www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

FORMULATION OF SYSTEMS AND INFORMATION ARCHITECTURE HIERARCHIES FOR BUILDING STRUCTURES

SUBMITTED: September 2012
REVISED: May 2013
PUBLISHED: August 2013 at http://www.itcon.org/2013/13
EDITOR: Peter Katranuschkov

Ryan Solnosky, EIT,
Ph.D. Candidate, Department of Architectural Engineering, The Pennsylvania State University, U.S.A.
email: rls5008@psu.edu

John Hill,
Instructor, College of Information Sciences and Technology, The Pennsylvania State University, U.S.A.
email: john.hill@psu.edu

SUMMARY: *Building Information Modeling (BIM) heavily utilizes information in various processes throughout the project lifecycle. However, with the magnitude of information needed, misrepresentation or missing data typically leads to interoperability issues. To ensure proper interoperability, a robust information architecture that encompasses a broad spectrum of the topic is needed to develop the most accurate and appropriate schemas. Within the AEC industry the structural domain is the forerunner of data exchange development initiatives. However, there is still a lack of a larger and more comprehensive view to information architectures within structures. The research presented here introduces a novel approach to the foundation of a detailed information architecture for the structural information domain based on systems, sub-systems, elements, components, and supporting resources within the AEC industry. This information architecture represented in aggregated hierarchies provides a breakdown of information related to projects from varying perspectives including product, process control, feedback and constraint information which is generated by capturing, analyzing, retrieving, and formulating.*

KEYWORDS: *Building Information Modeling, Structural Systems, Information Exchanges.*

REFERENCE: *Ryan Solnosky, John Hill (2013). Formulation of systems and information architecture hierarchies for building structures, Journal of Information Technology in Construction (ITcon), Vol. 18, pg. 261-278, http://www.itcon.org/2013/13*

1. INTRODUCTION

The success of Building Information Modeling (BIM) is presently being observed. However, for the information aspect of BIM to become widely utilized for its full potential, as literature has suggested (Eastman et al. 2011), serious work still needs to be conducted to investigate the many issues remaining. Solnosky (2013a) documents in detail the relevant issues in the structural domain. Among these are: legal and contractual conditions to information modeling, the interoperability of software, and what and when to model to name a few. The sector that is most referenced in needing improvement is information interoperability between software (NIBS 2007). This is due to the metadata that describes entities and attributes are not yet being shared effectively in interfaces.

To better understand why this occurs, an excerpt from the US National Building Information Modeling Standard's (NBIMS) which defines BIM is provided as follows: "an improved planning, design, construction, operation, and maintenance process using a standardized machine-readable information model for each facility, new or old, which contains all appropriate information created or gathered about that facility in a format usable by all throughout its lifecycle" (NIBS 2007). This standardized machine-readable information is a focal point in the definition of what BIM is. Venugopal et al. (2012a) further support information by commenting that that robust knowledge in sharing information between stakeholders should be of the highest priority. Coincidently, this standardized information is still in early phases of being automated.

The results from a study by McGraw-Hill Construction (2008) show that of the information and elements to be modeled within a building, those with the highest demand, are: structural systems elements, mechanical equipment, and enclosures elements. Emkin (1988) supported this in that until recently structural applications have focused heavily on the development of solution methods and software modifications for reducing computational time. As there is still much to be done with structural information modeling and a lack of uniformity amongst research and industry is still present, a more uniform hierarchy on structural information is needed to develop more robust and comprehensive solutions.

The formulated discussion, results, and recommendations presented here are part of a larger initiative called: Integrated Structural Process Model (ISPM). The focus of the ISPM is an integrated approach to structural design and construction for a given building project and the information needed at each stage (Solnosky 2012). A structural domain based information architecture for structuring the information that organizations, within the AEC industry, can capture, analyze, retrieve, and send, is presented here. In addition, structural systems, assemblies, elements and components have been aggregated to determine similarities to associate with the referenced and formulated information encompassing their definitions (Solnosky 2013b). These formatted hierarchies can be of great value to those who are developing modeling schemas and software packages to ensure the information is captured and represented accurately.

2. BACKGROUND

Ergen and Akinci (2008) found that researchers and practitioners agree that for a project to be completed successfully, accurate and relevant information should be available to the right people at the right time. This is a concept that requires seamless movement of information from one platform to another and/or from one person to another. This transfer process can produce a large volume of information that needs a classified and organized structure for seamlessness (Burt 2009). Such models contain data with rich attributes including: size, weight, materials, cost specifications, etc. (Jacobi 2007).

In terms of the modeling of this information, Björk (1989) stated that each item of information should only be defined within a model once even if it's used multiple times and for multiple reasons. A time proven principle of information science, this still holds true. For these exchanges to be both effective and efficient, it is best that the data be represented in this manner even if it may be used in many places throughout the project's lifecycle for different reasons. Messner (2003) inherently found a rationalized model to organize information but it was only projected for a single building project in terms of scope. A broader and more structured classification of information is critically needed for this to be achieved.

2.1 Information Architectures

Information can be classified in a variety of manners depending on the perspective taken. In the AEC domain these can include: the type of function or attribute (Eastman et al. 2010), constraint (Luth 1991), feedback, and control (Sanvido et al. 1995) classes. Using this breakdown, the concept of a hierarchy is formulated and the content is classified from the viewpoint of structural systems. A data structure for a generalized software architecture can then be developed so that it can apply to all structural systems (material based), and not only selectively. Two different approaches have been suggested in hierarchy architectures. They employ functional views and compositional views (Ekholm & Haggstrom 2011). A functional view of a system focuses on what it does in relating to the environment. A compositional view of a system focuses on the parts of a system from which that system is composed. If workflows are defined at an information level and traced, it is important to distinguish the information returned through cyclical repetition of activities from one generation to another (Lee et al. 2006). This implies that the information architecture must distinguish functions from classification definitions. With respect to structural systems, Sause and Powell (1990) state that the hierarchy model must: (1) support several different applications; (2) have well-defined entities, relationships, and dependencies; and (3) enable the creation of abstractions of real structures.

A method of modeling commonly used by IT specialists to depict information hierarchies is provided by the Unified Modeling Language’s (UML) object-oriented structural modeling diagrams (Dennis et al. 2009). At a functional level, UML models present a logical organization of the data without indicating how the data are stored, created, or manipulated, as such limiting potential errors (Ergen & Akinci 2008; Dennis et al. 2009). The UML language is particularly useful in modeling discrete component based systems elements such as steel beams but it has problems with monolithic based shapes such as continuous slabs with drop panels (Barak et al. 2009), unless aspects of monolithic shapes can be represented discretely (Anil et al. 2012). The authors recommend this notation for the creation of detailed hierarchies yet this study uses a combination of the UML methodology with a simpler representation for the entities. That is to say, those who may read this work will understand the discussion without having to be familiar with the UML notation.

2.2 Interoperability Limitations

The movement of information that takes place throughout a construction project's lifecycle is typically enabled by software or hardware to some extent. They perform one of two functions: retrieving data (e. g. about processes or material properties) or executing algorithms (e.g. performing structural analysis or member optimization). Interoperability, a property of information processing systems, relies extensively on the communication of data between data processing elements (Burt 2009; Gayer 2009). However, today many information processing systems do not provide all of the capabilities necessary to transfer all of the data between the processing elements. Gaps exist that result in incomplete exchanges of information. The result of these gaps is that exchanges of data depend on individual, often proprietary, applications and only provide limited-scope schemas with other applications (Burt 2009; Ikerd 2010). Barriers such as these slow the process or misrepresent information in the models. To facilitate robust interoperability that allows for more open sharing of data, thus making the design and construction lifecycle more efficient, data structures and formats need to be carefully conceived and constructed and they have to be standardized.

There are three primary types of interoperability formats being used currently in the structural industry (Gayer 2009; Eastman et al. 2010):

1. Proprietary File Formats occur when individual software vendors develop their own mapping structure. Other vendors support such formats only in a limited scope as they seldom correlate to their development agendas.
2. Application Programming Interfaces (API) allow software engineers to write translators between software components but often these translators and API structures are limited to advanced parameters such as material behavior properties and special boundary conditions.
3. Industry Foundation Classes (IFC) is a vendor-neutral model schema developed with the intent to be read by any software application. As such, IFC aims to be an open standard such that any developer can adopt it into their software language.

