Improving the Usability of Standard Schemas

Jiemin Zhang, April Webster, Michael Lawrence, Madhav Nepal,
Rachel Pottinger, Sheryl Staub-French, Melanie Tory

Abstract

Due to the development of XML and other data models such as OWL and RDF, sharing data is an increasingly common task since these data models allow simple *syntactic* translation of data between applications. However, in order for data to be shared *semantically*, there must be a way to ensure that concepts are the same. One approach is to employ commonly used schemas — called *standard schemas* — which help guarantee that syntactically identical objects have semantically similar meanings. As a result of the spread of data sharing, there has been widespread adoption of standard schemas in a broad range of disciplines and for a wide variety of applications within a very short period of time. However, standard schemas are still in their infancy and have not yet matured or been thoroughly evaluated. It is imperative that the data management research community takes a closer look at how well these standard schemas have fared in real-world applications to identify not only their advantages, but also the operational challenges that real users face.

In this paper, we both examine the usability of standard schemas in a comparison that spans multiple disciplines, and describe our first step at resolving some of these issues in our Semantic Modeling System. We evaluate our Semantic Modeling System through a careful case study of the use of standard schemas in Architecture, Engineering, and Construction, which we conducted with domain experts. We discuss how our Semantic Modeling System can help the broader problem and also discuss a number of challenges that still remain.

1. Introduction

Information sharing is widely and increasingly used across a variety of domains and applications. Some of these domains have vocabulary that is very broad (e.g., the semantic web), and some of them have vocabulary that is very narrow (e.g., civil engineering). Regardless of the scope of the application, they all share something in common: in order for data from one source to be compatible with data in another source, they need to be semantically integrated. This means that the information in one source has to be understood by the users of another source. For example, in Architecture, Engineering and Construction (AEC), if one application stores information about "lintels" and another uses the word "beams", moving information between these two applications is going to require changing the data from one format to the other.

One solution to this problem is to use a *standard schema* and force applications to adhere to it. Such standard schemas range from formally designed specifications by industry consortiums or working groups, to published proprietary formats, to informally created and de-facto standard schemas circulated within smaller communities. These standard schemas allow for easy interoperability between applications that use the same standard schema. For example, STEP [1] allows users to describe design models for all manners of technical product data.

As long as users are content with working with the existing applications, and the existing applications all precisely adhere to the same standard schema, the standard schemas work perfectly and allow semantic translation. However, this does not allow users to do all of the things that they want to do; sometimes they want to create new applications, or do things that fall outside the bounds of existing applications. For example: in a CAD model it is easy to find the height of an individual wall. However, if users want to perform more complex analysis, such as determining the average height of all of the walls in a building or finding all of the intersections between walls in a building — and quantifying the dimensions and size thereof, then the users are left to work with the raw data. Even if this raw data adheres to a standard schema, it can be very complex to understand, primarily because standard schemas were designed to syntactically encode data, not to be particularly understandable or usable. Additionally, there may be more than one standard schema to choose from because many standard schemas exist for slightly different but overlapping domains. This process requires users, such as domain experts, to be able to understand multiple standard schemas and possibly compare them as well which exacerbates the problem. The frustration that domain practitioners have experienced with these problems has been well-documented in the evaluation of the use of standard schemas in practice in many individual domains (such as civil engineering [2], commerce [3, 4, 5, 6], finance [7, 8], biology [9, 10, 11] and legislation [12, 13, 14]). While these studies validate that standard schemas fall short in exactly the scenarios above, they are focused on *fixing the standard schema for the domain* rather than the more global, cross-domain problem of *how to deal with the fact that standard schemas will inevitably not be able to be used for all needs in all applications*. Using a case study in the AEC domain (Section 2) in which we collaborated with domain experts, we show how even in this very tightly constrained domain, the existing standard schemas fail to exhibit usability (Section 3), both when considered by themselves and when trying to choose one standard schema among many possible standard schemas. We then show our solution, a Semantic Modeling System (Section 4), and discuss how our system can be applied to even larger domains — such as commerce, finance, biology, and legislation — suffering from the same problems (Section 5).

Throughout this work, we concentrate on XML standard schemas because they are very common for exchanging data between applications due to their expressiveness and extensibility. The most important feature from a user's perspective is the unprecedented flexibility in describing and structuring information that XML provides. It allows users to define their own custom tags and structures and therefore a data representation that is tailored to their unique needs. Traditional data storage structures, such as relational databases, can be restrictive in their expressiveness because they enforce a relatively rigid organizational structure which may not be conducive for many types of information. The trade-off of XML's expressiveness is complexity: in order to store data in a more flexible and extensible way, XML represents information in a much more complex manner often indirectly using reference identifiers as we demonstrate in Section 2.3.1. Increased complexity leads to a significant reduction in usability. It is worth noting that while we focus on XML, new data models such as OWL and RDF suffer from these same flaws — our approach is focused on XML, but can very easily be applied to OWL, RDF, and other data models as well.

1.1. Our Contributions

In summary, the contributions of our team of AEC domain experts and computer scientists are as follows:

- We compare and evaluate a set of XML standard schemas and one relational standard schema for an application in the AEC domain.
- We evaluate the usability of standard schemas in the AEC domain through the implementation of an application using ifcXML [15], a well-established AEC XML standard schema.
- We identify two key usability areas: "complexity of standard schemas" and "comparison of standard schemas".
- We propose our Semantic Modeling System which uses a conceptual model (specified by a domain ontology) to ameliorate the usability concerns above.
- We verify that the usability concerns exist for standard schemas in other domains; our solution is a generic one and could easily be extended to research on standard schemas in other domains.
- We highlight additional areas for improvement of the usability of standard schemas.

2. Case Study: the ARTIFACT Project

The past few decades have witnessed an explosive growth in the volume of data that is being generated, collected and stored. Managing this information and extracting usable knowledge is a huge challenge faced by many industries. Often this problem is exacerbated by the diversity of information that must be integrated. In the Architecture, Engineering and Construction (AEC) industry, the various types of information associated with a large construction project such as scheduling data, 3D design data, meeting notes, and construction photos are typically stored disparately in different and incompatible formats specific to the applications in which the data was created. The ARTIFACT (Advanced Research, Techniques, and Informatics for Future Advantages in Construction Technology) project, is a collaborative endeavor between AEC practitioners and researchers in civil engineering and computer science. Its goal is to develop novel technology to support the task of extracting all manners of construction information from their native applications and integrating them to more effectively support AEC practitioners in making critical decisions for large construction projects.

To acquire a better understanding of the problem, consider the following example.

