules</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rigidity</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bolted Connection</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Welded Connection</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3D Braced Frame</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Main Structure</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection to Modules</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rigidity</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bolted Connection</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Welded Connection</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shear Wall</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Main Structure</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection to Modules</td>
<td></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rigidity</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bolted Connection</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Welded Connection</td>
<td></td>
<td>G</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Podium</td>
<td></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Layout</td>
<td></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Position</td>
<td></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Number of Stories</td>
<td></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>
## Exchange Modules Specification

<table>
<thead>
<tr>
<th></th>
<th>EM.ACM.1</th>
<th>EM.CM.1</th>
<th>EM.MA.1</th>
<th>EM.AS.2</th>
<th>EM.SA.2</th>
<th>EM.SA.2</th>
<th>EM.AE.2</th>
<th>EM.AE.2</th>
<th>EM.AE.2</th>
<th>EM.SS.3</th>
<th>EM.SA.3</th>
<th>EM.AE.3</th>
<th>EM.AE.3</th>
<th>EM.SM.4</th>
<th>EM.SC.4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stories Height</strong></td>
<td>G</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Structural System</strong></td>
<td>G</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>M</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Main Structure</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Connection to Modules</strong></td>
<td>G</td>
<td>T</td>
<td>M</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td><strong>Rigidity</strong></td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
</tr>
<tr>
<td><strong>Bolted Connection</strong></td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
</tr>
<tr>
<td><strong>Welded Connection</strong></td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
</tr>
</tbody>
</table>
APPENDIX C: Model View Diagrams and Concept Definitions
**Generic AEC/FM Concept Description**

**Project Actor Assignment**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-001</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

```
<Variable Concept>  PSU-001
                   
                   Project Actor Assignment
```

**Definition**
Information such as name, address, and role related to a project actor.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description

### Project Manufacturer

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-002</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Project Actor Assignment”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Usage in view definition diagram

[Diagram showing relationships]

### Definition

Information about the manufacturer of the prefabricated parts like name, address, and role.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
### Generic AEC/FM Concept Description

**Project Contractor**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-004</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Project Actor Assignment”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Usage in view definition diagram**

```plaintext
<Variable Concept> ── PSU-001 Project Actor Assignment ── PSU-004 Project Contractor
```

**Definition**

Information about the project contractor like name, address, and role.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved.
## Generic AEC/FM Concept Description

### Project Structural Engineer

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-006</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**
Extends adapter concept “Project Actor Assignment”

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

```
<Variable Concept> → PSU-006:Project Actor Assignment → PSU-006:Project Structural Engineer
```

**Definition**
Information about the structural engineer of the project like name, address, and role.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description

### Building Part

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-008</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**
Extends adapter concept "Contained in Building"

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

![Diagram](image)

**Definition**
Categorizing building elements to different parts based on the construction method.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description
Contained in Building Part

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-010</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**References**

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

<Variable Concept> ➔ PSU-010 Contained in Building Part

**Definition**

Objects that are placed or are contained in building part.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description
Spatial Zone Aggregation

Reference | PSU-011 | Version | 1.0 | Status | Draft
---|---|---|---|---|
Relationships | Extends adapter concept "Contained in Building Part"
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

```
<Variable Concept> | PSU-0410<br>Contained in Building Part | PSU-0411<br>Spatial Zone Aggregation
```

Definition

Information about the spatial structure elements aggregated in the spatial zone.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description
### Type Assignment

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-013</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Usage in view definition diagram

```
<Variable Concept>  PSU-013  Type Assignment
```

#### Definition
Assigning type of a building element from a predefined type list.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved.
**Generic AEC/FM Concept Description**

**Object Common Properties**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-015</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**

**History**  
Created 01.01.2015

**Authors**  
Issa Ramaji

**Document Owner**  
Penn State University

**Usage in view definition diagram**

![Diagram showing the relationship between a variable concept and object common properties]

**Definition**

A collection of describing properties that are common in different objects.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
### Generic AEC/FM Concept Description

#### Module Common Properties

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-016</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

#### Relationships

<table>
<thead>
<tr>
<th>History</th>
<th>Created 01.01.2015</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Authors</th>
<th>Issa Ramaji</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Document Owner</th>
<th>Penn State University</th>
</tr>
</thead>
</table>

#### Usage in view definition diagram

![Diagram](image)

#### Definition

A collection of common properties of prefabricated modules.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description

### Module Tag

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-018</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Module Common Properties”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Usage in view definition diagram

![Diagram](image)

### Definition

Defines the element-id, which is not identical to the GUID. It is a module mark that would be used to refer to a specific module in all methods of human communications.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
## Generic AEC/FM Concept Description

### Module Type

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-020</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**
Extends adapter concept "Module Common Properties"

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

![Diagram](image)

**Definition**
Assigning a type information to modules that are sharing similar properties.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-022</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Module Common Properties”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Usage in view definition diagram**

```plaintext
<Variable Concept>  PSU-016  Module Common Properties  PSU-022  Module Dimensions
```

**Definition**

List of the external dimensions of the Module in all three dimensions of the cubic module. The exterior dimension is the distance between exterior sides of the connections in each direction.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
Generic AEC/FM Concept Description

Perimeter Gaps

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-025</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept &quot;Module Common Properties&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Usage in view definition diagram

```
<Variable Concept> PSU-016 Module Common Properties PSU-025 Perimeter Gaps
```

Definition

List of perimeter gaps on all six faces of the module. The perimeter gap is measured from the exterior face of the module, excluding the connection, to the exterior side of the connection.
Generic AEC/FM Concept Description
Module Common Limitations

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-027</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

Relationships

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

![Diagram](image)

Definition

Information about limitations in the manufacturing, transportation, and installation of the prefabricated module such as weight and dimension limitations.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description

Mass Limit

Reference | PSU-029 | Version | 1.0 | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept “Module Common Limitations”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Definition

Maximum allowable mass of the prefabricated module that could be dictated by the manufacturer, transporter, and/or contractor.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects – By The BLDS Consortium – All Other Rights Reserved
# Generic AEC/FM Concept Description

## Dimensional Limitations

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-031</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Module Common Limitations”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Usage in view definition diagram

```
<Variable Concept> | PSU-027 | Module Common Limitations | PSU-031 | Dimensional Limitations
```

### Definition

Maximum allowable dimensions of the prefabricated module that could be dictated by the manufacturer, transporter, and/or contractor.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
Generic AEC/FM Concept Description
Transportation Information

Reference | PSU-033 | Version | 1.0 | Status | Draft
---|---|---|---|---|---

Relationships

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

![Diagram]

Definition
Information relating to transportation of the prefabricated module to the construction site.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description

### Transportation Route

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-035</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relationships</strong></td>
<td>Extends adapter concept “Transportation Information”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>History</strong></td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Authors</strong></td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Document Owner</strong></td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Usage in view definition diagram

![Diagram](image)

#### Definition

Information about the route that prefabricated module is transported through to the construction site.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
Generic AEC/FM Concept Description

Transportation Time

Reference | PSU-037 | Version | 1.0 | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept "Transportation Information"
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

```
<VARIABLE CONCEPT> -> PSU-033 Transportation Information -> PSU-037 Transportation Time
```

Definition

Time required to deliver the module from the manufacturing plant to the construction site.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description
Module Actor Assignment

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-039</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

```
<Variable Concept> ~ PSU-039
   Module Actor Assignment
```

**Definition**
Information about the module’s actor like manufacturer and transporter.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description
### Module Manufacturer

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-040</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Module Actor Assignment”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Usage in view definition diagram**

- `<Variable Concept>`
- PSU-039: Module Actor Assignment
- PSU-040: Module Manufacturer

**Definition**

Information about the module’s manufacturer like name and address.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
Generic AEC/FM Concept Description
Module Transporter

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-042</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

Relationships
Extends adapter concept "Module Actor Assignment"

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Definition
Information of the actor that delivers prefabricated modules to the construction.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
### Generic AEC/FM Concept Description

#### Module Port Assignment

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-044</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**

**History**

Created 01.01.2015

**Authors**

Issa Ramaji

**Document Owner**

Penn State University

**Usage in view definition diagram**

![Diagram](image)

**Definition**

Information about the ports such as connection and hoisting points within the module that may or may not be used to connect the module with other objects.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description
Hoisting Point Assignment

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-046</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Module Port Assignment”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Usage in view definition diagram

Definition

Information related to the points considered for hoisting the module for transportation and installation purposes.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
Generic AEC/FM Concept Description
Connection Point Assignment

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-048</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept &quot;Module Port Assignment&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Usage in view definition diagram

```
<Variable Concept>  PSU-044  Module Port Assignment  PSU-048  Connection Point Assignment
```

Definition

Information of the points that are considered for connecting the module to other modules or building elements.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description

### Module Representation

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-050</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

### Relationships

<table>
<thead>
<tr>
<th>History</th>
<th>Created 01.01.2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
</tr>
</tbody>
</table>

### Usage in view definition diagram

![Diagram](image)

### Definition

A collection of geometry representations for the module.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description
Aggregated Shape Representation

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-051</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Module Representation”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Usage in view definition diagram

![Diagram showing relationships]

Definition

Representation of the components and subcomponents of the module.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
Module Placement

Placement of the module in the building.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description
Module Part Aggregation

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-053</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

![Diagram of Module Part Aggregation](image)

**Definition**
Assigning the module’s major parts (subassemblies) such as floor, walls, and ceiling to their corresponding module.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved.
Generic AEC/FM Concept Description
Module Wall

Reference | PSU-055 | Version | 1.0 | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept “Module Part Aggregation”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

| <Variable Concept> | PSU-063 | Module Part Aggregation | PSU-065 | Module Wall |
---|---|---|---|---|