US NBIMS promotes the third interoperability type, IFC, for use as the primary and preferred schema for Information Delivery Manual (IDM) and Model View Definition (MVD) development (NIBS 2007). IFC was chosen because it has the greatest potential of being successfully adopted by many vendors and parties due to its open format. While in theory this is true, yet in practice currently IFC is the least utilized format due to its stage of development in relation to its applicability for adoption by software developers (Eastman et al. 2010). Software developers can, at times, be hesitant to adopt IFC as the schemas are not complete. Furthermore, they do not directly see a benefit to developing these schemas as they are open source and could, as a result, weaken their competitive advantage in what each company is capable of doing based on APIs. A study by the US Applied Technology Council (ATC) started several initiatives to work on developing the schemas for several structural systems. The relevant ATC document is ATC 75: Development IFCs for the Structural Domain: NBIMS Abstract Process for Exchange Standard Development. Though the study was tailored towards BIM, it was not the earliest work in the structural domain data structures (cf. e.g. Powell & Bhateja 1988).

ATC 75 created a framework and a pilot set of IFCs for structural elements (Gayer 2009). The initial focus was on the exchange of member geometry and properties (ATC 2008) while it also sought to improve productivity in design and construction. This was achieved by taking the lead in developing a basis for incorporating and integrating structural design codes, analysis tools, and methods into the IFCs of the International Alliance for Interoperability (IAI) larger effort (ATC 2008). Since then, there have been several other IFC development initiatives for the structural domain in the frames of the buildingSMART initiative, including but not limited to Norway's Structural Design IDM, Architectural Precast Concrete's IDM, American Institute of Steel Construction's (AISC's) structural steel conversion from CIS/2, and the American Concrete Institute's concrete IDM. ATC and the other works continue to be significant in structuring the data to make structural systems complete. The results of these studies were not all inclusive on what should be covered throughout the structural domain as a whole in terms of information types and depth of the information. Venugopal et al. (2012b) supports this in that they state the current granularity of model views and schemas are not consistent across the industry. The results presented here are material and method non-specific, thus allowing for a broader application to the structural domain.

3. PROPOSED INFORMATION ARCHITECTURE HIERARCHIES FOR BUILDING STRUCTURES

3.1 Research Methodology

This study was conducted through a rigorous qualitative methodology consisting of two phases. The first was a detailed literature review followed by a series of industry expert interviews to confirm consistency and accuracy upheld the results presented. The literature review examined previous works that were published over the last 20 years and focused on information hierarchies, structural classifications, and process models to establish a baseline. Building on that baseline, a series of 30 interviews (25 experts in structures) were conducted on industry experts having BIM expertise and structural design and construction expertise. Sample statistics to support these individuals as experts are listed in Table 1. For more complete information refer to Solnosky (2013b). The titles and ranks for these experts ranged from project engineers and job captains to managers, principles and associates and in some cases there were directors of modeling and design, vice presidents, and chief technical officers.

TABLE 1: Structural Industry Expert Statistics

Participant Class	Count	Avg. Years of Experience		Have Licensure			
		Structural	BIM	PE	SE	LEED	RA
Software	4	15.3	5.7	4	2	0	0
Structural Engineer							
Consultant	8	22.0	5.0	8	2	1	0
BIM Based	8	20.0	10.4	7	2	1	0
AE firm	3	14.0	6.3	1	0	1	1
Detailer/Fabricator	2	24.0	17.0	1	0	0	0
Total	25	19.1	8.9	21	6	3	1

The interviews were in-depth consisting of closed and open ended questions to allow for collection of information. The information collected focused on modeling efforts, information exchanged between trades, and system classifications. To determine the classifications, the interviewees were shown baseline maps for markup after similar information was collected verbally so not to induce bias. Content analysis with decision rule sets were used to sort the data for similar terms and meanings. This information was sorted into the classes presented in the following sections.

3.2 Basis for Hierarchy Structures

For any system, assembly, or element, depending on the complexity and what is expected, a single classification of the data may not be sufficient to describe all aspects in an easy to understand format (Dennis et al. 2009). If we maintain Björk's (1989) thought of data being represented only once, then if we need multiple representations each attribute should have a unique classification though it can be used in different ways. For this to work a hierarchy system based on class and object aggregations and decompositions are often used at various levels

(Björk 1989; Dennis et al. 2009). These concepts were used as a guiding principle when possible to classify the structural systems and information at various levels and complexities for this study. The hierarchical structure of information provides a context to the overall model and helps modelers develop a step-by-step data structure from high-level classes to detailed classes without missing or misrepresenting any critical portion. This abstract hierarchy is used to break a system down into manageable parts and show relations. Alternatively, this research can be thought of as an ontology for systems and information. This can be claimed based on Venugopal et al.'s (2012b) ontological approach to exchanges in pre-stressed concrete. Here a similar approach was taken but Venugopal et al. (2012b) only used one particular material class.

Regarding the structural domain, there are many types of structures with vast and different functions to consider. The prime structural sectors are listed in Fig. 1 and, as indicated by the highlighted portion of the figure, this study focused specifically on building structures (low rise to high rise). This is not to say that what is presented will not work for other systems, instead it is a limitation to this study's validation. The results presented in the following sections describe the hierarchical aggregation/decomposition of structural systems, assemblies, elements and components, and lastly information into the dominant classes. The work does not differentiate between discrete based elements and monolithic based elements because tools and their correlated functions naturally approximate monolithic as discrete to perform their task(s). However, it must be noted that monolithic representation is a major concern potentially and a key reason different domains develop at different rates. As this study further develops beyond this body of work, discrete and monolithic will need to be considered and added within the attributes.

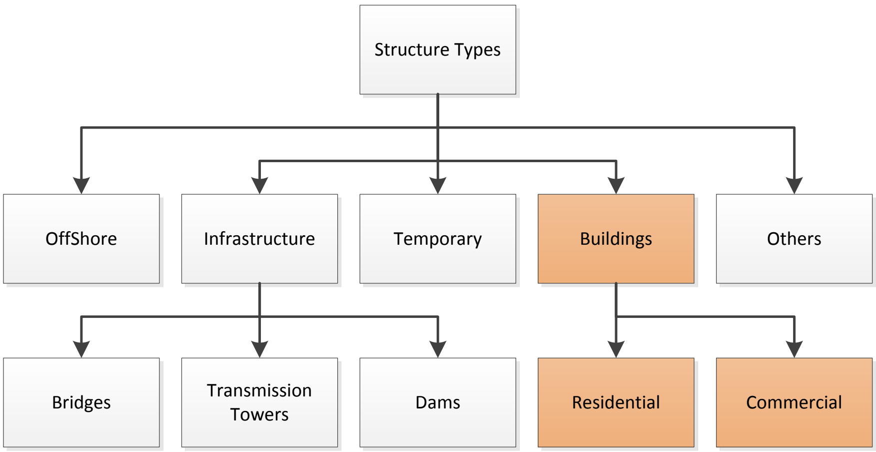

FIG. 1: Structural Sectors

3.3 Structural Systems

The classification of structural systems is critical in determining what information is needed to classify a structure (e.g. visual, behavior, function) and what subsystem elements formulate the composition of a given structure. In order to decompose the structural system to its basic elements, a determination of what each structural system consisted of was first organized. It was formulated that there are six primary material system types for buildings excluding any propriety material systems and other materials not heavily utilized in building applications (e.g. aluminum) for the primary and secondary systems. These six structural system classifications and subsystems are:

- Cast-in Place Concrete, further subdivided into Reinforced Concrete, Post-Tensioned Concrete and Fiber Reinforced Polymer (FRP) Concrete
- Precast Concrete
- Reinforced and Unreinforced Masonry
- Timber and Wood
- Structural Steel
- Foundations.

Composite structures, those with two or more materials in their makeup, were also considered but a separate category was not created at the material level for these as they are simply a mix of the primary material system types. Instead they are listed in the system decomposition as appropriate, primarily lateral systems can be affected by these interactions in terms of scale, as compared to gravity and foundation systems. To accurately represent each structural system, a hierarchical outline which each system would follow was created. The resulting representation that each system follows is shown in Fig. 2.