Example 1. By decreasing the intra-floor distance by 4 inches in a building design plan, a structural engineer can save the clients $50,000. While this appears to be a deceptively simple alteration, it is quite disruptive to the construction project as a whole. The effects of this change can potentially impact all other aspects of the project, such as duct work, electrical conduits, plumbing, schedule and, budget. Gauging the true impact of any change requires the structural engineer to have access to information about downstream activities and how the floor-to-floor distance in the structural design relates to and impacts the other components of the project. □

2.1. Design Data

We begin by describing our experience in developing a tool to support extracting domain-specific conditions which are important to construction practitioners. We extracted these conditions from a computerized building design plan (i.e., a building information model).

2.1.1. The Building Information Model (BIM)

A building information model is a parametric model of the elements that comprise a building: building components are represented by objects, which encapsulate the object's attributes (e.g., material properties), geometry, and spatial relationships to other building objects. A BIM is semantically richer than its predecessor, the entity-based model (EBM) [16]. In an EBM, building components are represented as entities that store geometric information but have no semantic information — i.e., what they represent or how the entities behave or interact [17]. For example, in an EBM the faces that comprise a wall do not know that they are connected together to form the wall. Obviously BIMs provide a significant improvement over EBMs and are unquestionably a step in the right direction for the future of construction modeling.

2.1.2. Why BIMs are Insufficient

While BIMs provide a great deal of data, other design knowledge that is essential to the various management tasks for which a construction practitioner is responsible (including cost estimation, selection of construction methods, scheduling, productivity analysis and project management) is not accessible.

Examples of these design conditions include the modularity, similarity, and layout of building components. This information is critical for constructing the walls, ducts, pipes, and columns in a building [16]. For example, knowing if and where building components penetrate other building components is important because penetrations typically require special construction procedures such as fire stopping, weather resistance, and the application of penetration seals.

Unfortunately, such conditions are not explicitly represented in a building design, and must be detected through manual inspection. This is a very inefficient, error-prone and frustrating endeavor which effectively limits the usability of BIMs. A greater amount of automation is needed to allow practitioners to query a building model in a way which shows them the features that they deem important. While some efforts have been made on this front, progress has been limited. The proposed solutions are not entirely usable: they have either focused on a simple and narrow set of conditions, required an unacceptable level of user input, allowed the user to query only a subset of conditions, and/or have not supported customization [16].

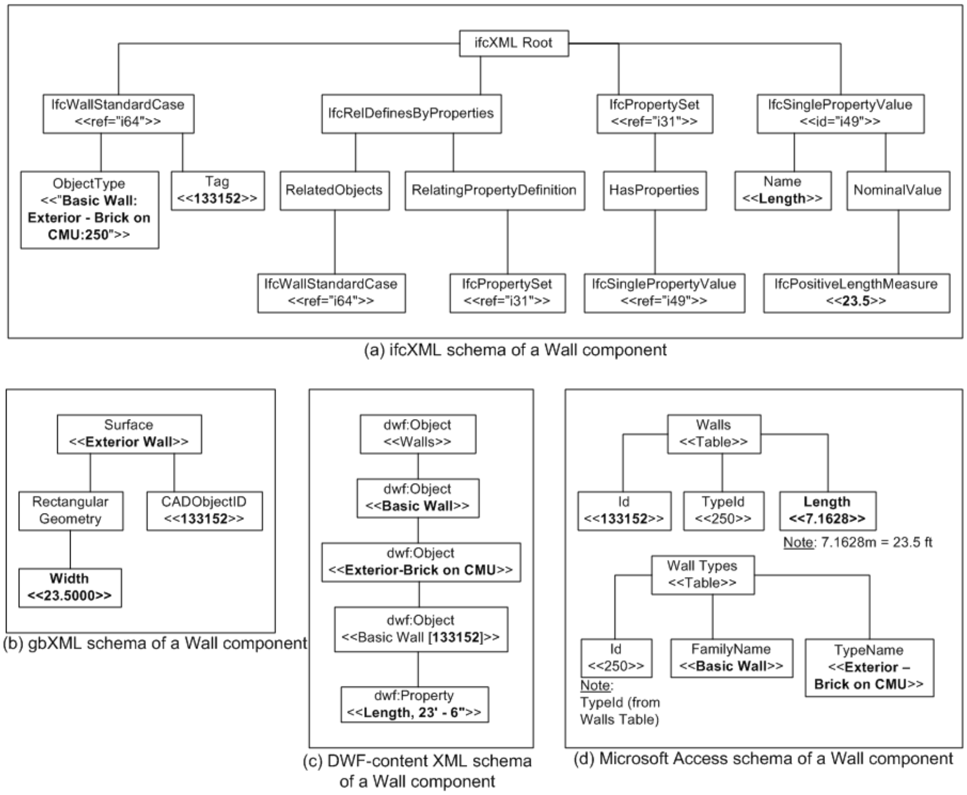
2.2. Extraction of Design Data: Multiple Standard Schemas

Designers of a building typically use software that is based on 3D CAD technology. Autodesk Revit Architecture (Revit), a CAD-based BIM application, is a common commercial building design software package. We chose Revit as an application to study since our domain experts informed us that it was representative of what was being used in practice and that it exported to the useful standard schemas in the domain. The data stored in Revit uses a proprietary format which cannot be read or written by non-Autodesk products. The inability to exchange data between two different applications is a common barrier to data extraction efforts as many of the applications and data storage solutions available today have proprietary data storage formats.

In order to extract information from Revit, the simplest option was to use one of its built-in export mechanisms. Revit provides several choices for exporting a Revit file (RVT) including image, XML-based, relational database and other CAD-based formats. We were able to eliminate a number of these formats from consideration as they did not improve our ability to access the information required by construction practitioners. Some of the export formats, such as image, are not machine-readable while others, such as DWG (DraWinG), require a license to use their read/write libraries. Based on these restrictions, we selected four of Revit's export formats as our candidate standard schemas: three XML-based standard schemas (DWF-content XML, gbXML, ifcXML) that are commonly used and supported throughout the AEC domain and one relational database export (in Microsoft Access). We provide a brief introduction to each of these standard schemas in the following section.