Definition
Assigning a wall assembly to the module that it belongs to.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved.
# Generic AEC/FM Concept Description

## Module Floor

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-057</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

### Relationships
Extends adapter concept “Module Part Aggregation”

### History
Created 01.01.2015

### Authors
Issa Ramaji

### Document Owner
Penn State University

#### Usage in view definition diagram

![Diagram]

#### Definition
Assigning a floor assembly to the module that it belongs to.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
### Generic AEC/FM Concept Description

**Module Ceiling**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-059</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Module Part Aggregation”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Usage in view definition diagram

![Diagram](image)

#### Definition

Assigning a ceiling assembly to the module that it belongs to.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved.
Generic AEC/FM Concept Description
Module Column

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-061</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Module Part Aggregation”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Usage in view definition diagram

<Variable Concept> → PSU-063 Module Part Aggregation → PSU-061 Module Column

Definition
Assigning a column assembly to the module that it belongs to.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
# Generic AEC/FM Concept Description

## Generic Module Part

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-063</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**

Extends adapter concept “Module Component Assignment”

**History**

Created 01.01.2015

**Authors**

Issa Ramaji

**Document Owner**

Penn State University

**Usage in view definition diagram**

![Diagram](image)

**Definition**

Assigning module elements that cannot be categorized in predefined module part types such as floor, wall, ceiling, etc., to a generic module part.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
# Generic AEC/FM Concept Description

## Module Utilization Type

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-065</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Module Common Properties”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Usage in view definition diagram

```
<Variable Concept>  PSU-016: Module Common Properties  PSU-065: Module Utilization Type
```

### Definition

Information about the building system that the module is going to be utilized in. An example is floor or envelope for planar modules.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description

### Building Element Aggregation

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-068</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**

<table>
<thead>
<tr>
<th>History</th>
<th>Created 01.01.2015</th>
</tr>
</thead>
</table>

**Authors**

<table>
<thead>
<tr>
<th>Issa Ramaji</th>
</tr>
</thead>
</table>

**Document Owner**

<table>
<thead>
<tr>
<th>Penn State University</th>
</tr>
</thead>
</table>

**Usage in view definition diagram**

```
<Variable Concept> --- PSU-068 --- Building Element Aggregation
```

**Definition**

Information about building elements aggregated in the building element as a sub-member.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
# Generic AEC/FM Concept Description

## Module Part Placement

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-071</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

### Relationships

### History

Created 01.01.2015

### Authors

Issa Ramaji

### Document Owner

Penn State University

### Usage in view definition diagram

![Diagram](image)

### Definition

Placement of the module part in the module.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description
### Module Part Representation

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-072</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**

**History** Created 01.01.2015

**Authors** Issa Ramaji

**Document Owner** Penn State University

**Usage in view definition diagram**

```
<Variable Concept>  PSU-072
Module Representation
```

**Definition**

A collection of geometry representations for the module part.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
# Generic AEC/FM Concept Description

## Wall Resultant Structural Properties

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-073</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Usage in view definition diagram

```
<Variable Concept>  PSU-073 Wall Resultant Structural Properties
```

### Definition

A collection of the resultant structural properties of the module’s wall.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
**Generic AEC/FM Concept Description**

**In-Plane Lateral Load Capacity**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-075</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept &quot;Wall Resultant Structural Properties&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Usage in view definition diagram**

```
<Variable Concept> PSU-075 Wall Resultant Structural Properties PSU-075 In-Plane Lateral Load Capacity
```

**Definition**

Information about the resultant in-plane lateral load capacity of the module’s wall.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description
In-Plane Lateral Stiffness

Reference | PSU-077 | Version | 1.0 | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept "Wall Resultant Structural Properties"
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Definition
Information about the resultant in-plane lateral stiffness of the module’s wall.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLUS Consortium – All Other Rights Reserved
### Generic AEC/FM Concept Description

#### Vertical Load Capacity

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-079</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept &quot;Wall Resultant Structural Properties&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Usage in view definition diagram

```
<Variable Concept> ➔ PSU-079 ➔ Wall Resultant Structural Properties ➔ PSU-079 ➔ Vertical Load Capacity
```

### Definition

Information about the resultant vertical load capacity of the module's wall.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## Generic AEC/FM Concept Description

### Vertical Stiffness

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-081</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Wall Resultant Structural Properties”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Usage in view definition diagram

![Diagram](image)

#### Definition

Information about the resultant vertical stiffness of the module’s wall.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description
Planar Element Part Properties

Reference | PSU-083 | Version | 1.0 | Status | Draft
Relationships

History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Definition
A collection of common properties of planar part of an element.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
# Generic AEC/FM Concept Description

## Planar Element Part Dimensions

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-085</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**
Extends adapter concept “Planar Element Part Properties”

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**
![Diagram]

**Definition**
List of the overall dimensions of a planar part of an element in two directions.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
**Generic AEC/FM Concept Description**

**Floor Resultant Structural Properties**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-087</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Usage in view definition diagram**

```
<Variable Concept> PSU-087
Floor Resultant Structural Properties
```

**Definition**

A collection of the resultant structural properties of the module’s Floor.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
**Generic AEC/FM Concept Description**

**Floor Occupancy Type**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-089</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Extends adapter concept “Floor Resultant Structural Properties”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage in view definition diagram</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram](image)

**Definition**

Information about module’s floor occupancy type such as bedroom, living room, office space. This information may be used for calculating design loads, deflection limits, etc.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
<table>
<thead>
<tr>
<th>Generic AEC/FM Concept Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Floor Diaphragm Rigidity</strong></td>
</tr>
</tbody>
</table>

**Reference**  | PSU-091  
**Version**   | 1.0  
**Status**    | Draft  

**Relationships**  | Extends adapter concept “Floor Resultant Structural Properties”  
**History**       | Created 01.01.2015  
**Authors**       | Issa Ramaji  
**Document Owner** | Penn State University  

**Usage in view definition diagram**

```
<Variable Concept> → PSU-087 → Floor Resultant Structural Properties → PSU-081 → Floor Diaphragm Stiffness
```

**Definition**

Information about rigidity of the floor element part as a diaphragm.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
Generic AEC/FM Concept Description
Ceiling Resultant Structural Properties

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-093</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

Relationships

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Definition
A collection of the resultant structural properties of the module’s Ceiling.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
### Generic AEC/FM Concept Description

#### Ceiling Diaphragm Rigidity

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-095</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Status</td>
<td>Draft</td>
</tr>
</tbody>
</table>

**Relationships**

Extends adapter concept "Ceiling Resultant Structural Properties"

**History**

Created 01.01.2015

**Authors**

Issa Ramaji

**Document Owner**

Penn State University

**Usage in view definition diagram**

```
<Variable Concept>  PSU-083  PSU-095
                     Module Ceiling Structural Properties  Ceiling Diaphragm Rigidity
```

**Definition**

Information about rigidity of the ceiling element part as a diaphragm.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
### Generic AEC/FM Concept Description

#### Structural Model Assignment

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-097</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Created 01.01.2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Issa Ramaji</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Owner</td>
<td>Penn State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage in view definition diagram</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Definition**

Equivalent structural model of a complex building element like a module independently from the building and adjacent building elements.

This document uses the official IFC Model View Definition Format version 1.1.0. of buildingSMART

Template Licensed for use in buildingSMART Projects – By The BLUS Consortium – All Other Rights Reserved
**IFC Release Specific Concept Description (IFC4)**

**Project Manufacturer**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-003</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**

Extends adapter concept “Project Actor Assignment”

**History**

Created 01.01.2015

**Authors**

Issa Ramaji

**Document Owner**

Penn State University

**Usage in view definition diagram**

![Diagram showing relationships between concepts]

**Instantiation diagram**

![Diagram showing instantiation of concepts]

**Implementation agreements**

**IfcRelAssignsToActor**

- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: Must be IfcLabel = “Project Manufacturer”
- **Description**: <Open>
- **RelatedObjects**: Must be IfcProject.
- **RelatedObjectsType**: <Open>
- **RelatingActor**: Must be IfcActor or its subtype.
- **ActingRole**: Must be IfcActorRole or its subtype.

**IfcActor**

- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **ObjectType**: <Open>
- **TheActor**: Must be IfcOrganization.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved.
IFC Release Specific Concept Description (IFC4)
Project Contractor

Reference | PSU-005 | Version | 1.0 | Status | Draft
Relationships | Extends adapter concept “Project Actor Assignment”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IfcRelAssignsToActor
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: Must be IfcLable="Project Contractor".
- **Description**: <Open>
- **RelatedObjects**: Must be IfcProject.
- **RelatedObjectsType**: <Open>
- **RelatingActor**: Must be IfcActor or its subtype.
- **ActingRole**: Must be IfcActorRole or its subtype.

IfcActor
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **ObjectType**: <Open>
- **TheActor**: Must be IfcOrganization.

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium –All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

**Project Structural Engineer**

**Reference**  PSU-007  
**Version**  1.0  
**Status**  Draft

**Relationships**  Extends adapter concept “Project Actor Assignment”

**History**  Created 01.01.2015

**Authors**  Issa Ramaji

**Document Owner**  Penn State University

**Usage in view definition diagram**

**Instantiation diagram**

**Implementation agreements**

---

**IfcRelAssignsToActor**

- **GlobalId**  Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**  Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**  Must be IfcLabel="Project Structural Engineer"
- **Description**  <Open>
- **RelatedObjects**  Must be IfcProject.
- **RelatedObjectsType**  <Open>
- **RelatingActor**  Must be IfcActor or its subtype.
- **ActingRole**  Must be IfcActorRole or its subtype.