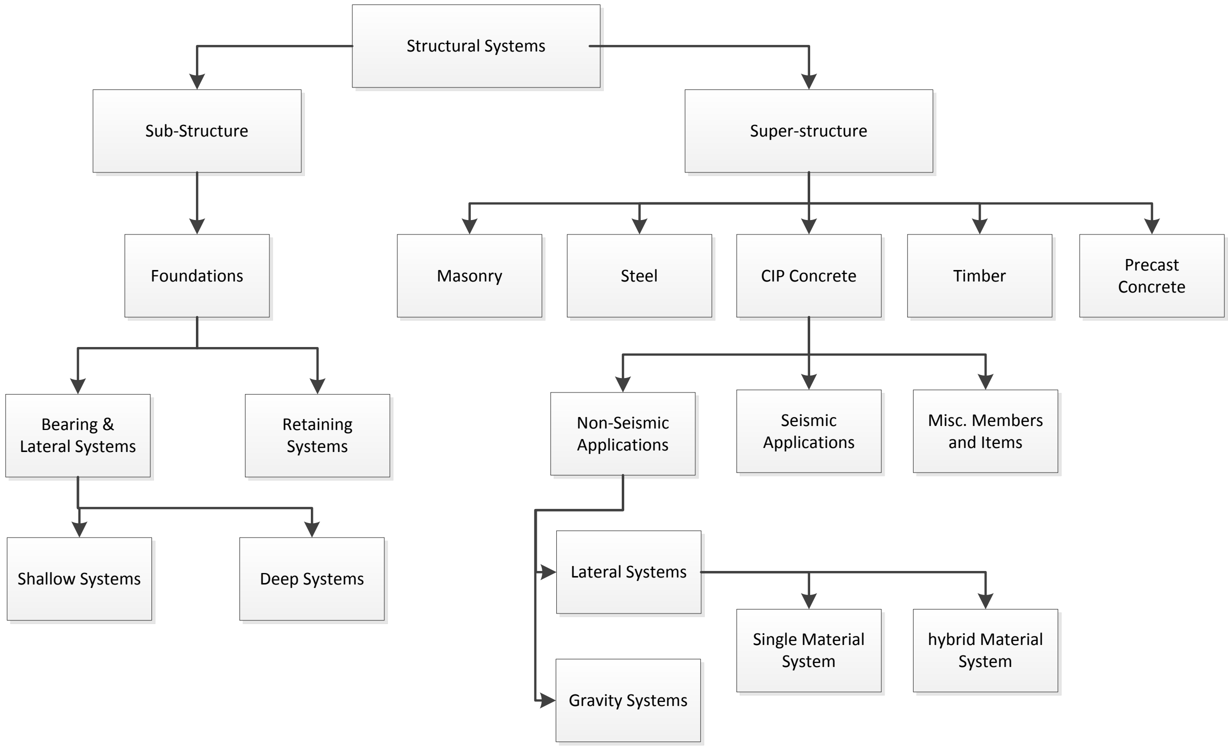

FIG. 2: Structural System Aggregation Template

First the hierarchy decomposes the structures into sub-structures and super-structures which are the two dominate classes. Next, these sub-structures are further decomposed into the system types described earlier. Each system is then decomposed into three sectors: Non-seismic Applications, Seismic Applications, and miscellaneous members and items. Miscellaneous members and items include connections, anchor plates, and other small elements of the primary system that an engineer, either engineer of record or detailer, must design. The next decomposition of this hierarchy is dividing the Applications down in gravity systems and lateral systems as these are the primary separations used by industry. A decomposition of lateral and gravity having seismic attributes was not considered due to the number of systems only feasible and specially conforming to seismic application (they are almost never considered for non-seismic conditions). Lastly is the decomposition of the lateral systems into two final classes: single material where the primary material is the only material used, and mixed/hybrid material where a combination of material and systems are employed. An example of the hybrid system is wood shear walls with steel strap bracing.

Having completed this hierarchy template with the systems' information, a visual comparison was carried out to see if and how structural systems at a material level exhibit similar traits. When comparing the different configurations and assemblies, it was observed that there was significant overlap between materials. This was especially true for gravity systems such as concrete, steel and masonry as they can often interchange the same floor types (e.g. one way slab, beams and girders, or two way slabs). Not surprisingly the material sub-types within reinforced concrete are nearly identical. An example is with the CIP and pre-tensioned materials, here they nearly all have the same possible gravity subsystem configurations but the differences are in the type of reinforcing (e.g. mild bar steel vs. high strength tendons).

Lateral systems provided less commonality due to limitations and the behavior of the configurations relative to all materials. An example of this is that timber and steel share bracing configurations yet timber moment frames are not utilized heavily as compared to steel. The hierarchy showed, at a system level, that ISPM's generalized process map undertaken can be developed such that it is non-material specific from this perspective. This is true as long as processes do not develop too deeply into what the tasks are defining from a particular material or configuration standpoint.

3.4 Structural Assemblies, Elements and Components

While the structural systems are classified by material, then load resisting capabilities, and subsystem assemblies, further organization is needed at a more detailed level to understand information relationships. As a result, the systems and subsystems were decomposed down into assemblies, elements and components. Further decomposition of the subsystems (lateral, gravity and foundation) resulted in the assembly class being generated. Assemblies are made up of many components and elements but behave in a different manner as a whole than the elements acting as if alone. Examples of structural assemblies include a wood bearing walls (elements present are beams, columns, and plates) or a stick/unitized facade (elements include glazing and mullion acting as slabs and beams). The assembly is further decomposed into two primary classes: the elements (members) and the components. The lower levels, below the element and component classes, have special terminology and requirements that need to be represented in a given schema structure that relates to the information discussed later. The hierarchy aggregation/decomposition described here is illustrated in Fig. 3.

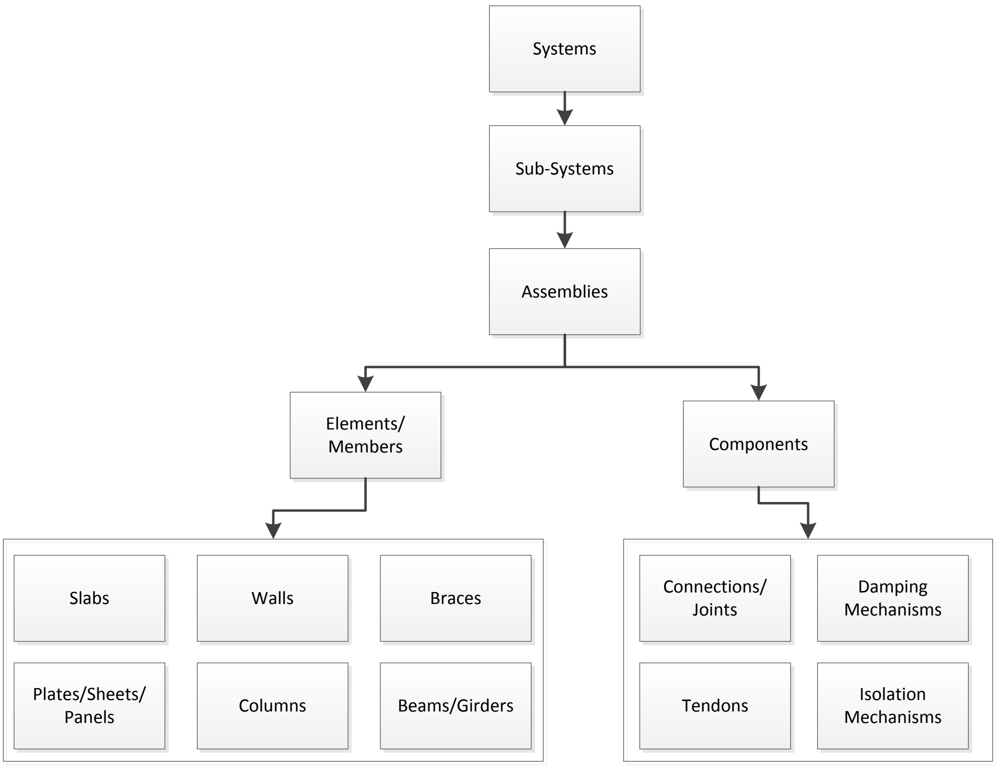

FIG. 3: Assembly to Element Hierarchy Breakdown

The element level is the level at which modeling software traditionally defines information. This extends that each individual element has unique characteristics when looking at analysis and design, manufacturing, construction, and operation throughout the system as a whole. This length of usage across the lifecycle and how behavior and usage of information can change supports why the broader hierarchy was developed. All elements can be considered to be part of larger assemblies that have their own unique meanings, characteristics, and behaviors separate from the element level but without elements they cannot function. Assemblies have a multitude of shapes, configurations, and functions whether they are for engineering, construction, and/or aesthetic reasons. The component side is similar to elements in that they are basic parts of the structure; however, they differ in that they do not fit within the basic structural elements defined by software and ATC 75. Examples of components are dampers, isolators and open web joists which are all unique and made up of many parts but practically are modeled as one equivalent representation such as a modified beam.