2.3. Candidate Standard Schemas

We introduce each candidate standard schema with a brief description. We informally gauge the relative complexity of each standard schema by the structure of the data necessary to represent a single wall of type "Exterior - Brick on


ifcXML Root
IfcWallStandardCase <<ref="i64">>
IfcRelDefinesByProperties
IfcPropertySet <<ref="i31">>
IfcSinglePropertyValue <<id="i49">>
ObjectType <<"Basic Wall: Exterior - Brick on CMU:250">>
Tag <<133152>>
RelatedObjects
RelatingPropertyDefinition
HasProperties
Name <<Length>>
NominalValue
IfcWallStandardCase <<ref="i64">>
IfcPropertySet <<ref="i31">>
IfcSinglePropertyValue <<ref="i49">>
IfcPositiveLengthMeasure <<23.5>>
Surface <<Exterior Wall>>
Rectangular Geometry
CADObjectID <<133152>>
Width <<23.5000>>
dwf:Object <<Walls>>
dwf:Object <<Basic Wall>>
dwf:Object <<Exterior-Brick on CMU>>
dwf:Object <<Basic Wall [133152]>>
dwf:Property <<Length, 23' - 6">>
Walls <<Table>>
Id <<133152>>
TypeId <<250>>
Length <<7.1628>>
Note: 7.1628m = 23.5 ft
Wall Types <<Table>>
Id <<250>>
Note: TypeId (from Walls Table)
FamilyName <<Basic Wall>>
TypeName <<Exterior – Brick on CMU>>

(a) ifcXML schema of a Wall component

(b) gbXML schema of a Wall component

(c) DWF-content XML schema of a Wall component

(d) Microsoft Access schema of a Wall component

Figure 1: A comparison of representations of a single wall and its length of 23.5 feet using 4 different standard schemas for AEC data.

CMU" and its length of 23.5 feet. We chose to use a wall for our illustration as it is one of the simplest and most common building elements in a building design. Furthermore, the details about walls and their interaction with other components are important to construction practitioners (e.g., wall-to-wall intersections which are described in Section 4.2.1). Figure 1 shows the data representing this wall for the four standard schemas described here.

2.3.1. ifcXML

IFC (Industry Foundation Class) is an open source object-oriented standard schema for the AEC industry that was developed by the International Alliance for Interoperability (IAI) [15]. The purpose of IFC is to facilitate the exchange of information used by AEC professionals during the building life-cycle. This includes planning the building, designing the building, constructing the building, and maintenance during the building's operation. Therefore, IFC represents not only the physical information that describes buildings, but also the information necessary to manage all of the tasks that comprise a building project including planning, cost estimation, scheduling and operation [18]. ifcXML is the XML version of IFC.

IFC is a very complex and content-rich standard with a correspondingly complex schema. Figure 1 (a) shows an ifcXML tree representing our example wall. The size of the tree necessary for such a small amount of information is indicative of ifcXML's complexity: four different length-three branches of ifcXML (fifteen nodes in total) are needed to represent this wall. Closer investigation reveals that a Wall component (such as the instance shown in this example) is indirectly linked to its properties through two different relationships using reference identifiers (i.e., the reference tags identified for some of the components in the schema, such as ref="i64" for the IfcWallStandardCase). For example, to determine the length of a wall one must first retrieve the set of properties contained in an IfcPropertySet element associated with the Wall through the IfcRelDefinesByProperties relationship. In our example Wall it is the IfcRelDefinesByProperties element with id 'i64' that links the IfcPropertySet element with id 'i31' to the Wall. This IfcPropertySet has a link to the IfcPropertySingleValue element with id 'i49;' this is the element that holds the actual length of 23.5 feet for the Wall. It is important to keep in mind that this is one of the simpler relationships in an ifcXML document.

2.3.2. gbXML

gbXML (Green Building XML) is another open source standard schema. The role of gbXML is to address the data representation needs of the green building design movement [2]. Because it was created with a more focused purpose, a RVT file exported to gbXML contains only a small subset of the original information. In particular, only information pertaining to building energy analysis [2] such as the components related to spaces and surfaces are represented: walls, windows, and doors are exported, but components such as columns, beams, slabs and ducts are not (as these are not deemed to be significant in analyzing a building's "green-ness".) The gbXML data for our example wall is shown in Figure 1 (b). It is evident that gbXML's wall schema is significantly simpler than ifcXML's wall schema, with properties such as length (referred to as width in gbXML) at the second level of the schema and directly associated with the component it describes.

2.3.3. DWF-content XML

DWF (Design Web Format) is an Autodesk file format for distributing design data created and stored in their products such as Autodesk Revit [20]. An exported DWF file is a compressed archive containing a number of files [21]. Among the set of files contained in a DWF archive is an XML file: content.xml (DWF-content XML), which contains the design information exported from the original RVT file. While containing most of the components available in Revit as well as a majority of the properties of these components, much of the information related to component relationships that are derived from the relative location of components is not represented. For example, wall intersections are not available in a DWF-content XML file. Figure 1 (c) shows the DWF-content

XML of our example wall. It has only a four nodes, and is significantly simpler than comparative representations of the same wall in other formats.

2.3.4. Relational Database Export

Revit provides users with the option of exporting a RVT file into a Microsoft Access relational database using ODBC. This export does not use a standard schema. However, it is one of the export options provided by Revit, so it satisfies the goal of facilitating data exchange between different systems [2]. Therefore, we consider it as well. The exported Access file is comprised of a set of tables, which represent all of the basic Revit components (e.g., walls, doors, floors, etc). For example, there is an Access table named "Wall" for Revit's basic component category "Wall." The columns of this table represent the properties provided by Revit for walls including Id, TypeId, Volume, Area, Length, TopOffset, BaseOffset, and so on. The data for the example wall (each table is drawn as a two level tree whose root is the table and leaves are the attributes) is shown in Figure 1 (d). As can be observed, it is fairly simple, requiring only two tables to represent our example wall.

2.3.5. Choosing a Standard Schema

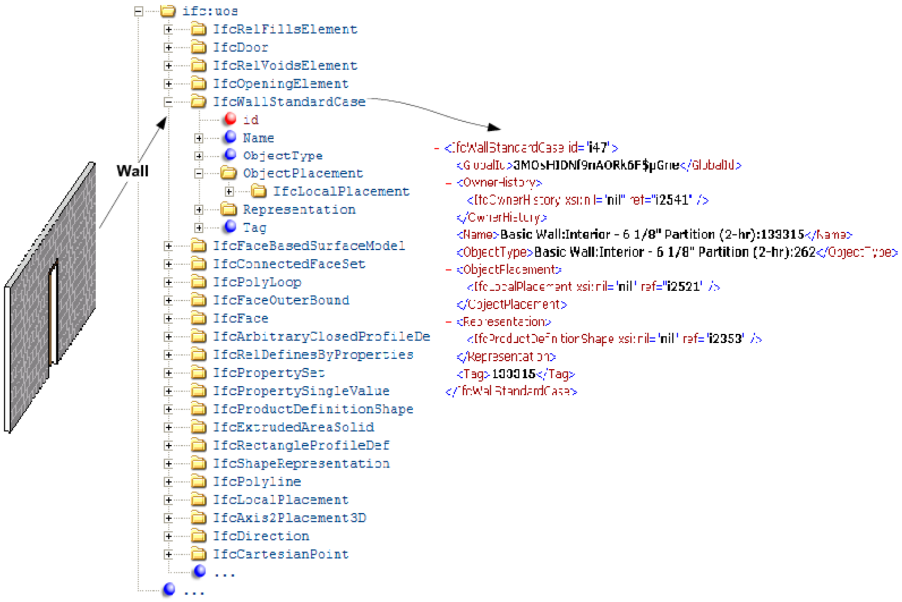
Our case study highlighted two constraints in addressing the problems, such as the one presented in Example 1 that AEC practitioners need to solve on a daily basis. First, these problems are not supported by existing software applications. Second, we ultimately must integrate the data in Revit with data from other applications such as scheduling data, financial data, and cost estimation data. The solution to these constraints was to export the data we needed to a common format — a standard schema. This required first understanding each standard schema and how it relates to the concepts in the domain, and ultimately then which standard schema to choose. Section 3 describes the specific problems that we encountered in this task that ultimately required us to build our Semantic Modeling System, as described in Section 4.