**IfcActor**

- **GlobalId**  Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**  Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**  <Open>
- **Description**  <Open>
- **ObjectType**  <Open>
- **TheActor**  Must be IfcOrganization.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

Building Part

Reference | PSU-009 | Version | 1.0 | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept “Generic Aggregation”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

![Diagram](image)

Instantiation diagram

![Diagram](image)

Implementation agreements

**IfcRelContainedInSpatialStructure**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: Must be IfcLabel="Building Part".
- **Description**: <Open>
- **RelatedElements**: Required, IfcBuilding.
- **RelatingStructure**: Required, IfcSpatialZone.

**IfcSpatialZone**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: Required, IfcLabel.
- **Description**: <Open>
- **ObjectType**: <Open>
- **ObjectPlacement**: <Open>
- **Representation**: <Open>
- **LongName**: <Open>
- **PredefinedType**: <Open>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
**IFC Release Specific Concept Description (IFC4)**

**Spatial Zone Aggregation**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-012</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Status</td>
<td>Draft</td>
</tr>
</tbody>
</table>

**Relationships**

Extends adapter concept "Generic Aggregation"

**History**

Created 01.01.2015

**Authors**

Issa Ramaji

**Document Owner**

Penn State University

**Usage in view definition diagram**

![Usage in view definition diagram](image1)

**Instantiation diagram**

![Instantiation diagram](image2)

**Implementation agreements**

<table>
<thead>
<tr>
<th>IfcRelContainedInSpatialStructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
</tr>
<tr>
<td>OwnerHistory</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>RelatedElements</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org) Template Licensed for use in buildingSMART Projects –By The BLIS Consortium –All Other Rights Reserved
**IFC Release Specific Concept Description (IFC4)**

**Building Part Type Assignment**

- **Reference:** PSU-014
- **Version:** 1.0
- **Status:** Draft

**Relationships**
Extends adapter concept “Generic Assignment”

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

**Instantiation diagram**

**Implementation agreements**

---

**This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)**

Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

Module Attributes

**Reference** | PSU-017  
---|---
**Version** | 1.0  
**Status** | Draft

**Relationships**

**History** | Created 01.01.2015  
**Authors** | Issa Ramaji  
**Document Owner** | Penn State University

**Usage in view definition diagram**

![IFC 4 Diagram](image)

**Instantiation diagram**

![Instantiation Diagram](image)

**Implementation agreements**

| IfcBuildingElementProxy |  
| --- | ---  
| **GlobalId** | Providing a GUID is mandatory, but the GUID is allowed to change.  
| **OwnerHistory** | Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.  
| **Name** | <Open>  
| **Description** | <Open>  
| **ObjectType** | Must be IfcLabel=“PrefabricatedModule”.  
| **ObjectPlacement** | Should carry the location of the module. See different placement methods.  
| **Representation** | Should carry the geometric representation of the module.  
| **Tag** | <Open>  
| **PredefinedType** | <Open>
IFC Release Specific Concept Description (IFC4)
Module Tag

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-019</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**
Extends adapter concept “Module Attributes”

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

![Diagram showing relationships between PSU-019, Module Attributes, and Module Tag]

**Instantiation diagram**

```
<IfcBuildingElementProxy>
  <GlobalId>
    Providing a GUID is mandatory, but the GUID is allowed to change.
  </GlobalId>
  <OwnerHistory>
    Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
  </OwnerHistory>
  <Name>
    <Open>
  </Name>
  <Description>
    <Open>
  </Description>
  <ObjectType>
    Must be IfcLabel="PrefabricatedModule".
  </ObjectType>
  <ObjectPlacement>
    Should carry the location of the module. See different placement methods.
  </ObjectPlacement>
  <Representation>
    Should carry the geometric representation of the module.
  </Representation>
  <Tag>
    Required, IfcIdentifier.
  </Tag>
  <PredefinedType>
    <Open>
  </PredefinedType>
</IfcBuildingElementProxy>
```

**Implementation agreements**

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium –All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Module Base Quantities

Reference | PSU-021 | Version | 1.0 | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept “Simple Element Quantities”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
### IFC Release Specific Concept Description (IFC4)

**Module Dimensions**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-023</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Status</td>
<td>Draft</td>
</tr>
</tbody>
</table>

**Relationships**
Extes adapter concept "Module Base Quantities"

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

#### Usage in view definition diagram

#### Instantiation diagram

**Implementation agreements**

**IfcRelDefinesByProperties**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **RelatedObjects**: Required, IfcBuildingElementProxy/(ObjectType="PrefabricatedModule")
- **RelatingPropertyDefinition**: Required, IfcElementQuantity.

**IfcElementQuantity**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: Must be "ModuleBaseQuantities".
- **Description**: <Open>
- **MethodOfMeasurement**: <Open>
- **Quantities**: Required, IfcQuantityLength.

**IfcQuantityLength**
- **Name**: Must be "DimensionInDirection1"
- **Description**: <Open>
- **Unit**: <Open>
- **LengthValue**: Required, IfcLengthMeasure.
<table>
<thead>
<tr>
<th>Formula</th>
<th>&lt;Open&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IfcQuantityLength</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Must be “DimensionInDirection2”</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Unit</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>LengthValue</td>
<td>Required, IfcLengthMeasure.</td>
</tr>
<tr>
<td>Formula</td>
<td>&lt;Open&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Formula</th>
<th>&lt;Open&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IfcQuantityLength</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Must be “DimensionInDirection3”</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Unit</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>LengthValue</td>
<td>Required, IfcLengthMeasure.</td>
</tr>
<tr>
<td>Formula</td>
<td>&lt;Open&gt;</td>
</tr>
</tbody>
</table>
IFC Release Specific Concept Description (IFC4)
3D-Module Specific Quantities

Reference | PSU-024  | Version | 1.0  | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept “Simple Element Quantities”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IfcRelDefinesByProperties
- **Globalld**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **RelatedObjects**: Required, IfcBuildingElementProxy/(ObjectType="PrefabricatedModule"; PredefinedType="Volumetric").
- **RelatingPropertyDefinition**: Required, IfcElementQuantity.

IfcElementQuantity
- **Globalld**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: Must be “3DModuleSpecificQuantities”.
- **Description**: <Open>
- **MethodOfMeasurement**: <Open>
- **Quantities**: <Open>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Perimeter Gaps

**Reference**  PSU-026  **Version**  1.0  **Status**  Draft

**Relationships**  Extends adapter concept “3D-Module Specific Quantities”

**History**  Created 01.01.2015

**Authors**  Issa Ramaji

**Document Owner**  Penn State University

**Usage in view definition diagram**

**Instantiation diagram**

**Implementation agreements**

```
IfcRelDefinesByProperties
GlobalId
OwnerHistory
Description
RelatedObjects

IfcElementProxy
Name
Description
HasAssignments
Har vestedBy
HasContext
InDecomposition
Decomposes
HasAssociations
ObjectType
IsDeclaredBy
IsDefinition

IfcBuildingElementProxy
Representation
Name
Description
HasAssignments
Har vestedBy
HasContext
InDecomposition
Decomposes
HasAssociations
ObjectType
IsDeclaredBy
IsDefinition

IfcRelDefinesByProperties
GlobalId
OwnerHistory
Description
RelatedObjects

IfcPropertySetDefinition
Name
Description
Properties

IfcG400Model
Name
Description
References
```

```
IfcLabel
= "3DModuleSpecificQuantities"

IfcLable
= "GapOnFace1"

IfcLengthMeasure
IfcLable
= "GapOnFace2"

IfcLengthMeasure
IfcLable
= "GapOnFace3"

IfcLengthMeasure
IfcLable
= "GapOnFace4"

IfcLengthMeasure
IfcLable
= "GapOnFace5"

IfcLengthMeasure
IfcLable
= "GapOnFace6"

IfcLabel
= "PrefabricatedModule"

IfcBuildingElementProxyTypeEnum
= "Volumetric"

GlobalId
Providing a GUID is mandatory, but the GUID is allowed to change.

OwnerHistory
Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.

Name
<Open>

Description
<Open>

RelatedObjects
Required, IfcBuildingElementProxy/ObjectType="PrefabricatedModule";
PredefinedType="Volumetric".
```
<table>
<thead>
<tr>
<th>RelatingPropertyDefinition</th>
<th>Required, IfcElementQuantity.</th>
</tr>
</thead>
</table>

**IfcElementQuantity**

- **GlobalId**
  - Providing a GUID is mandatory, but the GUID is allowed to change.

- **OwnerHistory**
  - Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.

- **Name**
  - Must be “3DModuleSpecificQuantities”.

- **Description**
  - <Open>

- **MethodOfMeasurement**
  - <Open>

- **Quantities**
  - Required, IfcQuantityLength.

**IfcQuantityLength**

- **Name**
  - Must be “GapOnFace1”

- **Description**
  - <Open>

- **Unit**
  - <Open>

- **LengthValue**
  - Required, IfcLengthMeasure.

- **Formula**
  - <Open>

**IfcQuantityLength**

- **Name**
  - Must be “GapOnFace2”

- **Description**
  - <Open>

- **Unit**
  - <Open>

- **LengthValue**
  - Required, IfcLengthMeasure.

- **Formula**
  - <Open>

**IfcQuantityLength**

- **Name**
  - Must be “GapOnFace3”

- **Description**
  - <Open>

- **Unit**
  - <Open>

- **LengthValue**
  - Required, IfcLengthMeasure.

- **Formula**
  - <Open>

**IfcQuantityLength**

- **Name**
  - Must be “GapOnFace4”

- **Description**
  - <Open>

- **Unit**
  - <Open>

- **LengthValue**
  - Required, IfcLengthMeasure.

- **Formula**
  - <Open>

**IfcQuantityLength**

- **Name**
  - Must be “GapOnFace5”

- **Description**
  - <Open>

- **Unit**
  - <Open>

- **LengthValue**
  - Required, IfcLengthMeasure.