Looking more closely at the elements, we can see that most of the basic elements such as beam, column, brace and slab for example are defined in all undertaken modeling approaches but often in different manners (Gayer 2009; ATC 2009). ATC 75 does not distinguish between a non-composite and a composite beam or a tapered and a non-tapered column for example, relating to IFC. The developed hierarchy samples within the element class here are at more comprehensive depth than what was previously defined in ATC 75. This hierarchy takes into account unique characteristics to each material and type, as shown in Fig. 4. While there could be an infinite amount of configurations and combinations of how to break these elements down, the approach taken focuses on the primary elements and classes. The hierarchy structure can impose a specific material characteristic as often as it's needed at this lowest element level. The configurations for the elements generated for the preliminary study, however, provide only a representative sample to show that better definitions are still needed. This hierarchy can be used as a reference point when developing current schemas to ensure the representation of each element and component fit within the larger picture, particularly when looking at the same element but for different materials.

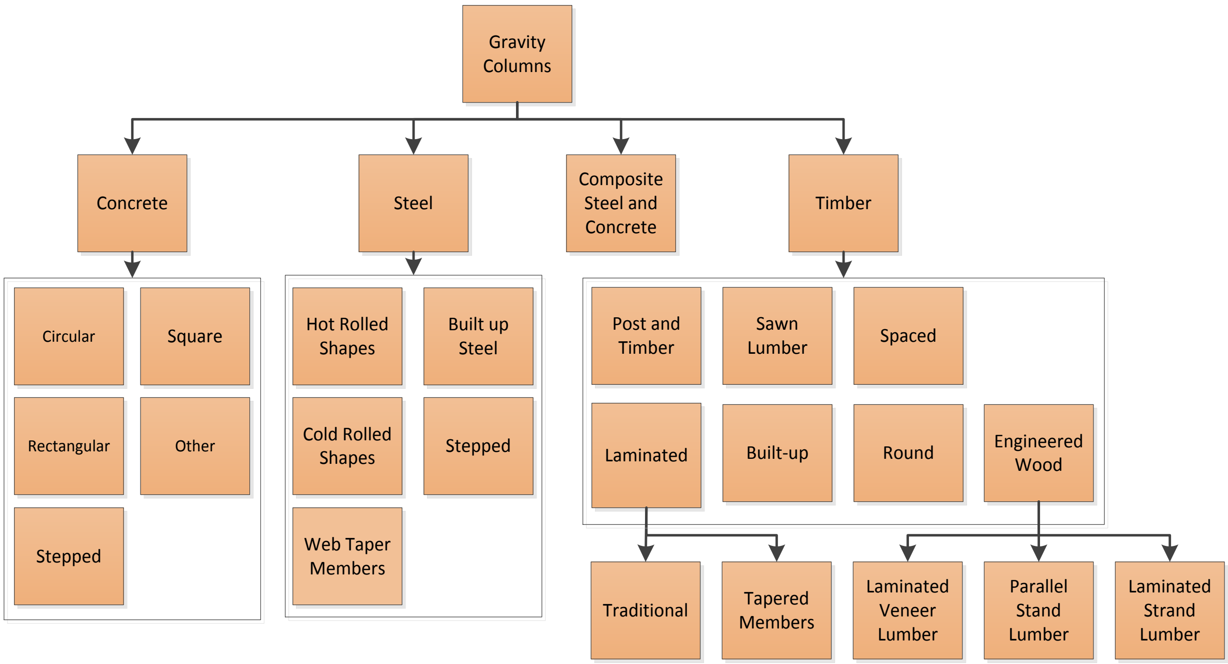

FIG. 4: Material Classes Mapped to a Column Element

Components, as mentioned, fall outside these traditional element classes and often are not modeled analytically in complex geometrical detail. Instead, they are given equivalent representations in the models. However, these equivalent representations need proper structuring to delineate the differences between them and others while holding all necessary component information. Decompositions under components are vast and as such were not all fully explored. Yet the common traits which indicate a component are as follows: (1) while components can be decomposed into subparts, this level is of the only concern to an engineer, (2) often but not always the components are not designed by the engineer of record, and (3) many of these can be proprietary in nature.

4. MODELING STRUCTURAL INFORMATION

The systems, assemblies, elements, and components discussed previously can exist visually in the software but without information cannot be used or truly defined. Knowing how previous studies had classified and formulated the information and how this vast amount of structural domain related information is generated, used, and exchanged within the lifecycle; an information type hierarchy aggregation was developed. In most cases it was observed that structural information is similar for each system and element. Yet often the notation, value, and meanings are different depending on the system material type. The information which tends to vary between systems is the relationship between objects, geometry (e.g. parametric properties) and behavior rules.

The information architecture produced takes a more streamlined approach and uses broad terminology such that it can be applied to other types of structures as needed in future implementation and expansion. The first level of the decomposition breaks the information down into four classes: product information, process control, feedback, and constraint. The second level aggregates theses four types into different primary classes; the number of these classes varies but all are at the same level.

The hierarchy of the two levels is depicted in Fig. 5. For space reasons, all classes at the same level are represented by a grouping box. Below the second level of the hierarchy (boxed in Fig. 5) there can be several more decompositions. These are more commonly grouped by attribute or a certain characteristic. To further clarify the hierarchy figures, solid arrows indicate a “is-a” hierarchy while a lined arrow indicates a “has” relationship. The sub-sections to follow present the most common occurrences recorded in the study. It is not necessary however, for some of the classes represented in Fig. 5 to be further decomposed before the exact (specific) values, attributes, and properties for a particular element and/or component are listed.

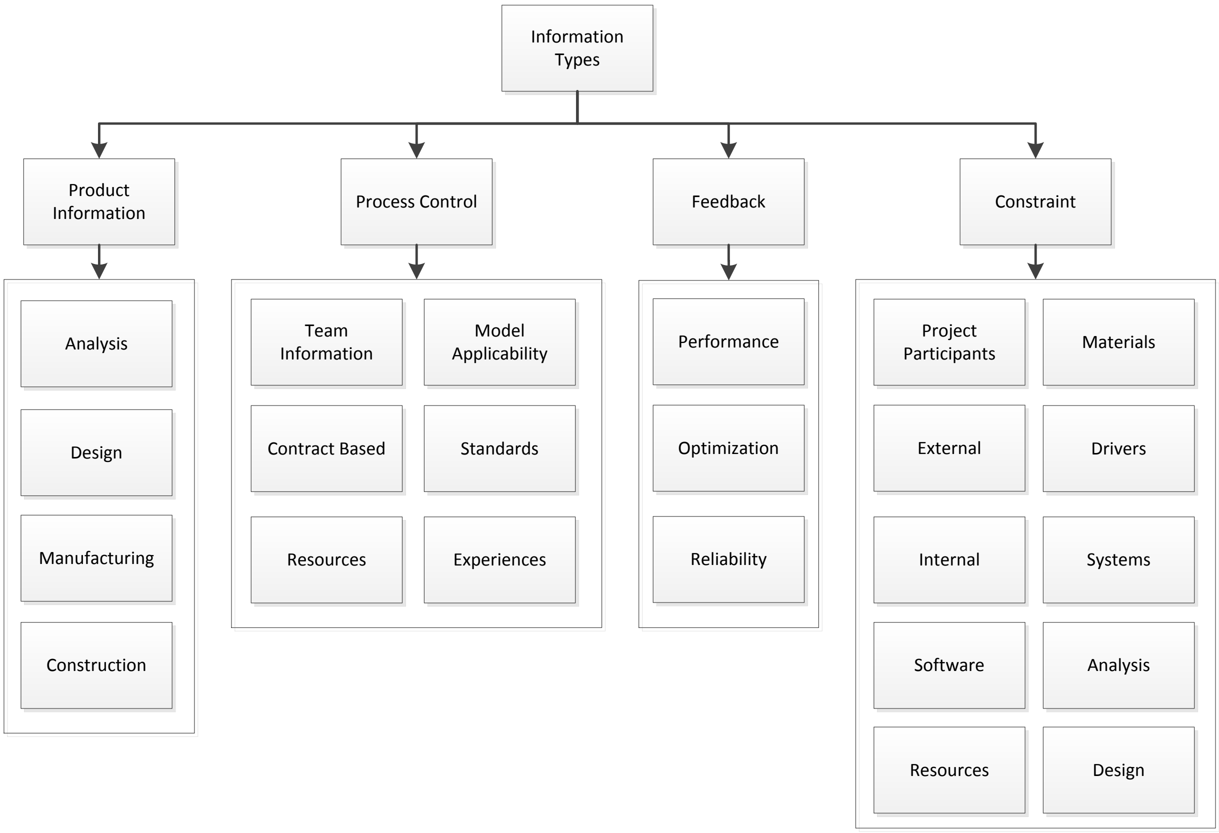

FIG. 5: Level 1 and 2 of the information hierarchy

4.1 Product Class

Product information is the information that associates the product to the project's systems, assemblies, elements and components while it focuses on their various functions as the lifecycle changes. Essentially this class of information defines the structure from various viewpoints within the structural domain. There were four primary decompositions related to either the behavior or phase the system, element, or component is in. The four classes are:

- Analysis: information needed and generated for analysis
- Design: information specifically for design
- Manufacturing: information about the fabrication process
- Construction: information regarding the construction phase

Looking at these four classes at the third level as represented in Fig. 6 there are multiple sub-classes which further defines each. Design information at the third level is composed of the following:

- Geometric information: defines the element or component shape
- Design documentation: used to represent the results visually (fonts, colors, ledgers)
- Calculations: indicate checks performed (e.g. resulting capacities and deflections for various conditions)
- Functional role: defines the element's role,
- Equipment specifications: document manufacturers' conditions
- Limit states: govern the design.