3. Usability Problems

The work we had to do to select the best candidate standard schema for our case study and then extract the knowledge required by construction domain experts highlighted several usability problems. We discuss these findings in the section that follows.

3.1. Complexity of Data Standard Schemas

The expressive power of ifcXML carries a hefty cost in increased schema complexity. This results in more complicated query expressions to represent the building design concepts that are important to domain practitioners. Consider the following example: Figure 1 shows the data for a Wall component in each of the four candidate standard schemas. Even for a single relatively simple building component, it is immediately evident that ifcXML (Figure 1 (a)) has a



```
- <IfcWallStandardCase id='i47'>
    <GlobalId>3MOsHIDNf9nAORk6F$pGne</GlobalId>
  - <OwnerHistory>
      <IfcOwnerHistory xsi:nil='nil' ref='i2541' />
    </OwnerHistory>
    <Name>Basic Wall:Interior - 6 1/8" Partition (2-hr):133315</Name>
    <ObjectType>Basic Wall:Interior - 6 1/8" Partition (2-hr):262</ObjectType>
  - <ObjectPlacement>
      <IfcLocalPlacement xsi:nil='nil' ref='i2521' />
    </ObjectPlacement>
  - <Representation>
      <IfcProductDefinitionShape xsi:nil='nil' ref='i2353' />
    </Representation>
    <Tag>133315</Tag>
  </IfcWallStandardCase>
```

(a) 3D Model (b) Hierarchical overview of ifcXML (c) ifcXML

Figure 2: A 3D wall component with the corresponding ifcXML representations.

much more complicated, indirect representation than the other three standard schemas.

There are four different paths of the ifcXML schema which must be traversed, each with a length of three, to determine that the length of the 'Exterior-Brick on CMU' wall with a reference (ref) of 'i64' is 23.5 feet. Most information that describes a wall is indirectly associated to it in this manner as is demonstrated in Figure 2; to query features from an ifcXML file requires analyzing how elements are linked with different properties and relationships.

The indirect method of relating components to their properties makes it difficult to understand ifcXML as one must manually track the reference identifiers to determine the path required to map a concept in ifcXML to a concept that a domain expert wants to represent. Additionally, it also leads to very large file sizes considering the scale of the building being represented. For example, the ifcXML file for a building design consisting of a single room (6 walls, 4 columns, 11 openings) is 1MB in size and for a simple two-level house (9 walls, 30 openings) is more than 10MB. Most construction projects involve buildings of significantly greater complexity and size — twenty storey buildings with hundreds of rooms, walls, columns, openings, etc; clearly an ifcXML file for such a building would be exceedingly large and difficult not only to navigate but to query as well. This scalability issue underscores the importance of having a solution that is more usable — the standard schema output is simply too unwieldy to be used.

3.2. Comparison of Data Standard Schemas

As the candidate standard schemas are all designed with the same general purpose, one would expect them to be redundant; they are not. While all describe the AEC domain, each standard schema has a slightly different flavor. This leads to differences in the coverage of content that each provides. For example, gbXML (Section 2.3.2) was designed for the green building design domain and only represents information relating to building spaces and surfaces. If a construction practitioner needed information about building components that fell outside this domain, another standard schema or combination of standard schemas would have to be consulted. In a sense, each of the candidate standard schemas provides a view of the data that is tailored to their particular building design niche.

In addition to representing different and overlapping sets of construction data, different standard schemas also represent the construction data they cover in different ways. For example, each standard schema has different representations of the notion of a "wall" including what attributes are represented and the way in which this information is structured, as was covered in greater depth in Figure 1 in Section 3.1.

Due to the these complications, comparing standard schemas is exceedingly difficult. Even our domain experts who are very familiar with the concepts that are important to the construction process and moderately familiar with the standard schemas required three full months to understand the relationships between the domain concepts and the standard schemas in sufficient depth. The civil engineering team members found this unacceptable as most construction professionals would not have the time, inclination, nor expertise to do the work themselves. Note that because the difficulty was in *understanding* the schema not *mapping* the schema, standard schema matching and ontology matching literature and ontology alignment literature (see [22, 23] for recent surveys) do not solve the problem.

We also encountered problems in determining a matching expression in the candidate standard schemas for the concepts identified by domain experts. For example, the shape of a slab (and therefore any relationships involving a slab and other building components), whether a component has a penetration, and the number of wall clippings are among the concepts that could not be represented in any of the evaluated standard schemas.

4. Semantic Modeling System

The tools that are currently available to support the usability of standard schemas (whether for XML or another data model) are either inadequate or nonexistent. Our Semantic Modeling System addresses this void by helping users to understand and compare competing standard schemas. It is comprised of two components: (1) a **conceptual model** of the shared knowledge in a domain — in our case, the AEC domain — which we encode in a domain ontology and, (2) the **mappings** from a standard schema to the domain ontology. The first component provides a common language for comparing the coverage provided by each candidate standard schema of the desired knowledge. The second component facilitates the automatic extraction of this knowledge from the underlying source data to guide us in the evaluation of the candidate standard schemas identified in Section 2.3. In this paper we describe both of these components at a fairly high level. Additional details regarding our conceptual model can be found in [16, 37] and our mappings in [36].

4.1. The Conceptual Model: a Common Language for Comparing Standard Schemas

Our conceptual model was developed jointly by the civil engineering and data management experts on our team. It is an ontology of features — the objects in the design that have meaning to domain experts — and the relationships between them. We designed an ontology by using Protégé [24] which captures knowledge based on the universal concepts of a building design; it is therefore independent of the various XML standard schemas used to represent building designs. Using Protégé also ensures that data can be easily exchanged between applications [25]. As the purpose of our ontology is to represent the design conditions that are critical to building construction practitioners, only those features and relationships that can contribute to the articulation of these conditions are represented. Each of the conditions that we chose to identify were identified by the domain experts on our team [16].