- **Formula**
  - <Open>

**IfcQuantityLength**

- **Name**
  - Must be “GapOnFace6”

- **Description**
  - <Open>

- **Unit**
  - <Open>

- **LengthValue**
  - Required, IfcLengthMeasure.

- **Formula**
  - <Open>

---

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Module Common Limitation

Reference | PSU-028 | Version | 1.0 | Status | Draft
--- | --- | --- | --- | --- | ---
Relationships | Extends adapter concept “Property Set”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IfcRelDefinesByProperties
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- RelatedObjects: Required, IfcBuildingElementProxy/(ObjectType="PrefabricatedModule").
- RelatingPropertyDefinition: Required, IfcPropertySet

IfcPropertySet
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: Must be IfcLabel = “ModuleCommonLimitations”.
- Description: <Open>
- HasProperties: <Open>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Mass Limit

Reference | PSU-030 | Version | 1.0 | Status | Draft
--- | --- | --- | --- | --- | ---

Relationships
Extends adapter concept “Module Common Limitations”

History
Created 01.01.2015
Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

<table>
<thead>
<tr>
<th>IfcRelDefinesByProperties</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
</tr>
<tr>
<td>OwnerHistory</td>
<td>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</td>
</tr>
<tr>
<td>Name</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>RelatedObjects</td>
<td>Required, IfcBuildingElementProxy/(ObjectType=&quot;PrefabricatedModule&quot;).</td>
</tr>
<tr>
<td>RelatingPropertyDefinition</td>
<td>Required, IfcPropertySet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcPropertySet</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
</tr>
<tr>
<td>OwnerHistory</td>
<td>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</td>
</tr>
<tr>
<td>Name</td>
<td>Must be IfcLable = “ModuleCommonLimitations”</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>HasProperties</td>
<td>Required, IfcPropertySingleValue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcPropertySingleValue</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HasExternalReferences</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>PropertyForDependance</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>PropertyDependsOn</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>PartOfComplex</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>NominalValue</td>
<td></td>
</tr>
<tr>
<td>Unit</td>
<td></td>
</tr>
<tr>
<td>IfcMassMeasure</td>
<td></td>
</tr>
<tr>
<td>IfcLable = “MassLimitation”</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcBuildingElementProxy</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
<td></td>
</tr>
<tr>
<td>OwnerHistory</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>HasAssignments</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>Nests</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>IsNestedBy</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>HasContext</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>IsDecomposedBy</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>Decomposes</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>HasAssociations</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>ObjectType</td>
<td></td>
</tr>
<tr>
<td>IsDeclaredBy</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>Declares</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>IsTypedBy</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>IsDefinedBy</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>ObjectPlacement</td>
<td></td>
</tr>
<tr>
<td>Representation</td>
<td>ReferencedBy</td>
</tr>
<tr>
<td>Tag</td>
<td></td>
</tr>
<tr>
<td>FileVoids</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>ConnectedTo</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>IsInterfacedByElements</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>InterfersElements</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>HasProjections</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>ReferencedInStructures</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>HasOpenings</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>IsConnectionRealization</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>ProvidesBoundaries</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>ConnectedFrom</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>ContainedInStructure</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>HasCoverings</td>
<td>S[0:7]</td>
</tr>
<tr>
<td>PredefinedType</td>
<td></td>
</tr>
<tr>
<td>IfcLable = “PrefabricatedModule”</td>
<td></td>
</tr>
<tr>
<td>IfcPropertySingleValue</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Name</td>
<td>Must be IfcLabel = “MassLimit”.</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>NominalValue</td>
<td>Required, IfcMassMeasure.</td>
</tr>
<tr>
<td>Unit</td>
<td>Required</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium — All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

**Dimensional Limitations**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-032</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

**Relationships**
Extends adapter concept "Module Common Limitations"

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

**Instantiation diagram**

**Implementation agreements**

<table>
<thead>
<tr>
<th>IfcRelDefinesByProperties</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
</tr>
<tr>
<td>OwnerHistory</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>RelatedObjects</td>
</tr>
<tr>
<td>RelatingPropertyDefinition</td>
</tr>
<tr>
<td>PropertySet</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>GlobalId</td>
</tr>
<tr>
<td>OwnerHistory</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>HasProperties</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcPropertySingleValue</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Must be IfcLabel = &quot;MaximumAllowableLength&quot;</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>NominalValue</td>
<td>Required, IfcPositiveLengthMeasure.</td>
</tr>
<tr>
<td>Unit</td>
<td>Required.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcPropertySingleValue</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Must be IfcLabel = &quot;MaximumAllowableWidth&quot;</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>NominalValue</td>
<td>Required, IfcPositiveLengthMeasure.</td>
</tr>
<tr>
<td>Unit</td>
<td>Required.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcPropertySingleValue</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Must be IfcLabel = &quot;MaximumAllowableHeight&quot;</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>NominalValue</td>
<td>Required, IfcPositiveLengthMeasure.</td>
</tr>
<tr>
<td>Unit</td>
<td>Required.</td>
</tr>
</tbody>
</table>
IFC Release Specific Concept Description (IFC4)

Transportation Information

Reference: PSU-034
Version: 1.0
Status: Draft

Relationships:
Extends adapter concept "Property Set"

History:
Created 01.01.2015

Authors:
Issa Ramaji

Document Owner:
Penn State University

Usage in view definition diagram:

Instantiation diagram:

Implementation agreements:

IfcRelDefinesByProperties
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- RelatedObjects: Required, IfcBuildingElementProxy/(ObjectType="PrefabricatedModule").
- RelatingPropertyDefinition: Required, IfcPropertySet.

IfcPropertySet
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- HasProperties: <Open>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

Transportation Route

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-036</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

Relationships
Extends adapter concept “Transportation Information”

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IfcRelDefinesByProperties
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- RelatedObjects: Required, IfcBuildingElementProxy/(ObjectType="PrefabricatedModule").
- RelatingPropertyDefinition: Required, IfcPropertySet.

PropertySet
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: Must be IfcLabel = “TransportationInformation”.
- Description: <Open>
<table>
<thead>
<tr>
<th>IfcPropertySingleValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>NominalValue</td>
</tr>
<tr>
<td>Unit</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org) Template Licensed for use in buildingSMART Projects – By The BLUS Consortium – All Other Rights Reserved.
IFC Release Specific Concept Description (IFC4)
Transportation Time

Reference | PSU-047
Version    | 1.0
Status     | Draft

Relationships
Extends adapter concept “Transportation Information”

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IFcRelDefinesByProperties

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
</tr>
<tr>
<td>OwnerHistory</td>
<td>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</td>
</tr>
<tr>
<td>Name</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>RelatedObjects</td>
<td>Required, IfcBuildingElementProxy/ObjectType=&quot;PrefabricatedModule&quot;.</td>
</tr>
<tr>
<td>RelatingPropertyDefinition</td>
<td>Required, IfcPropertySet.</td>
</tr>
</tbody>
</table>

IFcPropertySet

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasProperties</td>
<td>Required, IfcPropertySingleValue.</td>
</tr>
</tbody>
</table>

IFcPropertySingleValue

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasExternalReferences</td>
<td>Required, IfcPropertySingleValue.</td>
</tr>
<tr>
<td>Name</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>PartOfProduct</td>
<td>Required, IfcBuildingElementProxy/ObjectType=&quot;PrefabricatedModule&quot;.</td>
</tr>
<tr>
<td>PropertyDependsOn</td>
<td>Required, IfcPropertySingleValue.</td>
</tr>
<tr>
<td>PartOfComplex</td>
<td>Required, IfcPropertySingleValue.</td>
</tr>
<tr>
<td>NominalValue</td>
<td>Required, IfcPropertySingleValue.</td>
</tr>
<tr>
<td>Unit</td>
<td>Required, IfcPropertySingleValue.</td>
</tr>
</tbody>
</table>

IFcPropertySingleValue

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasTimeMeasure</td>
<td>Required, IfcTimeMeasure.</td>
</tr>
<tr>
<td>IfcLabel</td>
<td>“TransportationTime”</td>
</tr>
</tbody>
</table>

IFcTimeMeasure

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IfcLabel</td>
<td>“TransportationInformation”</td>
</tr>
</tbody>
</table>

PropertySet

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasProperties</td>
<td>Required, IfcPropertySingleValue.</td>
</tr>
</tbody>
</table>

PrefabricatedModule

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IfcLabel</td>
<td>“PrefabricatedModule”</td>
</tr>
</tbody>
</table>

Implementation agreements
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Must be IfcLabel = “TransportationTime”.</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>NominalValue</td>
<td>Required, IfcTimeMeasure.</td>
</tr>
<tr>
<td>Unit</td>
<td>Required.</td>
</tr>
</tbody>
</table>
IFC Release Specific Concept Description (IFC4)
Manufacturer

Reference | PSU-041 | Version | 1.0 | Status | Draft
---|---|---|---|---|---

Relationships
Extends adapter concept "Actor Assignment"

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

**IfcRelAssignsToActor**
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: Must be IfcLabel = "ModuleManufacturer".
- Description: <Open>
- RelatedObjects: Required, IfcBuildingElementProxy/(ObjectType="PrefabricatedModule").
- RelatedObjectsType: <Open>
- RelatingActor: Must be IfcActor or its subtype.
- ActingRole: Must be IfcActorRole or its subtype.