Moving clockwise in Fig. 6 the analysis class comes next. At the third level there are nodes that define the locations where elements meet, element types are the basic discretization of the actual representation into numerical space (e.g. lines, plates, and solids), boundary conditions that define the boundaries' behavior reactions, loads that are the applied forces and displacements a structure sees, results that are the analytical and numerical solutions, and external input that can be raw collected data that defines e.g. a particular reaction. Loads are decomposed down into a fourth level which contains: combinations that tell how loads are combined for scenarios, durations that document the length of time the load is acting, directions that state spatial paths, and types that define what the loads are.

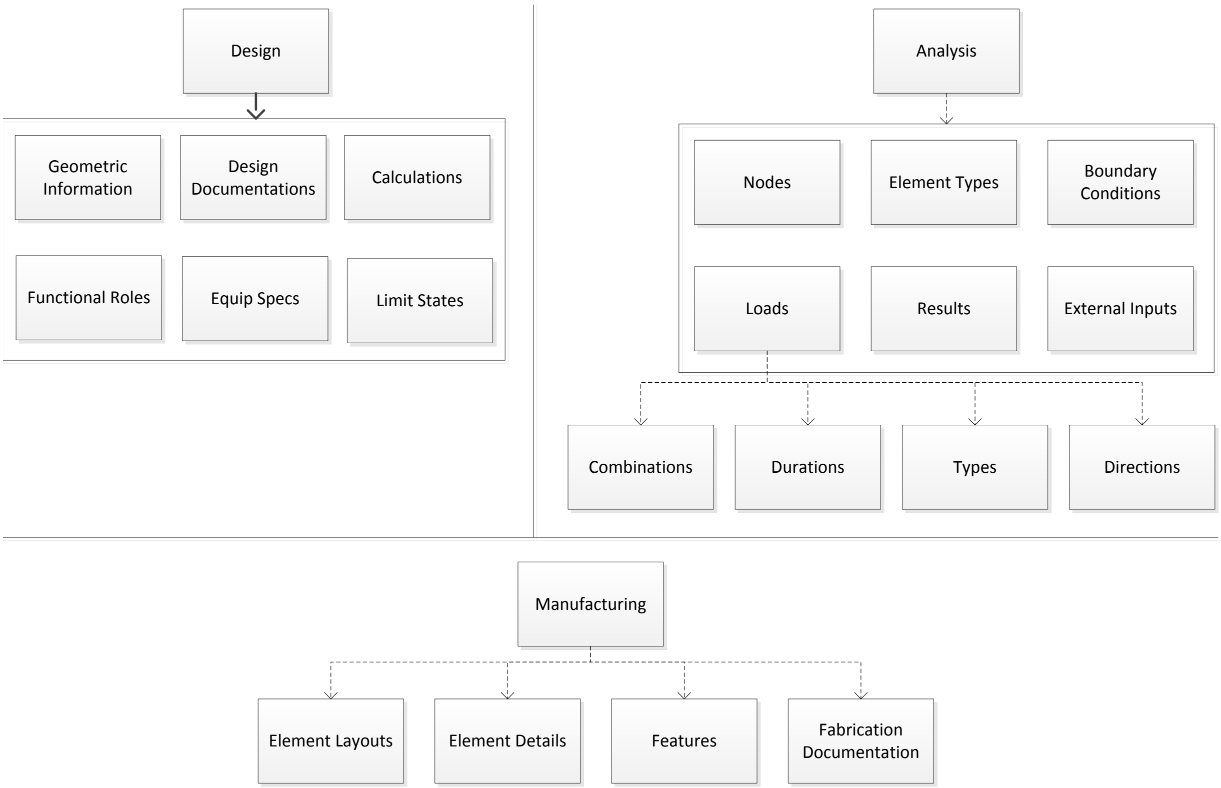

FIG. 6: Decomposition of Product Information Class

Construction class was not expanded to the third level; however, this would classify information on construction aspects such as cost, schedule, and time to name a few. The last product class is the manufacturing class that focuses on the fabrication phase where the items are produced either by hand or machine but need to be modeled nevertheless. Manufacturing's third level is broken into four classes: element layout that describes the layout of a part in raw material, element details which define the refined characteristics unique to each part (e.g. copes, bends, cuts, voids), fabrication documentation that is visual to represent the results (fonts, colors, ledgers), and features that are unique aspects to the element/component (e.g. finishes, textures, colors).

4.2 Process Control Class

Process control information sets and controls the sequence of an activity or set of activities. Sanvido et al. (1995) observed that this information is not typically looked at or divided in detail within the AEC domain. This still holds true today in structural firms for the most part. That said, there are noticeable trends present in the collected data. Six classes were decomposed in process control; they are (Fig. 7):

- Team information: describes the team roles on the project
- Contract based: delineates the responsibilities
- Resources: items that can be used on/in a project
- Model applicability: focuses on techniques used
- Standards: the controls governing design
- Experience: can limit or enhance the ability to do a task.

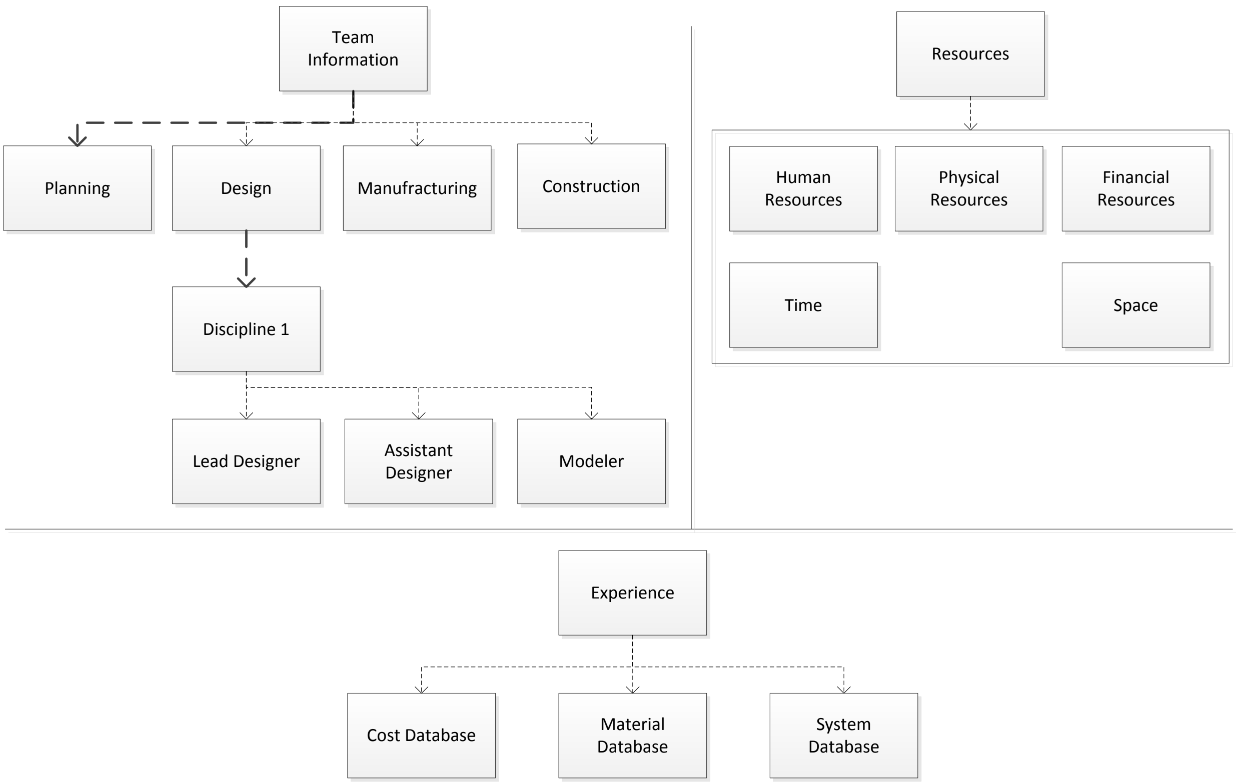

FIG. 7: Decomposition of Process Control Information Class

The first decomposition within the process control class is team information. Team information consists of a third, fourth, and fifth level as represented in Fig. 7. The third level is composed of planning that contains planning team characteristics. These can be used to determine the quality and acceptability of the produced work. Following planning is design, manufacturing, and construction that all focus on the information about the teams in the represented phases. Level four breaks each of these phases by discipline; then the fifth level breaks each discipline down by individual incorporating controls to be monitored (e.g. responsibilities and duties).