The ontology is comprised of three basic elements: features, relationships between features, and properties of each. Both the features and the relationships are represented as classes: Component, Opening, Intersection, Penetration, Design Uniformity, Spacing, and Alignment [16]. We refer to features and relationships more generally as concepts.

The Component class represents the standard building elements such as Walls, Columns, and Ducts; these elements constitute what are typically thought of as features. The six remaining classes describe feature relationships that may impact construction. Openings modify a Component by removing some part of it and optionally replacing it with another Component such as a Door or Window; an Opening impacts which construction methods can be used. An Intersection occurs when two building Components meet or interact with each other; Wall-Column Intersections, for example, may require additional time to be set up, and additional framing. Penetrations involve a building services Component such as a duct passing through another building Component like a wall. Penetrations require special care since they require different procedures to include adequate fire stopping and weather resistance [16].

Design Uniformity and Alignment are somewhat more abstract relationships and apply to a set of building Components. Uniformity represents consistency on a set of Components and can be characterized, for example, on the spacing between or the shape/size/location of Components. The more uniform a design, the easier it is to reuse components and construction methods, which ultimately speeds up construction and decreases cost. Alignment represents how the Components in a set are located with respect to some reference like a building plan gridline which be important in selecting which construction method to use, e.g., flying form tables [16].

As has been demonstrated, each of these six concepts (i.e., features or relationships) are important because they largely dictate the construction process. They are also some of the more common conditions that construction professionals look for in a building design plan. Having this domain ontology allowed us to easily understand the concepts that were necessary to the practitioners in a clear, precise representation. Without the ontology it would have been a process of trying to determine how to answer an arbitrary set of queries; while each individual query could have been answered, the potential for reuse of the common concepts — which are reified in the elements in the domain ontology — would have been lost.

4.1.1. Comparing Standard Schemas Using the Conceptual Model

After careful analysis of the candidate standard schemas using the domain ontology presented in the previous section (which is our conceptual model), the details of which are presented in [37], we selected ifcXML to export the Revit data since it provided the most complete representation of the concepts represented in the domain ontology. Table 1 summarizes the results of our evaluation. Fully supported concepts are those that are either explicitly represented by a standard schema or can be derived. A partially supported concept is one that is supported in some cases, but not all. For example, Intersection (see Section 4.2.1 is supported for straight walls, but not for curved walls because the location of curved walls cannot be determined; in cases such as these, we say that the concept is only partially supported [37].

As summarized in Table 1, of the 57 concepts under consideration, ifcXML can represent (either fully or partially) almost 80% of them whereas the MS Access export and DWF-content XML can each represent 53% and gbXML only 26%. Having the domain ontology allowed us to focus easily on which concepts were present in each standard schema and compare on a principled basis, rather than having to painstakingly determine if each query could be answered on each standard schema. This slower process would also have lost the ability to be sure that the standard schema would have access to potentially interesting but currently unspecified queries.

4.2. Mappings from a Standard Schema to the Conceptual Model: Automating the Knowledge Extraction Process

The process of extracting the knowledge required by domain experts is a cumbersome task. Once a standard schema — such as ifcXML — has been selected, it is necessary to formally describe the mappings from a concept or combination of concepts in a this schema to each building design concept in the conceptual model. This automates the knowledge extraction process and significantly improves the usability of the standard schema. For our case study,

	Level of Support		
Standard Schema	**Complete**	**Partial**	**None**
Microsoft Access	22	8	27
gbXML	6	9	42
DWF-based XML	22	8	27
ifcXML	33	12	12

Table 1: Comparison of civil engineering design standard schemas by the number of distinct construction practitioner domain concepts (from a set of 57 in our domain ontology) supported.

we created the mappings using XQuery [26], the standard query language for XML. If the format of our selected standard schema had been relational, we would have chosen SQL to specify the query mappings; the query language used is simply an artifact of the format of the selected standard schema.

We created a formal mapping from the selected XML standard schema — ifcXML — to each of the concepts and relationships expressed in our domain ontology. It is important to point out that we also informally specified the mappings from the other candidate standard schemas — gbXML, DWF-based XML and Microsoft Access — during the comparison step; although we chose to formally specify only the mappings from ifcXML to the ontology, we could easily extend this process to formalize the mappings from the other standard schemas to the ontology as well.

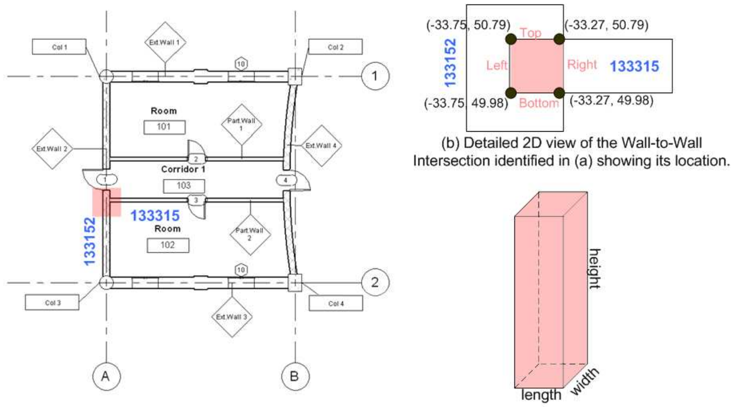
To illustrate some of the complexity of this work, we provide an in-depth description of the mappings from ifcXML to the domain ontology for two of the representative relationships in the ontology — Intersection and Spacing — in Sections 4.2.1 and 4.2.2. It is worth noting that while current research on schema matching and ontology alignment (see [22, 23] for recent surveys) could be used to find the initial correspondences between elements, the work in this section would still be necessary to discover the precise complex relationships between elements.

4.2.1. Intersection

An Intersection occurs when two building Components intersect as shown in Figure 3. The Intersection query returns more detailed information about the intersecting region: its location (i.e., the corner points of the region), dimensions (i.e., width, length, height), area and volume. Construction practitioners use this information in a number of different ways. For example, Wall-to-Column Intersections can require additional framing for movement joints and, Wall-to-Wall Intersections impact drywall construction costs [25]. Both examples require information above and beyond whether building components simply intersect or not.

4.2.2. Spacing

Spacing identifies the minimum or maximum distance between proximate (i.e., adjacent) features; this is shown in Figure 4 for the columns on the first



(a) A Wall-to-Wall Intersection between Walls 133152 and 133315 is highlighted in red in the building design floorplan.