**IfcActor**
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- ObjectType: <Open>
- TheActor: Must be IfcOrganization.
<table>
<thead>
<tr>
<th><strong>IfcOrganization</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identification</strong></td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td><strong>Roles</strong></td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td><strong>Addresses</strong></td>
<td>Required, IfcPostalAddress.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>IfcPostalAddress</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purpose</strong></td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Required, IfcText = &quot;PrefabricationPlant&quot;.</td>
</tr>
<tr>
<td><strong>UserDefinedPurpose</strong></td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td><strong>InternalLocation</strong></td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td><strong>AddressLines</strong></td>
<td>Required, IfcLabel.</td>
</tr>
<tr>
<td><strong>PostalBox</strong></td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td><strong>Town</strong></td>
<td>Required, IfcLabel.</td>
</tr>
<tr>
<td><strong>Region</strong></td>
<td>Required, IfcLabel.</td>
</tr>
<tr>
<td><strong>PostalCode</strong></td>
<td>Required, IfcLabel.</td>
</tr>
<tr>
<td><strong>Country</strong></td>
<td>Required, IfcLabel.</td>
</tr>
</tbody>
</table>
IFC Release Specific Concept Description (IFC4)

Transporter

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-043</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Status</td>
<td>Draft</td>
</tr>
</tbody>
</table>

**Relationships**
Extends adapter concept "Actor Assignment"

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

**Implementation agreements**

**IfcRelAssignsToActor**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: Must be IfcLabel = "ModuleTransporter".
- **Description**: <Open>
- **RelatedObjects**: Required, IfcBuildingElementProxy(ObjectType="PrefabricatedModule").
- **RelatedObjectsType**: <Open>
- **RelatingActor**: Must be IfcActor or its subtype.
- **ActingRole**: Must be IfcActorRole or its subtype.

**IfcActor**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
<table>
<thead>
<tr>
<th>Description</th>
<th>&lt;Open&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectType</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>TheActor</td>
<td>Must be IfcOrganization.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>IfcOrganization</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Roles</td>
</tr>
<tr>
<td>Addresses</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)

Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

Generic Nesting

Reference | PSU-045 | Version | 1.0 | Status | Draft
--- | --- | --- | --- | --- | ---

Relationships

History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

| IfcRelNests | 
| --- | --- |
| GlobalId | Providing a GUID is mandatory, but the GUID is allowed to change. |
| OwnerHistory | Providing an OwnerHistory is mandatory, but it is allowed to use dummy data. |
| Name | <Open> |
| Description | <Open> |
| RelatedObjects | Required, IfcObject. |
| RelatingPropertyDefinition | Required, IfcObject. |

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)

Template Licensed for use in buildingSMART Projects — By The BLIS Consortium — All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Hoisting Port Nested in Module

Reference: PSU-047  Version: 1.0  Status: Draft

Relationships: Extends adapter concept “Generic Nesting”

History: Created 01.01.2015

Authors: Issa Ramaji

Document Owner: Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

<table>
<thead>
<tr>
<th>IfcRelNests</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
</tr>
<tr>
<td>OwnerHistory</td>
<td>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</td>
</tr>
<tr>
<td>Name</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>RelatedObjects</td>
<td>Required, IfcBuildingElementProxy/(ObjectType=&quot;PrefabricatedModule&quot;)</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>RelatingPropertyDefinition</td>
<td>Required, IfcPort.</td>
</tr>
</tbody>
</table>

**IfcPort**

- **GlobalId**
  Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**
  Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**
  <Open>
- **Description**
  <Open>
- **ObjectType**
  Required, IfcLabel = “HoistingPoint”.
- **ObjectPlacement**
  <Open>
- **Representation**
  Required, IfcProductDefinitionShape.

**IfcProductDefinitionShape**

- **Name**
  <Open>
- **Description**
  <Open>
- **Representation**
  Required, IfcShapeRepresentation.

**IfcShapeRepresentation**

- **ContextOfItems**
  <Open>
- **RepresentationIdentifier**
  <Open>
- **RepresentationType**
  <Open>
- **Items**
  Required, IfcCartesianPoint.

**IfcCartesianPoint**

- **Coordinates**
  Required.

---

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)

Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Connection Port Nested in Module

Reference: PSU-049  Version: 1.0  Status: Draft

Relationships: Extends adapter concept “Generic Nesting”

History: Created 01.01.2015
Authors: Issa Ramaji
Document Owner: Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

<table>
<thead>
<tr>
<th>IfcRelNests</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
<td></td>
</tr>
<tr>
<td>OwnerHistory</td>
<td>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>&lt;Open&gt;</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
<td></td>
</tr>
<tr>
<td>RelatedObjects</td>
<td>Required, IfcBuildingElementProxy/(ObjectType=&quot;PrefabricatedModule&quot;)</td>
<td></td>
</tr>
<tr>
<td>RelatingPropertyDefinition</td>
<td>Required, IfcPort.</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>IfcPort</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GlobalId</td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
<td></td>
</tr>
<tr>
<td>OwnerHistory</td>
<td>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>&lt;Open&gt;</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
<td></td>
</tr>
<tr>
<td>ObjectType</td>
<td>Required, IfcLable = “ConnectionPoint”.</td>
<td></td>
</tr>
<tr>
<td>ObjectPlacement</td>
<td>&lt;Open&gt;</td>
<td></td>
</tr>
<tr>
<td>Representation</td>
<td>Required, IfcProductDefinitionShape.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcProductDefinitionShape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Representation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcShapeRepresentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContextOfItems</td>
</tr>
<tr>
<td>RepresentationIdentifier</td>
</tr>
<tr>
<td>RepresentationType</td>
</tr>
<tr>
<td>Items</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcCartesianPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinates</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Element Part Aggregation

Reference: PSU-054    Version: 1.0    Status: Draft

Relationships:
Extends adapter concept “Generic Aggregation”

History:
Created 01.01.2015

Authors:
Issa Ramaji

Document Owner: Penn State University

Usage in view definition diagram:

Instantiation diagram:

Implementation agreements:

### IfcRelAggregates
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **RelatingObject**: Required, IfcElement.
- **RelatedObjects**: Required, IfcBuildingElementPart.

### IfcBuildingElementPart
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
<table>
<thead>
<tr>
<th>Description</th>
<th>&lt;Open&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectType</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>ObjectPlacement</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Representation</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Tag</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>PredefinedType</td>
<td>&lt;Open&gt;</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Module Wall

Reference: PSU-056  Version: 1.0  Status: Draft

Relationships: Extends adapter concept “Element Part Aggregation”

History: Created 01.01.2015

Authors: Issa Ramaji

Document Owner: Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IfcRelAggregates
- Globalid: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- RelatingObject: Required, IfcProxy/(ObjectType="PrefabricatedModule"; PredefinedType="Volumetric").
- RelatedObjects: Required, IfcBuildingElementPart.

IfcBuildingElementPart
- Globalid: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
<table>
<thead>
<tr>
<th>Description</th>
<th>&lt;Open&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectType</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>ObjectPlacement</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Representation</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Tag</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>PredefinedType</td>
<td>Required, IfcBuildingElementPartTypeEnum=&quot;ModuleWall&quot;.</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org) Template Licensed for use in buildingSMART Projects – By The BLJS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Module Floor

**Reference** | PSU-058 | **Version** | 1.0 | **Status** | Draft
--- | --- | --- | --- | --- | ---

**Relationships**
Extends adapter concept “Element Part Aggregation”

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

<table>
<thead>
<tr>
<th>IfcRelAggregates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GlobalId</strong></td>
</tr>
<tr>
<td><strong>OwnerHistory</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>RelatingObject</strong></td>
</tr>
<tr>
<td><strong>RelatedObjects</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcBuildingElementPart</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GlobalId</strong></td>
</tr>
<tr>
<td>OwnerHistory</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>ObjectType</td>
</tr>
<tr>
<td>ObjectPlacement</td>
</tr>
<tr>
<td>Representation</td>
</tr>
<tr>
<td>Tag</td>
</tr>
<tr>
<td>PredefinedType</td>
</tr>
</tbody>
</table>
**IFC Release Specific Concept Description (IFC4)**

**Module Ceiling**

<table>
<thead>
<tr>
<th>Reference</th>
<th>Version</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSU-060</td>
<td>1.0</td>
<td>Draft</td>
<td></td>
</tr>
</tbody>
</table>

**Relationships**
Extends adapter concept "Element Part Aggregation"

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

**Instantiation diagram**

**Implementation agreements**

**IfcRelAggregates**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **RelatingObject**: Required, IfcProxy/(ObjectType="PrefabricatedModule"; PredefinedType="Volumetric").
- **RelatedObjects**: Required, IfcBuildingElementPart.