Resource information describes information that lies ready for use or can be drawn upon for aid when the need arises. The resultant decomposition contains six sub-classes. For each of the sub-classes they can be identified as either preexisting for the project or available information new to the project. This identification can be done as a simple identification attribute.

These sub-classes were not heavily explored beyond their listing in Fig. 7. As a result the descriptions of each are still preliminary.

- Human Resources: the amount of main power to conduct tasks
- Physical Resources: how much material there is that can be used
- Financial Resources: the allotment of money to do a task
- Time: how much allotment it will take to do something or is available to do something
- Space: this can refer to computational space or physical space to conduct a task or store items

Experience is the last control, here experience information is defined as background knowledge from previous projects and/or education the firm and individuals have acquired over time. The decomposition of experience results in three sub-classes. The first is a cost database that contains the actual cost data for projects types (e.g. stadiums, office, classroom, etc.) performed previously. Material database contains the materials and their associated properties for a particular project and lastly is the system database that contains systems solutions. All databases here can be a starting baseline for driving the design and/or construction.

4.3 Feedback Class

Feedback information relates to the control and decision-making characteristics within the project lifecycle. Often this information is generated when comparisons are made or at decision-making points on how to proceed to the next step. There are three classes decomposed within Feedback. They are (Fig. 8):

- Performance: looks at the resultants against the postulated plan,
- Optimization: looks at the efficiency of a system, assembly or element
- Reliability: looks at the behavior performance (owner and self-defined) viewpoint.

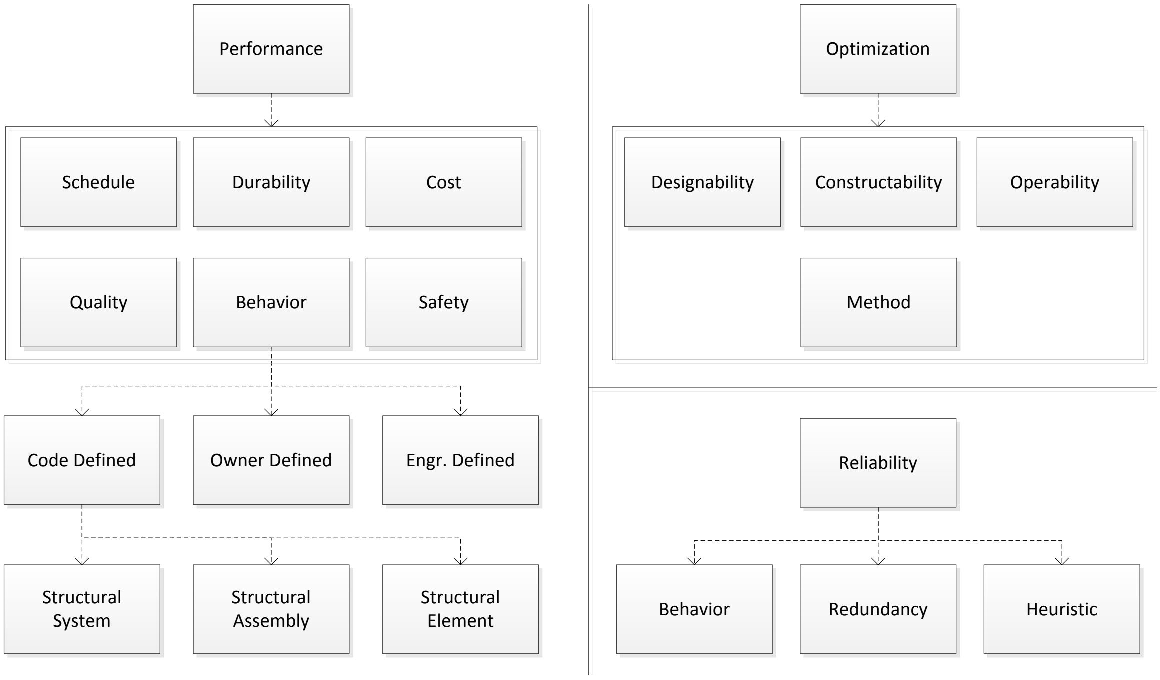

FIG. 8: Decomposition of Feedback Information Class

Performance feedback is decomposed to a third, fourth, and fifth level. At the third level there is schedule that tracks performance against time, durability of the design against particular factors, cost that associates the dollar value to performance, quality that can be on many aspects (e.g. material, labor, design etc.), behavior of the structure against benchmarks, and safety in how the structure will protect the occupants. At the fourth level only the behavior is decomposed, its sub-classes are: code defined that are those standards dictated by governing bodies, owner defined that are specific effects the owner specified, and engineer defined that the designer mandates personally. The fifth level shown in Fig. 8 only presents the decomposition for code defined, yet owner and engineer defined follow the same decomposition. This level takes the classes and groups them by system, assembly or element as each will have different performance attributes at their level.

Optimization feedback is similarly structured but only to the third level. Here, there are the designability, constructability, operability, and method classes. Designability looks at the design process information that considers the design method particulars. Constructability describes when the design is optimization based on construction to keep infield issues to a minimum. Operability is similarly focused, in this case on operations. Method classes are different types of quantitative and qualitative means (e.g. linear, multi-criteria, AHP) for determining the most efficient result. Reliability is decomposed into behavior, redundancy, and heuristic. Behavior is the repetition of the same results and probability of that behavior happening. Redundancy is the built in safety factor on expected to occurrences. Lastly, heuristic is simply best practices to ensure reliability and have no official standard definitions.

4.4 Constraint Class

Constraint information defines the boundaries and limitations of different functional abilities of situations and methods. These can and do affect the project outcomes. Depending on one's perspective, the constraint class is the most critical in successfully fulfilling design and construction requirements. Ten primary classes at the second level were identified; they are (Fig. 9):

- Project participant: the abilities and limitations of the project team members
- External: various disciplines which constraint tasks
- Internal: from within the structural domain
- Software: emphasizing the limitations and capabilities to perform computations
- Resource: limiting the selection of various efforts
- Material: the material limitations to do a purpose
- Drivers: the goals and scope the design must meet
- System: the structural system ability limitations
- Analysis: methods that have inherent limitations (e.g. complexity and speed)
- Design: certain methods imposing boundaries for efficient design based on limit states and background formulations.

A decomposition of the classes was possible due to the broadness that they enveloped. Many of the names to these sub-classes are reflected in the other sections of this paper. However, the classes used here have different meanings and functions at the class and attribute level. The first class to be taken to the third level is the material constraint. Within that class there are six sub-classes, represented in Fig. 9. Cost will constraint an issue based on a monetary value, availability bounds the material by if it can be acquired, quality is the paramount value (perceived) and the lack of defects, and aesthetic constrains materials based on visual and sensory perceptions. These constraints are often applied in other disciplines, too (e.g. mechanical, plumbing and site). The two structural specific constraints (a concern to the engineering team) are (1) Capacity that focuses on what a material could carry before exhibiting unwanted behavior, and (2) Efficiency where the material is not optimized to resist the loads.

The analysis class focuses on the design phases where a structure is analyzed to determine the behavior due to defined conditions. Limitations is the first sub-class comprising constraints that a particular method is valid for. Building on this, is the complexity class that encompasses the capability of computation ability (simple to advance). The type class focuses on the iteration and solution method limitations. Information needed is the last analysis sub-class; here are the attributes which must be present for an analysis to be performed. The counter point to analysis is design; as such the design constraint class has the same decomposition as analysis with only slightly different attribute definitions. Limitation is the first subclass that constrains what a particular design method is capable of doing. Complexity encompasses the computational capabilities (simple to advanced) in constraining the design options. The design type class focuses on the limitations of the design and solution methods in terms of appropriateness. Lastly is the Information Needed class, here are the attributes which must be present for a design to be performed.