(b) Detailed 2D view of the Wall-to-Wall Intersection identified in (a) showing its location.

(c) 3D view of the Wall-to-Wall Intersection identified in (a) showing its dimensions.

Figure 3: Example of a Wall-to-Wall Intersection and the details provided by the Intersection spatial query predicate

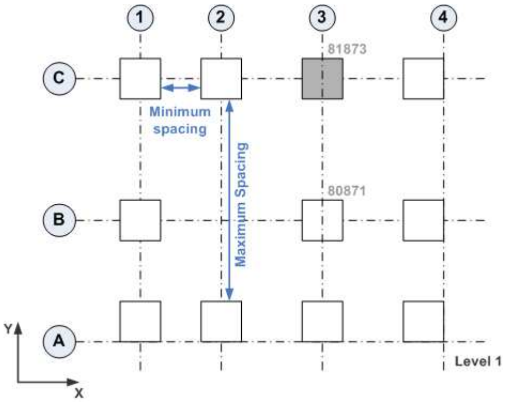


Figure 4: Spacing of on-grid columns on the first level of a building design plan

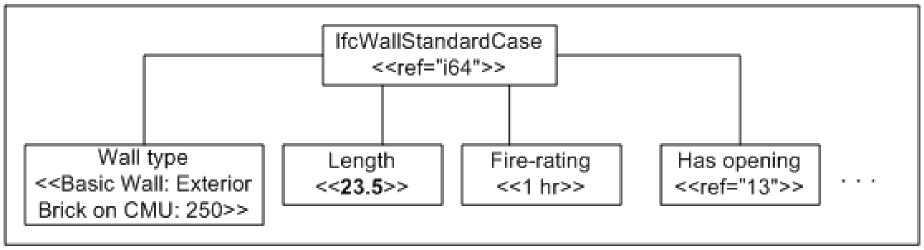


Figure 5: featureXML schema for a Wall component.

level of a simple building. A column is Proximate to another column if both are located along the same gridline. The Proximate column in a given direction – the positive or negative x or y direction from a specified column – must also be the closest such column to the specified column in that direction. For example, in Figure 4, column "81873" is the "northern" (i.e., in the positive y-direction) proximate column of column "80871" along gridline 3.

The Spacing between features is important to construction practitioners because it directly impacts the construction and/or installation of components. For example, practitioners will commonly analyze a building design plan to ensure that the spacing between components is less than the maximum constraint specified by a desired construction method.

While the Intersection relationship and Spacing concepts described above seem fairly straightforward at the abstract level in which they are presented (i.e., at the level of understanding held by domain experts), the need to compose several different queries compounded by the complexity of ifcXML's schema made it very difficult to understand how to formulate such queries. The Intersection query described above, for example, required several days of work by a computer scientist on our team.

4.3. An Intermediate XML Schema: Materializing the Mappings

To address the problem of scalability (see Section 3.1) we also materialize the mappings for a specific building design plan in an xml file whose schema corresponds directly to the domain ontology. The resulting xml file whose schema is a simpler, flattened, two-level xml tree. This simpler schema addresses the complexity problem we encountered with ifcXML (see Section 3.1). The simpler schema is automatically materialized by extracting the information represented by the feature ontology from the ifcXML file. Since this intermediate schema represents the feature ontology, it was named featureXML. Some example featureXML data representing our 23.5 foot long wall is shown in Figure 5. As can be seen, finding the length of a specific wall is much simpler in featureXML than in ifcXML (Figure 1(a)).

While the initial process of transforming ifcXML into featureXML is still incurred, it only has to be done once, and the cost of all subsequent queries posed by the user on the flattened featureXML have a much greater performance.

4.4. Semantic Modeling System Summary

Our Semantic Modeling System demonstrates how a conceptual model can significantly improve the usability of standard schemas and enable the more sophisticated analysis we describe in Example 1. In particular, our Semantic Modeling System helps the domain user in two ways: (1) it provides a view of the data in the language users understand making it easier for them to specify queries and, (2) because it is much easier to use, domain users themselves can easily extend their queries to new concepts that are in the ontology but have not previously been queried. Our approach also has the added benefit of producing a standard schema that is much smaller and therefore more scalable, which addresses the problem with schema complexity described in Section 3.1.

5. Generalization to Other Domains

The primary barrier to data sharing in many domains is the wide range of applications available to users, applications that are typified by proprietary and incompatible formats as well as user-defined syntax. As in the AEC domain, the e-business [4], finance [8], biology [9] [10] and legislation [12] [13] domains also commonly use standard schemas to address this heterogeneity at a syntactical level. In this section we validate that other domains have similar problems to those that we encountered in our case study, and in some cases these problems are exacerbated because the sheer quantity of data is significantly greater. We also briefly discuss how our Semantic Modeling System could be extended to these areas. The standard schemas that we have studied are in XML; however, there is nothing specific to XML in the results — they would be just as applicable to OWL, RDF, or any other data model.

5.1. Complexity of Standard Schemas

Standard schemas provide a very flexible way to structure and express the knowledge stored by an institution. However, this flexibility is accompanied by an increase in complexity of structure and semantics and, more importantly, in usability. For example, in the e-business domain, [3] provides a rigorous account of the complexity of several different XML e-business standard schemas showing that as the volume and/or complexity of information represented by a standard schema increases, so too does its complexity. To quantitatively compare standard schemas, [3] used the number of structures (e.g., elements, and sub-structures) as a proxy of the complexity of a standard schema and found for every single type of structure considered, OAGIS, a cross-industry XML standard schema for business applications — the most complex standard schema of those considered — had the greatest number of structures and OCF, the simplest standard schema, had the fewest. This complexity problem is one that has also been shown to plague users in the field of biology. For example, Stromback et al. [9] compare XML standard schemas for systems biology and found that one such standard schema, Sequentry XML, has a 26-level tree structure to represent the same information as three other systems biology XML standard schemas — BXML, INSDseq, and EMBxml — each of which have less than seven levels. Similarly, most of the legislative standard schemas introduced in [14] have complicated schemas that are composed of multiple sub-schemas. For example, LexDania a Danish XML standard schema for legislative documentation has one meta-schema, but 41 derived sub-schemas [14]. Clearly trying to understand any one of these standard schemas would be a difficult problem, let alone trying to compare two or more of them.