**IfcBuildingElementPart**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
<table>
<thead>
<tr>
<th>OwnerHistory</th>
<th>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>ObjectType</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>ObjectPlacement</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Representation</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Tag</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>PredefinedType</td>
<td>Required, IfcBuildingElementPartTypeEnum=&quot;ModuleCeiling&quot;.</td>
</tr>
</tbody>
</table>
IFC Release Specific Concept Description (IFC4)
Module Column

Reference | PSU-062  | Version | 1.0  | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept “Element Part Aggregation”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

<table>
<thead>
<tr>
<th>IfcRelAggregates</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
</tr>
<tr>
<td>OwnerHistory</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>RelatingObject</td>
</tr>
<tr>
<td>RelatedObjects</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcBuildingElementPart</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
</tr>
<tr>
<td>OwnerHistory</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>ObjectType</td>
</tr>
<tr>
<td>ObjectPlacement</td>
</tr>
<tr>
<td>Representation</td>
</tr>
<tr>
<td>Tag</td>
</tr>
<tr>
<td>PredefinedType</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Generic Module Part

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-064</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

Relationships: Extends adapter concept “Element Part Aggregation”

History: Created 01.01.2015

Authors: Issa Ramaji

Document Owner: Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

<table>
<thead>
<tr>
<th>IfcRelAggregates</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
</tr>
<tr>
<td>OwnerHistory</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>
| RelatingObject   | Required, IfcProxy/(ObjectType="PrefabricatedModule";
|                  | PredefinedType="Volumetric"). |
| RelatedObjects   | Required, IfcBuildingElementPart. |

<table>
<thead>
<tr>
<th>IfcBuildingElementPart</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>ObjectType</td>
</tr>
</tbody>
</table>

IfcBuildingElementProxy

- GlobalId
- OwnerHistory
- Name
- Description
- ObjectPlacement
- Representation
- Tag
- FillsVoids
- ConnectedTo
- InterfacedByElements
- InterfacedElements
- HasProjections
- ReferencedInStructures
- HasOpenings
- IsConnectionRealization
- ProvidesBoundaries
- ConnectedFrom
- ContainedInStructure
- PredefinedType

IfcBuildingElementProxyTypeEnum = "Volumetric"
<table>
<thead>
<tr>
<th>OwnerHistory</th>
<th>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>ObjectType</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>ObjectPlacement</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Representation</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Tag</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>PredefinedType</td>
<td>Required, IfcBuildingElementPartTypeEnum=&quot;GenericModulePart&quot;.</td>
</tr>
</tbody>
</table>
IFC Release Specific Concept Description (IFC4)

2D-Module Specific Properties

Reference | PSU-066 | Version | 1.0 | Status | Draft
--- | --- | --- | --- | --- | ---

Relationships Extends adapter concept “Property Set”

History Created 01.01.2015

Authors Issa Ramaji

Document Owner Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IfcRelDefinesByProperties

GlobalId Providing a GUID is mandatory, but the GUID is allowed to change.

OwnerHistory Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.

Name

Description

RelatedObjects Required, IfcProxy/(ObjectType="PrefabricatedModule";

PredefinedType="Planar").

RelatingPropertyDefinition Required, IfcPropertySet

PropertySet

GlobalId Providing a GUID is mandatory, but the GUID is allowed to change.

OwnerHistory Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.

Name Must be IfcLable = “PlanarModuleSpecificProperties”. 

IfcLabel = “PlanarModuleSpecificProperties”
IFC Release Specific Concept Description (IFC4)
Module Utilization Type

Reference | PSU-067 | Version | 1.0 | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept “2D-Module Specific Properties”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

<table>
<thead>
<tr>
<th>IfcRelDefinesByProperties</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OwnerHistory</td>
<td>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>&lt;Open&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RelatedObjects</td>
<td>Required, IfcProxy/(ObjectType=&quot;PrefabricatedModule&quot;; PredefinedType=&quot;Planar&quot;).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RelatingPropertyDefinition</td>
<td>Required, IfcPropertySet.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PropertySet</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
<td>Providing a GUID is mandatory, but the GUID is allowed to change.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OwnerHistory</td>
<td>Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Must be IfcLable = “PlanarModuleSpecificProperties”.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Predefined Types:
- “Floor”
- “Wall”
- “Ceiling”
- “Envelope”
HasProperties: Required, IfcPropertySingleValue.

<table>
<thead>
<tr>
<th>IfcPropertySingleValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>NominalValue</td>
</tr>
<tr>
<td>Unit</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium –All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

Building Element Aggregation

Reference: PSU-069
Version: 1.0
Status: Draft

Relationships:
Extends adapter concept “Generic Aggregation”

History:
Created 01.01.2015

Authors:
Issa Ramaji

Document Owner:
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium –All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

Port Connection

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-070</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Status</td>
<td>Draft</td>
</tr>
</tbody>
</table>

Relationships
Extends adapter concept “Generic Connectivity”

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium –All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Wall Resultant Structural Properties

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-074</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

Relationships
Extends adapter concept "Property Set"

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

<table>
<thead>
<tr>
<th>IfcRelDefinesByProperties</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
</tr>
<tr>
<td>OwnerHistory</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>RelatedObjects</td>
</tr>
<tr>
<td>RelatingPropertyDefinition</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcPropertySet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchical</td>
</tr>
<tr>
<td>GlobalId</td>
</tr>
<tr>
<td>OwnerHistory</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>HasProperties</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
In-Plane Lateral Load Capacity

Reference | PSU-076 | Version | 1.0 | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept “Wall Resultant Structural Properties”
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IfcRelDefinesByProperties
GlobalId | Providing a GUID is mandatory, but the GUID is allowed to change.  
OwnerHistory | Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.  
Name | <Open>  
Description | <Open>  
RelatedObjects | Required, IfcBuildingElementPart/[PredefinedType="ModuleWall").  
RelatingPropertyDefinition | Required, IfcPropertySet.

PropertySet
GlobalId | Providing a GUID is mandatory, but the GUID is allowed to change.  
OwnerHistory | Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.  
Name | Must be IfcLable = “ModuleWallStructuralProperties”.  
Description | <Open>  
HasProperties | Required, IfcPropertySingleValue.

IfcPropertySingleValue
<table>
<thead>
<tr>
<th>Name</th>
<th>Must be IfcLabel = “In-PlaneLateralLoadCapacity”.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>NominalValue</td>
<td>Required, IfcForceMeasure.</td>
</tr>
<tr>
<td>Unit</td>
<td>&lt;Open&gt;</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iaiernational.org)
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
## IFC Release Specific Concept Description (IFC4)

### In-Plane Lateral Stiffness

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-078</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Status</td>
<td>Draft</td>
</tr>
</tbody>
</table>

**Relationships**
Extends adapter concept “Wall Resultant Structural Properties”

**History**
- Created: 01.01.2015

**Authors**
- Issa Ramaji

**Document Owner**
- Penn State University

### Usage in view definition diagram

- Generic Definition
- Property Definition
- Property Set
- Wall Resultant Structural Properties
- In-Plane Lateral Stiffness

### Instantiation diagram

#### IfcRelDefinesByProperties
- GlobalId
- OwnerHistory
- Name
- Description
- RelatedObjects
- RelatingPropertyDefinition

#### IfcPropertySet
- GlobalId
- OwnerHistory
- Name
- Description
- HasContext
- HasAssociations
- DefinesType
- IsDefinedBy
- DefineOccurrence
- HasProperties

#### IfcPropertySingleValue
- HasExternalReferences
- Name
- Description
- PropertyForDependence
- PropertyDependsOn
- PartOfComplex
- NominalValue
- Unit

#### IfcForceMeasure
- IfcLabel = “In-PlaneLateralStiffness”

### Implementation agreements

#### IfcRelDefinesByProperties
- GlobalId
- OwnerHistory
- Name
- Description
- RelatedObjects
- RelatingPropertyDefinition

#### PropertySet
- GlobalId
- OwnerHistory
- Name
- Description
- HasProperties

**RelatedObjects**
- Required, IfcBuildingElementPart/[PredefinedType="ModuleWall"]

**RelatingPropertyDefinition**
- Required, IfcPropertySet.
<table>
<thead>
<tr>
<th><strong>IfcPropertySingleValue</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>NominalValue</strong></td>
</tr>
<tr>
<td><strong>Unit</strong></td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium –All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Vertical Load Capacity

Reference | PSU-080 | Version | 1.0 | Status | Draft
--- | --- | --- | --- | --- | ---
Relationships | Extends adapter concept "Wall Resultant Structural Properties"
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IIfcRelDefinesByProperties
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- RelatedObjects: Required, IfcBuildingElementPart/[PredefinedType="ModuleWall"]).
- RelatingPropertyDefinition: Required, IfcPropertySet.

IIfcPropertySet
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: Must be IfcLable = "ModuleWallStructuralProperties".
- Description: <Open>

IIfcPropertySingleValue
- HasExternalReferences: <Open>
- Name: <Open>
- Description: <Open>
- RelatedObjects: Required, IfcBuildingElementPart/[PredefinedType="ModuleWall"]).
- RelatingPropertyDefinition: Required, IfcPropertySet.

IIfcRelDefinesByProperties
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- RelatedObjects: Required, IfcBuildingElementPart/[PredefinedType="ModuleWall"]).
- RelatingPropertyDefinition: Required, IfcPropertySet.
<table>
<thead>
<tr>
<th>HasProperties</th>
<th>Required, IfcPropertySingleValue.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IfcPropertySingleValue</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Must be IfcLabel = “VerticalLoadCapacity”.</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>NominalValue</td>
<td>Required, IfcForceMeasure.</td>
</tr>
<tr>
<td>Unit</td>
<td>&lt;Open&gt;</td>
</tr>
</tbody>
</table>
**IFC4 Release Specific Concept Description (IFC4)**

**Vertical Stiffness**

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-082</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Status</td>
<td>Draft</td>
</tr>
</tbody>
</table>

**Relationships**
Extends adapter concept "Wall Resultant Structural Properties"

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

**Instantiation diagram**

**Implementation agreements**

---

**IfcRelDefinesByProperties**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **RelatedObjects**: Required, IfcBuildingElementPart/[PredefinedType=“ModuleWall”].
- **RelatingPropertyDefinition**: Required, IfcPropertySet.

**IfcPropertySet**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: Must be IfcLabel = “ModuleWallStructuralProperties”.
- **Description**: <Open>

**IfcPropertySingleValue**
- **HasExternalReferences**: S[0:7]
- **Name**: Description
- **PropertyForDependence**: S[0:7]
- **PropertyDependsOn**: S[0:7]
- **PartOfComplex**: S[0:7]
- **NominalValue**: Unit

**IfcForceMeasure**
- **IfcLabel**: "VerticalStiffness"
<table>
<thead>
<tr>
<th>HasProperties</th>
<th>Required, IfcPropertySingleValue.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IfcPropertySingleValue</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Must be IfcLabel = “VerticalStiffness”.</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>NominalValue</td>
<td>Required, IfcLinearStiffnessMeasure.</td>
</tr>
<tr>
<td>Unit</td>
<td>&lt;Open&gt;</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)
Planar Element Part Base Quantities

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-084</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

Relationships
Extends adapter concept “Simple Element Quantities”

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

**IfcRelDefinesByProperties**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **RelatedObjects**: Required, IfcBuildingElementProxy.
- **RelatingPropertyDefinition**: Required, IfcElementQuantity.