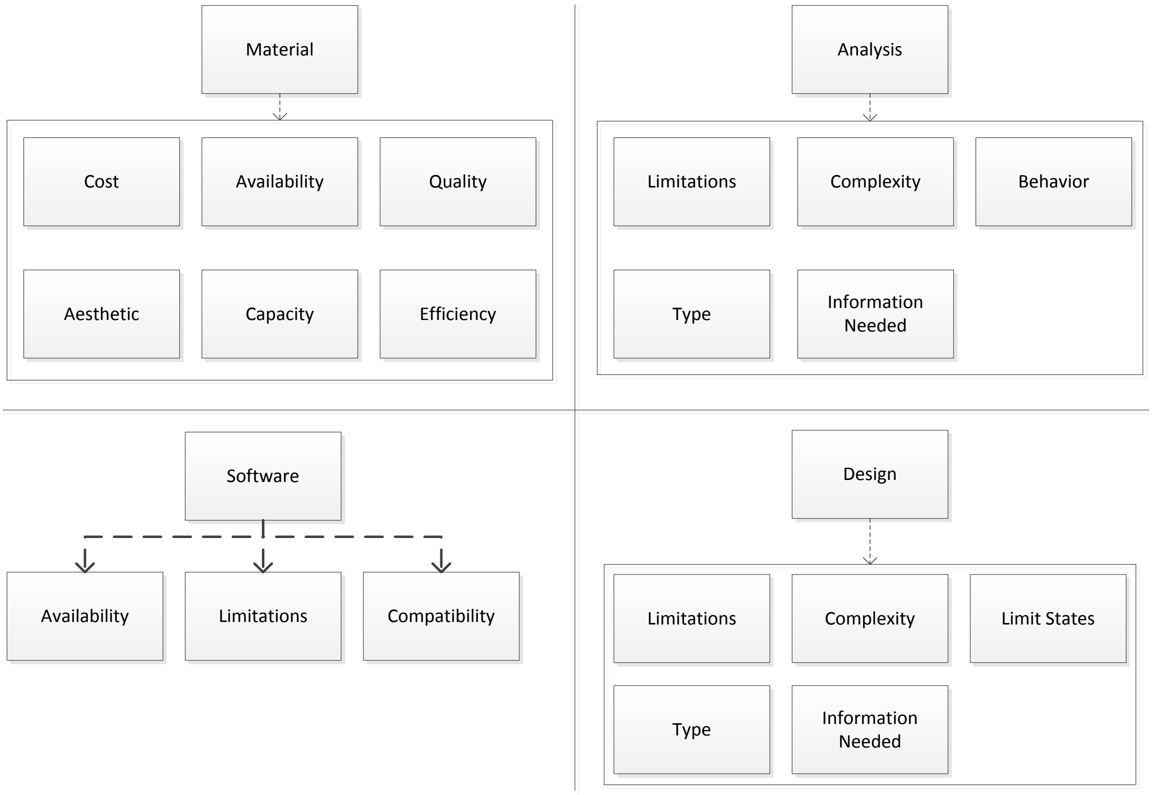

FIG. 9: Constraint Classes: Material, Analysis, Software, and Design Tree Hierarchies

Proceeding clockwise in Fig. 9, the next class that is decomposed is Software. It looks specifically at the computational software and hardware aspects. The Availability sub-class can help automated systems figure out what software to utilize. The Compatibility sub-class looks at software to keep information flow seamless, which can often be a major constraint. Limitations is the last sub-class constraint that deals with the software package as a whole with regard to its overall limitations (e.g. model size, element types, numerical solvers, etc.).

The last two classes decomposed in this study under the Constraint class are the Geometric and System sub-classes, shown in Fig. 10. The Geometric class consists of an aggregation of five different sub-classes, they are:

- Configurations: define global member arrangements in the system and how these can limit selection
- Sections: define the cross section of a member
- Size: defines a limiting factor based on lengths or availability of certain types of structures
- Spatial: is the nodal constraints which formulate the shape boundaries
- Location: defines the members within a space at the element/component level.

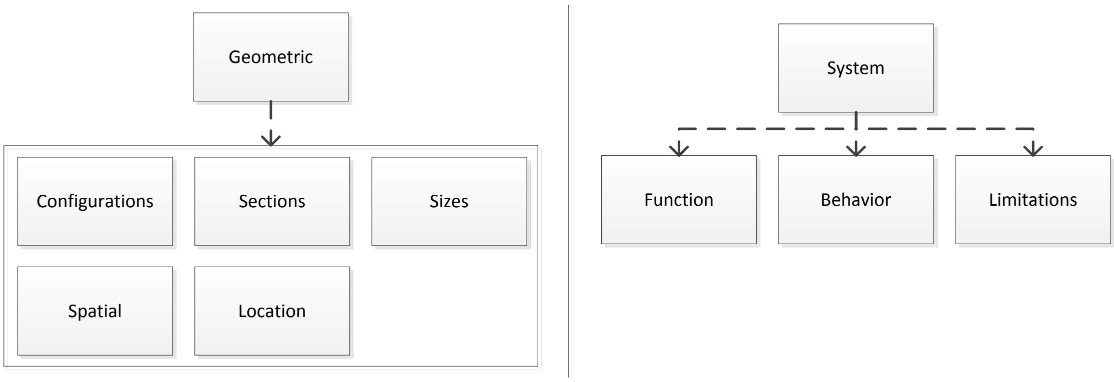

FIG. 10: Constraint Classes: Geometric and System Tree Hierarchies

The System constraint sub-class is decomposed only to the third level into three sub-classes:

- Function is the constraint the system is capable of doing or not doing
- Behavior is the reaction to loading the system exhibits and defines unwanted behavior to avoid
- Limitations lumps the other system constraints that narrow the ability to perform.

5. PROPOSED HIERARCHIES AGAINST THE CURRENT STATUS

Currently, there is a plethora of options for different schemas to be used in the BIM domain such as IFC, XML, SAT, and CIS/2 to name a few. IFC is now the widely accepted standard schema (NIBS 2007; Eastman et al. 2011). As mentioned previously, IFC initiatives are being conducted to develop the exchange attributes for each element and component type. This leads to a common question that may come up in schema discussions, why a new hierarchy and how does it relate to the current initiatives?

The architecture presented in the preceding chapters is formulated differently than current undertakings in hopes to provide a more robust result as the study continues. The primary reason for this effort, as it is presented, is to properly understand the relationships between attributes, commonalities between systems, and uses of the information. Resulting from this study to date are the novel hierarchies and architectures that have no intention to discredit IFC and other efforts. Instead, they are meant to help in the development of standards, custom interfaces and even users who generate models in how to add and use this metadata. This could be achievable as the hierarchies represent the different classes at more comprehensive depth than what was previously defined.

Knowing the commonalities between systems, allows for the ability to structure schemas with definitions that are more applicable and general to a series of systems. For example, when the GTI (2008) and ACI (2013) developed their schemas in IFC, the basis started from the beginning each time even though there was overlap. Furthermore, when MVDs are attempted to be converted from one case to another, significant rework potentially needs to be done based on the way the old MVD is constructed. If the proposed novel approach is used, the formulation would allow for a more generic understanding but still be more comprehensive. A simple example is if concrete’s cracked moment of inertia is formulated originally it would not be acceptable for steel yet if a generic inertia is formulated then its basis can be used for steel and concrete. From here, method specific information can be added to makeup the material level. Another prime example of being comprehensive yet general in an IFC formulation is the structural analysis IFC (cf. Lehtinen and Hietanen 2007). Here, terminology and attributes are extensive and cover a wide range of information classes. They do have some limiting factors in feedback and process control information. A reason for the completeness can be that structural analysis is inherently generalizable due to its applicability to any structural type and complexity.

Usage of information beyond the topic of interest for a material specific IFC initiative seems to be limited in current developments. It is believed that a more comprehensive knowledge of what information is used for and how it overlaps (e.g. cross section can define the shape but it also can be a constraint) is critical in writing each line of the IFC schema. The information hierarchies discussed in this paper form the basis of these classes that identifies what overlaps at the lower levels. This provides guidance for IFC developers to ensure coding is accurate for all information classes. Another advantage to having a defined hierarchy of information classes with correlating attributes is that it allows designers and modelers to understand the relationships of the metadata they are inserting into models.