5.2. Comparison of Standard Schemas

As stated earlier, standard schemas only work when the applications exchanging the data agree on the same standard schema. Unfortunately, there will always be competing standard schemas; different communities of users even within a single domain have different needs and views of the same underlying data. For example, [5] identified sixteen different e-catalog XML standard schemas. We can highlight the reason why so many standard schemas exist by considering a relatively simple business transaction: the placement of a purchase order. In this scenario two different standard schemas are needed. To send the purchase order to the manufacturer, the customer uses an XML-based e-business standard schema such as OAGIS. However, a different Internet commerce XML standard schema such as Internet Open Trading Protocol (IOTP) [7] is required for the manufacturer to send the payment to the bank [3]. As in the domain of e-business, there is ample evidence for the existence of multiple standard schemas in the fields of finance, biology and legislation: between [27] and [8], nine different finance XML standard schemas were presented; in the field of systems biology [9] found eighty-five different XML standard schemas; and, [14] identified six different key legislative XML standard schemas in Europe.

On the surface, this would appear to be a blessing: the more choices, the better. Unfortunately, the underlying differences between standard schemas are often not apparent [4] making it difficult for users to determine which particular standard schema best suits their requirements. Evaluating competing standard schemas is necessary in the development of any platform that supports data exchange. Each paper we reviewed provided a comparison of the standard schemas they were considering. It is important to note that identifying the comparison criteria to be used necessitates a deep understanding of the standard schemas being evaluated. Acquiring such knowledge can be an extremely difficult and arduous task, one that will be magnified by the number of standard schemas that need to be compared. For example, in the finance domain, Knox [28] states that financial service providers are being pressured to decide which XML standard schemas to adopt, but implies that the choice is confusing due to so many competing and/or overlapping standard schemas. Stromback et al. [9] state that even at a much smaller scale in the sub-domain of systems biology that looks at molecular interaction, there still exist major differences in the information represented by XML standard schemas [9]. In the ESTRELLA project, a comprehensive comparison of the available standard schemas for the legislation domain, it was necessary to extract “the best and most convincing principles” that could then be applied in developing a single integrated solution [14].

5.3. Custom Domain-Specific Solutions

Regrettably, the identification of differences in standard schemas is not an easy task. To compound this problem, support for comparing standard schemas has not been forthcoming. This forces those responsible for selecting an XML standard schema to develop their own custom comparison frameworks for their particular domain and application. Within the e-business domain, [5] developed a six-level evaluation model (data types, vocabulary, documents, processes, framework and meta model) with three general criteria for analysis (the standardization organization and methodology and, the content of the standard schema). It was then necessary for the authors to review the documentation and content of every one of the thirteen e-business XML standard schemas being compared to determine if, and to what extent, each standard schema met the criteria set forth in their comparison framework. In both [9] and [11], the authors found it necessary to create a complex comparison framework to help them evaluate the systems biology XML standard schemas under consideration. In [9] both a general comparison on name, version, definition, purpose and data and a more in depth comparison on content (including if the standard schemas provided information on subjects such as interactions and pathways) is provided. In [11], Stromback proposed a more formal two-dimensional comparison framework, one for semantic concepts and the other for automatically identifying matches between standard schemas. In the legislative domain, Lupo et al. [14] used a comparison framework comprised of six difference criteria each composed of several different parameters; such a framework would require users to have a solid understanding of the intricacies of each standard schema to be able to extract all of the information required by the comparison framework.

Obviously there is a need for some sort of system to address the usability problems associated with XML standard schemas. Our proposed solution, a Semantic Modeling System, goes beyond the solutions presented within the other domains we investigated. In particular, as validated in the AEC domain, our Semantic Modeling System can help to solve these much more global cross-domain problems. We hope to extend our Semantic Modeling System to be a more generic solution, however, there still remain additional challenges to enable users to better work with standard schemas. Section 6 reviews some of those challenges that our Semantic Modeling System has not yet addressed.

6. Challenges for the Data Management Community: Understanding and Comparing Schemas.

It is natural for humans to introduce complexity. In standard schemas, this is manifested both in the complex structure of the standard schemas themselves and in the range of different standard schemas available, even within a single domain. For example, in the sub-domain of systems biology alone, there are at least eighty-five different standard schemas to represent the knowledge therein [9]. Preventing complexity in standard schemas is nearly impossible. A far more realistic strategy would be to focus efforts on supporting users in managing the complexity by creating tools that can make XML standard schemas easier to understand, easier to use and easier to learn.

It is infeasible to prevent the existence of multiple overlapping standard schemas — semantic differences will always exist. Instead we must find ways to make it easier for users to understand and compare competing standard schemas. The solution presented in the other domains we investigated and that we adopted in our Semantic Modeling System was to develop a comparison framework to identify the relevant information; this is the conceptual model we identify in Section 4.1. However, creating the comparison framework constitutes only the first step in evaluating and comparing standard schemas; it merely provides the instrument for comparison.

To determine how well each standard schema represents the concepts identified in the comparison framework requires a sound understanding of the standard's schema or, in other words, understanding what content is represented by each standard schema as well as how it is structured. In our case study, this task was extremely time-consuming and took months of painstaking work, the bulk of which was spent on figuring out the structure of ifcXML. Much of the work required to create a comparison framework and then determine which concepts in the framework are supported by each standard schema under investigation could have been prevented. For example, if we had had a tool to assist us in the creation of the comparison framework and to help us discover compliance with this framework semi-automatically, the process of selecting ifcXML for our purposes would have been much more efficient. Since the standard schemas, as in many real world applications, are imposed on us, they do not adhere to the recommendations on the literature on how to create schemas that are easy to understand (e.g., [29, 30, 31]).However, looking at these works can help understand how advanced schemas differ from those created for novice users — who are not focused on long term understandability. Similarly, schema visualization literature (see [32] for a survey) shows both (1) what parts of schemas are crucial to initial understanding and (2) what parts of schemas must then be revealed. Others have motivated that databases need to be more usable [33] in general, which shows that these types of problems are broadly felt throughout databases.

Some emerging approaches aim to allow users to query without knowing the schema (e.g., [34]). However, in the many applications (such as the case study in Section 2), the programmer needs to be able to answer semantically deep queries consistently, and that is not going to happen without understanding the schema. Other emerging work is on schema summarization — summarizing the schema so that users can have a general idea of what is in the schema [35]. While this is helpful for trying to understand where to begin, it is insufficient for those who need to understand the schema in sufficient depth to write queries.

The data management research community must build on these works and help to solve the problems so that data, especially spatial data such as CAD data, can be integrated and used effectively.

7. Conclusion

As shown in this paper, there is a great need for open exchange of data in a variety of domains such as AEC, biology, e-business, finance, and legislation. This need has resulted in a number of different data representation standard schemas being proposed in these domains. A significant challenge which occurs as a result of the multiplicity of standard schemas is in choosing which standard schema should be used for a particular application. Developing an understanding of each standard schema which is sufficient to make an informed choice is prohibitively time consuming for the domain experts who are defining the application's requirements. This may be one reason there are so many standard schemas: it may be easier to create a new standard schema than to determine which of a given set of existing standard schemas meets the user's requirements.