**IfcElementQuantity**
- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: Must be “PlanarElementPartBaseQuantities”.
- **Description**: <Open>
- **MethodOfMeasurement**: <Open>
- **Quantities**: <Open>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
## IFC Release Specific Concept Description (IFC4)
### Planar Element Part Dimensions

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-086</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

### Relationships
Extends adapter concept “Planar Element Part Base Quantities”

### History
Created 01.01.2015

### Authors
Issa Ramaji

### Document Owner
Penn State University

### Usage in view definition diagram

![Diagram Image]

### Instantiation diagram

![Diagram Image]

### Implementation agreements

#### IfcRelDefinesByProperties
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- RelatedObjects: Required, IfcBuildingElementProxy.
- RelatingPropertyDefinition: Required, IfcElementQuantity.

#### IfcElementQuantity
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: Must be “PlanarElementPartBaseQuantities”.
- Description: <Open>
- MethodOfMeasurement: <Open>
- Quantities: Required, IfcQuantityLength.

#### IfcQuantityLength
- Name: Must be “DimensionInDirection1”
- Description: <Open>
<table>
<thead>
<tr>
<th>Unit</th>
<th>&lt;Open&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>LengthValue</td>
<td>Required, IfcLengthMeasure.</td>
</tr>
<tr>
<td>Formula</td>
<td>&lt;Open&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IfcQuantityLength</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Must be “DimensionInDirection2”</td>
</tr>
<tr>
<td>Description</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>Unit</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>LengthValue</td>
<td>Required, IfcLengthMeasure.</td>
</tr>
<tr>
<td>Formula</td>
<td>&lt;Open&gt;</td>
</tr>
</tbody>
</table>
IFC Release Specific Concept Description (IFC4)

Floor Resultant Structural Properties

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-088</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

Relationships
Extends adapter concept "Property Set"

History
Created 01.01.2015

Authors
Issa Ramaji

Document Owner
Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IfcRelDefinesByProperties
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: <Open>
- Description: <Open>
- RelatedObjects: Required, IfcBuildingElementPart/(PredefinedType="ModuleFloor").
- RelatingPropertyDefinition: Required, IfcPropertySet.

PropertySet
- GlobalId: Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name: Must be IfcLable = "ModuleFloorStructuralProperties".
- Description: <Open>
- HasProperties: <Open>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

Floor Occupancy Type

Reference | PSU-090 | Version | 1.0 | Status | Draft

Relationships | Extends adapter concept “Floor Resultant Structural Properties”

History | Created 01.01.2015

Authors | Issa Ramaji

Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

IfcRelDefinesByProperties

- GlobalId Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name <Open>
- Description <Open>
- RelatedObjects Required, IfcBuildingElementPart/[PredefinedType="ModuleFloor"]/.
- RelatingPropertyDefinition Required, IfcPropertySet.

IfcPropertySet

- GlobalId Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name Must be IfcLable = "ModuleFloorStructuralProperties".
- Description <Open>
- HasProperties Required, IfcPropertySingleValue.

IfcPropertySingleValue

- HasExternalReferences S[0:?]
- Name <Open>
- Description <Open>
- PartOfPset S[0:?]
- PropertyForDependence S[0:?]
- PropertyDependsOn S[0:?]
- PartOfComplex S[0:?]
- NominalValue
- Unit

IfcLable = "ModuleFloorStructuralProperties"

IfcBuildingElementPart

- GlobalId Providing a GUID is mandatory, but the GUID is allowed to change.
- OwnerHistory Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- Name <Open>
- Description <Open>
- HasProperties Required, IfcPropertySingleValue.

IfcLable = "Floor Occupancy Type"
<table>
<thead>
<tr>
<th>IfcPropertySingleValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>NominalValue</td>
</tr>
<tr>
<td>Unit</td>
</tr>
</tbody>
</table>

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
IFC Release Specific Concept Description (IFC4)

Floor Diaphragm Rigidity

Reference PSU-092  Version 1.0  Status Draft

Relationships Extends adapter concept “Floor Resultant Structural Properties”

History Created 01.01.2015

Authors Issa Ramaji

Document Owner Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

**IfcRelDefinesByProperties**
- **GlobalId** Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory** Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name** <Open>
- **Description** <Open>
- **RelatedObjects** Required, IfcBuildingElementPart/(PredefinedType="ModuleFloor").
- **RelatingPropertyDefinition** Required, IfcPropertySet.

**PropertySet**
- **GlobalId** Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory** Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name** Must be IfcLable = “ModuleFloorStructuralProperties”.
- **Description** <Open>
- **HasProperties** Required, IfcPropertySingleValue.
<table>
<thead>
<tr>
<th><strong>IfcPropertySingleValue</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>NominalValue</strong></td>
</tr>
<tr>
<td><strong>Unit</strong></td>
</tr>
</tbody>
</table>
IFC Release Specific Concept Description (IFC4)

Ceiling Resultant Structural Properties

<table>
<thead>
<tr>
<th>Reference</th>
<th>PSU-094</th>
<th>Version</th>
<th>1.0</th>
<th>Status</th>
<th>Draft</th>
</tr>
</thead>
</table>

Relationships

Extends adapter concept “Property Set”

History

Created 01.01.2015

Authors

Issa Ramaji

Document Owner

Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

**IfcRelDefinesByProperties**

- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **RelatedObjects**: Required, IfcBuildingElementPart/(PredefinedType="ModuleCeiling").
- **RelatingPropertyDefinition**: Required, IfcPropertySet.

**IfcPropertySet**

- **GlobalId**: Providing a GUID is mandatory, but the GUID is allowed to change.
- **OwnerHistory**: Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.
- **Name**: <Open>
- **Description**: <Open>
- **HasProperties**: Must be IfcLable = "ModuleCeilingStructuralProperties".

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects – By The BLIS Consortium – All Other Rights Reserved
Ceiling Diaphragm Rigidity

**Reference** | PSU-096  
**Version** | 1.0  
**Status** | Draft

**Relationships**
Extends adapter concept “Ceiling Resultant Structural Properties”

**History**
Created 01.01.2015

**Authors**
Issa Ramaji

**Document Owner**
Penn State University

**Usage in view definition diagram**

**Instantiation diagram**

**Implementation agreements**

| **IfcRelDefinesByProperties** |  
| GlobalId | Providing a GUID is mandatory, but the GUID is allowed to change.  
| OwnerHistory | Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.  
| Name | <Open>  
| Description | <Open>  
| RelatedObjects | Required, IfcBuildingElementPart/PredefinedType="ModuleCeiling".  
| RelatingPropertyDefinition | Required, IfcPropertySet.

| **PropertySet** |  
| GlobalId | Providing a GUID is mandatory, but the GUID is allowed to change.  
| OwnerHistory | Providing an OwnerHistory is mandatory, but it is allowed to use dummy data.  
| Name | Must be IfcLabel = "ModuleCeilingStructuralProperties".  
| Description | <Open>  
| HasProperties | Required, IfcPropertySingleValue.  

**IfcPropertySingleValue**

| HasExternalReferences | S[0..7]  
| Description |  
| PartOfPset | S[0..7]  
| PropertyForDependance | S[0..7]  
| PropertyDependsOn | S[0..7]  
| PartOfComplex | S[0..7]  
| NominalValue |  

IfcLabel = "Rigid", "Semi-Rigid" or "Non-Rigid"
<table>
<thead>
<tr>
<th><strong>IfcPropertySingleValue</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>NominalValue</strong></td>
</tr>
<tr>
<td><strong>Unit</strong></td>
</tr>
</tbody>
</table>
IFC Release Specific Concept Description (IFC4)
Structural Model Assignment

Reference | PSU-098 | Version | 1.0 | Status | Draft
---|---|---|---|---|---
Relationships | Extends adapter concept "Generic Assignments"
History | Created 01.01.2015
Authors | Issa Ramaji
Document Owner | Penn State University

Usage in view definition diagram

Instantiation diagram

Implementation agreements

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)
Template Licensed for use in buildingSMART Projects –By The BLIS Consortium –All Other Rights Reserved
<table>
<thead>
<tr>
<th>VIEW ID</th>
<th>VIEW NAME</th>
<th>APPLICATION NAME</th>
<th>APP. VERSION</th>
<th>EXCHANGE TYPE</th>
<th>DIAGRAM STATUS</th>
<th>DIAGRAM VERSION</th>
<th>DIAGRAM DATE</th>
<th>DIAGRAM AUTHORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSU-001</td>
<td>Modular Buildings-Architectural Design to Structural Design/Analysis</td>
<td>Generic</td>
<td>Generic</td>
<td>Generic</td>
<td>Draft</td>
<td>1.0</td>
<td>01/01/2015</td>
<td>Issa Ramaji</td>
</tr>
</tbody>
</table>

**Diagram: IFC Model View Definition Diagram: Building**

- **PSU-005 Building**
  - **MVC-006 Spatial Container Identification**
    - **MVC-066 Unique Identifier**
    - **MVC-067 Name**
    - **VBL-056 BIM Object Owner/History**
    - **MVC-077 Postal Address**
    - **MVC-023 Building Storey in Spatial Container**
      - **MVC-296 Space in Spatial Container**
      - **VBL-353 Structural Analysis Model Contained in Building**
    - **PSU-008 Building Part**
  - **MVC-035 Building Properties**
  - **VBL-396 Contained in Building**
  - **MVC-929 Local Absolute Placement**