6. DISCUSSION

The proposed hierarchy presented in this paper takes into account unique characteristics for each material and type. The performed study also indicated that the efforts to define the elements and components of the structure are presently too generic by the examined other initiatives as they only focus on the upper classes while the details of the lower decompositions are omitted. This work, if mapped against others, will show what is currently omitted. Additionally, the developed architecture can be a contribution towards works on current and future initiatives in the domain.

While there could be an infinite amount of configurations and combinations of how to break the system elements down, the approach taken focuses on the primary elements and classes. The classifications and structuring of the structural systems, elements and component compositions show that there is a significant overlap between systems and subsystems in what information is needed, obtained, used, and produced. Returning to Björk's (1989) thought of data classification, it was attempted to only classify information once yet in certain conditions this was not possible. Hence, indicator attributes may have to be used to distinguish cases. The configurations for the elements generated for the preliminary study, however, provide only a representative sample to show that better definitions are still needed. The hierarchy structure can impose a specific material characteristic as often it is only applicable to one material type at this level. This hierarchy can be used as a reference point when developing current schemas to ensure the representation of each element and component fits within the larger picture, particularly when looking at the same element but for different materials in how they differ.

Of all the classes listed, the functional attributes for each one are never generated at the same single point in time. Instead, the information builds, develops, and evolves over the project lifecycle. When inserting and representing specific attribute sets into each of these four primary classes and their sub-classes on three additional levels, there will, in certain instances, be overlap (same values). This overlap is due partially to that in one stage of the lifecycle information may be a product while later it is a constraint that must be satisfied. An example of this is the geometry of a structural element. The primary focus in the next phase of this work is to ensure definitions/structures of the attributes are unique. This implies that the callout on the data is represented once, yet its functional uses are known and incorporated into metadata properties allowing for cross linking. The specific functional attributes for each boxed class are not shown for brevity nor fully developed at this time due to the vastness of structural systems and components. However, when they are applied and expanded upon, they must satisfy the rules and properties listed previously. The classes developed are clear enough and tailored towards a structural nomenclature so that understanding this information hierarchy should have little resistance when applying detailed attributes.

7. CONCLUSIONS

The research presented here suggested a novel approach to modeling the structural systems and information classes as they apply to construction than that which has been previously defined. From this, relationships with various attributes across classes of information are important to understand. This is not just at a material or function level but at many different levels of a structure (e.g. system and element). It was the intent in having the different hierarchies allow for the understanding of relationships between system materials and configurations, information classes, and system breakdown. It is projected that as more automation in design with BIM advances, the more these relationships will be needed. The proposed hierarchies can help the next generation programmer understand these for implementation purposes. The presented information architecture has the potential to enhance initiative by spanning the entire building lifecycle of a project, looking at a more comprehensive format that is robust in cross-material and usage applicability.

Currently, this work is in the developmental stage where strict observations and successes are not yet easily accessible. However, this is not to say the work presented has limited merit, as the hierarchies defined in this study do define characteristics similar to other initiatives while adding to areas where work has been limited in terms of consideration. The next step in developing this structural information architecture is to expand each class and subclass to document the functional attributes of each. This documentation will lead the way in assisting the interoperability of data exchange structures in the structural engineering domain.

8. REFERENCES

American Concrete Institute (ACI) (2013). *Draft version of the Concrete Information Delivery Manual*, ACI, February 2013.

Anil E. B., Unal, G. and Kurc, O. (2012). Information requirements for design and detailing of reinforced concrete frames in multi-user environments. *J. Comput. Civ. Eng.*, 26(4), 465-477.

Applied Technology Council (ATC) (2008). *ATC-75 Development of IFCs for the Structural Domain Strategic Work Plan*, Applied Technology Council, Revision 1, June

Barak, R. Jeong, Y.-S. Sacks, R. and Eastman, C.M. (2009). Unique requirements of building information modeling for cast-in-places reinforced concrete, *J. Comput. Civ. Eng.,* 23(2), 64-74

Björk, B.C. (1989). Basic structure of a proposed building product model. *Computer-aided Design*, 21(2), pp 1-78.

Burt, B. A. (2009). BIM interoperability, the promise and the reality. *Structure Magazine*, December 2009, 19-21.

Dennis, A., Wixom, B.H., and Tegarden, D. (2009) *Systems Analysis & Design with UML Version 2.0 3rd Edition*, Wiley and Sons, Hoboken, N.J.,

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors 2nd Edition*, John Wiley & Sons., Hoboken, N.J.

Eastman, C.M., Jeong, Y.-S., Sacks, R., and Kaner, I (2010). Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards. *ASCE Journal of Computing in Civil Engineering*, 24(1), 25-34.

Ekholm, A. and Haggstrom, L. (2011). Building Classification for BIM – Reconsidering the Framework. *Proc. of the CIB W78-W102 2011: International Conference*, 26-28.

Emkin, L.Z. (1988). Computers in Structural engineering practice. *American Society of Mechanical Engineers*, 142, 37-43.

Ergen, E. and Akinci B. (2008). Formalization of the Flow of Component-Related Information in Precast Concrete Supply Chains. *Journal of Construction Engineering and Management*, 134(2), 112-121.

Gayer, A. W. (2009). BIM Power: Interoperability. *Structure Magazine*, October 2009, 14-16.

Georgia Technical Institute (GTI) (2008). Precast BIM Standard Project Building Information Modeling Standard for Precast Concrete Construction, http://dcom.arch.gatech.edu/pcibim/index.asp , Last accessed 02/22/11

Ikerd, W. (2010). Who's using BIM: Trends and drivers affecting structural engineers. *Structure Engineering and Design*, April 2010, 32-35.

Jacobi, J. (2007). Transforming the Delivery Process Using Building Information Modeling. *Structure Engineering and Design,* April 2007, 19-21.

Lee, G., Eastman, C. M., and Sacks, R. (2006). Eliciting information for product modeling using process mapping. *Data & Knowledge Engineering*, 62, 292-307.

Lehtinen, S. and Hietanen, J (2007). buildingSMART Structural Design to Structural Analysis model view definitions, *buildingSMART International*, http://www.buildingsmart-tech.org/specifications/ifc-view-definition/structural-analysis-view , Last accessed: March 28, 2013.

Luth, G.P. (1991). *Representation and reasoning for integrated structural design*. PhD. Dissertation, Dept. of Civil Engineering, Stanford Univ., Stanford, CA.

McGraw-Hill Construction (2008). *Building Information Modeling (BIM): Building Information Transforming Design and Construction to Achieve Greater Industry Productivity*. Smart Market Report, McGraw-Hill Construction, New York, NY, USA.

Messner, J. (2003). An Architecture for Knowledge Management in the AEC Industry. *Construction Research Congress*, March 19-21, Honolulu, HI, 1-8.

NIBS (2007). United States National Building Information Modeling Standard: Version 1 – Part 1: Overview, Principles, and Methodologies. National Institute of Building Sciences (NIBS), 2007

Powell, H.P. and Bhateja, R. (1988). Data Base Design for Computer-Integrated Structural Engineering. *Engineering with Computers*, 4, 135-143.

Sanvido, V. E., Anzola, G., Bennett, S., Cummings, D., Hanlon, E., Kuntz, K., Lynch, T., Messner, J., O'Connor, E., Potter, K., Riley, D., and Yoshigi, T. (1995). "A process based information architecture." *Technical Report No. 31*, Computer Integrated Construction Research Program, University Park, PA.

Sause, R. and Powell, H.P. (1990). A Design Process Model for Computer Integrated Structural Engineering. *Engineering with Computers*, 6, 129-143.

Solnosky, R. (2012). “Formulation of High-Level Hierarchical Aggregations for Building Structural Systems within BIM.” *College of Engineering Research Symposium*, University Park, Pa, 1-6.

Solnosky, R. (2013a). Current Benefits, Challenges, and the Future Potential for the Structural Discipline, *ASCE Structures Congress*, May 2-4, Pittsburgh, PA, 1-11.

Solnosky, R. L. (2013b). Integrated Structural Process Model: An inclusive non-material specific approach to determining the required tasks and information exchanges for structural BIM. *Ph.D. Dissertation, Department of Architectural Engineering*, The Pennsylvania State University, University Park, Pa.

Venugopal, M., Eastman, C.M., and Teizer, J. (2012a). An Ontological approach to Building Information Model Exchanges in the Precast/Pre-stressed Concrete Industry, *Construction Research Congress*, May 21-23, West Lafayette, IN,1114-1123.

Venugopal, M., Eastman, C.M., Sacks, R., and Teizer, J. (2012b). Semantics of model views for information exchanges using the industry foundation class schema, *Advanced Engineering Informatics*, 26(2), 411-428.