*Document owner: Penn State University -- This document uses the official IFC Model View Definition Format version 1.1, of the IAI (www.iai-international.org).*
MVC-006 Spatial Container Identification
MVC-906 Building Storey Properties
VBL-400 Contained in Building Storey
MVC-073 Origin (Placement)

MVC-360 Unique Identifier
MVC-067 Name
VBL-056 BIM Object Owner/History

VBL-402 Space in Spatial Container
VBL-402 Building Element Contained in Building Storey
VBL-354 Structural Analysis Model Contained in Building Storey
IFC Model View Definition Diagram: Building Part

VIEW ID: PSU-001
VIEW NAME: Modular Buildings - Architectural Design to Structural Design/Analysis
APPLICATION NAME: Generic
APP. VERSION: Generic
EXCHANGE TYPE: Generic
DIAGRAM STATUS: Draft
DIAGRAM VERSION: 1.0
DIAGRAM DATE: 01/01/2015
DIAGRAM AUTHORS: Issa Ramaji

MVD-PSU-001
Building Part

MVC-066
Spatial Container Identification

MVC-360
Unique Identifier

MVC-067
Name

MVC-080
Spatial Zone Properties

VBL-056
BIM Object Owner/History

PSU-010
Contained in Building Part

PSU-011
Spatial Zone Aggregation

PSU-013
Type Assignment

Document owner: Penn State University -- This document uses the official IFC Model View Definition Format version 1.1, of the IAI (www.iai-international.org).
<table>
<thead>
<tr>
<th>VIEW ID</th>
<th>VIEW NAME</th>
<th>APPLICATION NAME</th>
<th>APP. VERSION</th>
<th>EXCHANGE TYPE</th>
<th>DIAGRAM STATUS</th>
<th>DIAGRAM VERSION</th>
<th>DIAGRAM DATE</th>
<th>DIAGRAM AUTHORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSU-001</td>
<td>Modular Buildings-Architectural Design to Structural Design/Analysis</td>
<td>Generic</td>
<td>Generic</td>
<td>Generic</td>
<td>Draft</td>
<td>1.0</td>
<td>01/01/2015</td>
<td>Issa Ramaji</td>
</tr>
</tbody>
</table>

**IFC Model View Definition Diagram: Module Wall**

- **Object Common Properties**
  - Unique Identifier
  - Name
  - BIM Object Owner/History

- **Module Wall Structural Properties**
  - In-Plain Lateral Load Capacity
  - In-Plain Lateral Stiffness
  - Vertical Load Capacity
  - Vertical Stiffness

- **Planar Element Part Properties**
  - Planar Element Part Dimensions

- **Module Part Placement**
  - Origin (Placement)

- **Module Part Representation**
  - Aggregated Shape Representation

- **Building Element Aggregation**

---

**Document owner: Penn State University** -- This document uses the official IFC Model View Definition Format version 1.1, of the IAI (www.iai-international.org).
Document owner: Penn State University -- This document uses the official IFC Model View Definition Format version 1.1. of the IAI (www.iai-international.org).
Document owner: Penn State University -- This document uses the official IFC Model View Definition Format version 1.1. of the IAI (www.iai-international.org).
<table>
<thead>
<tr>
<th>VIEW ID</th>
<th>VIEW NAME</th>
<th>APPLICATION NAME</th>
<th>APP. VERSION</th>
<th>EXCHANGE TYPE</th>
<th>DIAGRAM STATUS</th>
<th>DIAGRAM VERSION</th>
<th>DIAGRAM DATE</th>
<th>DIAGRAM AUTHORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSU-001</td>
<td>Modular Buildings-Architectural Design to Structural Design/Analysis</td>
<td>Generic</td>
<td>Generic</td>
<td>Generic</td>
<td>Draft</td>
<td>1.0</td>
<td>01/01/2015</td>
<td>Issa Ramaji</td>
</tr>
</tbody>
</table>

**Document owner: Penn State University** -- This document uses the official IFC Model View Definition Format version 1.1. of the IAI (www.iai-international.org).
IFC Model View Definition Diagram: Module Wall IFC4

- VIEW ID: PSU-001
- VIEW NAME: Modular Buildings - Architectural Design to Structural Design/Analysis
- APPLICATION NAME: Generic
- APP. VERSION: Generic
- EXCHANGE TYPE: Generic
- DIAGRAM STATUS: Draft
- DIAGRAM VERSION: 1.0
- DIAGRAM DATE: 01/01/2015
- DIAGRAM AUTHORS: Issa Ramaji

MVD-PSU-001 - IFC4
Module Wall

- METHOD OF USE: To be used in Architectural to Structural BIM information exchanges

MVC-581
Root Attributes
- IFC Global ID
- Name
- Description
- BIM Object Owner/History

MVC-811
Shape Representation
- Aggregated Shape Representation
- Body
- Bounding Box
- Topological Representation

MVC-855
Generic Definition
- Property Definition
- Property Set
- Wall Resultant Structural Properties
- In-Plain Lateral Load Capacity
- In-Plain Lateral Stiffness
- Vertical Load Capacity
- Vertical Stiffness
- Planar Element Part Dimensions

MVC-854
Shape Representation
- Simple Element Quantities
- Planar Element Part Base Quantities
- Planar Element Part Dimensions

MVC-853
Generic Object Placement
- Local Relative Placement

MVC-852
Generic Aggregation
- Building Element Aggregation

Note: This document uses the official IFC Model View Definition Format version 1.1, of the IAI (www.iai-international.org).
IFC Model View Definition Diagram: Module Column IFC4

**VIEW ID**: PSU-001

**VIEW NAME**: Modular Buildings Architectural Design to Structural Design/Analysis

**APPLICATION NAME**: Generic

**APP. VERSION**: Generic

**EXCHANGE TYPE**: Generic

**DIAGRAM STATUS**: Draft

**DIAGRAM VERSION**: 1.0

**DIAGRAM DATE**: 01/01/2015

**DIAGRAM AUTHORS**: Issa Ramaji

---

**MVC-581** Module Column

- **MVC-583** Root Attributes
  - **MVC-584** IFC Global ID
    - **MVC-585** Name
      - **MVC-586** Description
        - **MVC-587** BIM Object Owner/History
  - **MVC-588** Shape Representation
    - **MVC-589** Aggregated Shape Representation
  - **MVC-590** Generic Object Placement
    - **MVC-591** Local Relative Placement
    - **MVC-592** Building Element Aggregation
  - **MVC-593** Generic Aggregation

---

**MVC-611** Body

- **MVC-612** Swept Solid
  - **MVC-613** Linear Extruded Solid

---

**MVC-601** Bounding Box

- **MVC-602** Topological Representation
  - **MVC-603** To topological Straight Edge Representation
  - **MVC-604** To topological Curved Edge Representation

---

Document owner: Penn State University — This document uses the official IFC Model View Definition Format version 1.1 of the IAI (www.iai-international.org).
Document owner: Penn State University – This document uses the official IFC Model View Definition Format version 1.1. of IAI (www.iai-international.org).
## IFC Model View Definition Diagram: Structural Analysis Model IFC4

### Diagram Elements:
- **Root Attributes**
  - IFC Global ID
  - Name
  - Description
  - BIM Object Owner/History

- **Structural Analysis Model Attributes**
  - Generic Loaded by
  - Has Results
  - Structural Analysis Model Predefined Type

- **Generic Assignments**
  - System Assignment
  - Generic Grouping

- **Nesting**
  - Structural Analysis Model Nesting Parent
  - Structural Analysis Model Nesting Child

### Table: View Information

<table>
<thead>
<tr>
<th>VIEW ID</th>
<th>VIEW NAME</th>
<th>APPLICATION NAME</th>
<th>APP. VERSION</th>
<th>EXCHANGE TYPE</th>
<th>DIAGRAM STATUS</th>
<th>DIAGRAM VERSION</th>
<th>DIAGRAM DATE</th>
<th>DIAGRAM AUTHORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSU-001</td>
<td>Modular Buildings Architectural Design to Structural Design/Analysis</td>
<td>Generic</td>
<td>Generic</td>
<td>Generic</td>
<td>Draft</td>
<td>1.0</td>
<td>01/01/2015</td>
<td>Issa Ramaji</td>
</tr>
</tbody>
</table>

---

**Document owner:** Penn State University  -- This document uses the official IFC Model View Definition Format version 1.1. of the IAI [www.iai-international.org].
VITA
Issa Jafar Ramaji
ramaji.issa@gmail.com

Education:
• PhD Candidate, Architectural Engineering
  Pennsylvania State University, PA, USA
  08.2012 – 05.2016
• Master of Science in Civil Engineering
  Sharif University of Technology (SUT), Tehran, Iran
  09.2007 – 01.2010
• Bachelor of Science in Civil Engineering
  Sharif University of Technology (SUT), Tehran, Iran
  09.2003 – 09.2007

Work Experience:
• Band Engineering and Construction Company, Iran
  08.2009 – 08.2012
• Haraz-Rah Engineering Company, Iran
  07.2007 – 08.2009
• Kayson Construction Company, Iran
  06.2006 – 09.2006

Selected Publications:

Academic Honors:
• Winner of second best paper and presentation award from the 12th College of Engineering Research Symposium 2015
• Winner of the Roger J. Glunt Graduate Fellowship 2014 and 2015
• Winner of the Chicago Committee on High Rise Buildings (CCHRB) Scholarship 2015
• Winner of the American Society of Civil Engineers (ASCE) Student Award 2014
• Winner of the Hankins Endowment Scholarship 2014
• Winner of the “Design Excellence” and “Design Integrity” awards of the DOE Home Design Competition for two years in a row (I led the “Enclosure Design” and “Structural Design” groups of the Penn State University team) 2014-2015
• Winner of the Builders Association of Central PA Scholarship 2013 and 2015