We have addressed this challenge by proposing our Semantic Modeling System. We used our Semantic Modeling System in a case study in the AEC domain to evaluate the standard schemas. While it does not solve all of the challenges inherent in standard schemas, it did show two main benefits:

1. The task of evaluating a standard schema takes the form of a set of specific questions (e.g., does the standard schema represent feature X?)
2. Standard schemas can be compared quantitatively based on the number of features they represent.

Because our Semantic Modeling System uses an ontology, we also allow the comparison framework to be tailored to the needs of the individual who is performing the evaluation.

We understand that the complexity of standard schemas are a necessity of their expressiveness. We found four tasks in particular were the most cumbersome for users: determining criteria for comparing standard schemas, understanding a schema, matching concepts in different schemas, and mapping concepts between schemas. We believe that fully realizing the potential of interoperability of data — particularly data that exists in spatial or multimedia applications rather than natively existing in databases — requires creating methods and tools which support a better understanding of schemas and data for users who are not necessarily data management experts. Only then can we hope that users will be able to fully make use of all the data that their applications include.

Acknowledgements

This work is funded in part by NSERC Canada. The authors wish to thank Raymond Ng and the anonymous reviewers for their helpful comments on previous drafts of this work.

References

[1] M. J. Pratt, Introduction to iso 10303—the step standard for product data exchange, Journal of Computing and Information Science in Engineering 1 (1) (2001) 102–103.

[2] W. Behrman, Best practices for the development and use of XML data interchange standards, Tech. rep., Center for Integrated Facilities Engineering, Stanford University, USA (2002).

[3] H. Li, XML and industrial standards for electronic commerce, in: Knowledge and Information Systems, 2000.

[4] D. J. Kim, M. Agrawal, B. Jayaraman, H. R. Rao, A comparison of B2B e-service solutions, in: Communications of the ACM, 2003.

[5] V. Schmitz, J. Leukel, F.-D. Dorloff, Do e-catalog standards support advanced processes in B2B e-commerce? findings from the CEN/ISSS workshop eCAT, in: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), 2005.

[6] J. Nurmilaakso, P. Kotinurmi, H. Laesvuori, XML-based e-business frameworks and standardization, in: Computer Standards and Interfaces, 2006.

[7] The IETF Trade working group, Internet Open Trading Protocol (IOTP), Technical Report. December 30, 2002. Available at xml.coverpages.org/otp.html. Accessed April 7, 2010.

[8] A. Malik, XML standards for financial services, www.xml.com/pub/a/2003/03/26/financial.html (2003). Accessed April 7, 2010.

[9] L. Stromback, D. Hall, P. Lambrix, A review of standards for data exchange within systems biology, in: Proteomics, 2007.

[10] F. Achard, G. Vaysseix, E. Barillot, XML, bioinformatics and data integration, in: Bioinformatics, 2001.

[11] L. Stromback, A method for comparison of standardized information within systems biology, in: Proceedings of the 37th conference on Winter simulation, 2006.

[12] A. Marchetti, F. Megale, E. Seta, F. Vitali, Using XML as a means to access legislative documents: Italian and foreign experiences, in: ACM SIGAPP Applied Computing Review, 2002.

[13] R. Winkels, A. Boer, R. Hoekstra, Metalex: An XML standard for legal documents, in: Proceedings of the XML Europe Conference, 2003.

[14] C. Lupo, F. Vitali, E. Francesconi, M. Palmirani, R. Winkels, E. de Maat, A. Boer, P. Mascellani, Deliverable d3.1 general XML format(s) for legal sources, Tech. rep., ESTRELLA: European project for Standardized Transparent Representations in order to Extend Legal Accessibility (2007).

[15] buildingSMART, ifcXML2x3 Release www.iai-tech.org/products/ifc_specification/ifcxml-releases/ifcxml2x3-release/summary (2010). Accessed May 5, 2010.

[16] M. Nepal, S. Staub-French, J. Zhang, M. Lawrence, R. Pottinger, Deriving construction features from an IFC model, in: Canadian Society for Civil Engineering (CSCE) annual conference, 2008.

[17] M. Ibrahim, R. Krawcyzk, The level of knowledge of CAD objects within the building information model, in: Association for Computer-Aided Design in Architecture Conference, 2003.

[18] T. Froese, M. Fischer, F. Grobler, J. Ritzenthaler, K. Yu, S. Sutherland, S. Staub, B. Akinci, R. Akbas, B. Koo, A. Barron, J. Kunz, Industry foundation classes for project management - a trial implementation, in: ITCon, 1999.

[19] John Kennedy, About gbXML, www.gbxml.org/aboutgbxml.php. Accessed April 7, 2010.

[20] California CAD Solutions, DWF: The best file format for published design information, White paper, Version 1.2, April 2004. www.calcad.com/products/docs/Autodesk%20DWF%20WhitePaper.pdf. Accessed April 7, 2010.

[21] Autodesk, DWF 6 specification.

[22] A. Doan, A. Halevy, Semantic integration research in the database community: A brief survey, AI Magazine 26 (1) (2005) 83–94.

[23] P. Shaviko, J. Euzenat, A survey of schema-based matching approaches, Journal on Data Semantics IV (2005) 146–171.

[24] Protégé, protege.stanford.edu (2010). Accessed May 5, 2010.

[25] M.P. Nepal, S. Staub-French, J. Zhang, M. Lawrence, R. Pottinger, Deriving Construction Features from an IFC Model, Canadian Society for Civil Engineering (CSCE) annual conference, 2008.

[26] World Wide Web Consortium (W3C), XQuery 1.0: An XML Query Language, www.w3.org/TR/xquery (2007). Accessed May 5, 2010.

[27] A. B. Coates, The role of XML in finance, in: XML Conference & Exposition, 2001.

[28] M. Knox, Commentary: XML in the financial industry, `news.cnet.com/2009-1001-275607.html` (2001). Accessed April 7, 2010.

[29] F. Bodart, A. Patel, M. Sim, R. Weber, Should Optional Properties be Used in Conceptual Modeling? A Theory and Three Empirical Tests, Information Systems Research 12 304–405.

[30] S. Kalyuga, P. Ayres, P. Chandler, J. Sweller, The Expertise Reversal Effect, Educational Psychologist 38 23–31.

[31] Khatri, Vessey, Ramesh, Clay, Park, Exploring the Role of Application and IS Domain Knowledge, Information Systems Research 17 81–99.

[32] T. Catarci, M. Costabile, C. Batini, Visual Query Systems for Databases: A Survey, Journal of Visual Languages and Computing 8 215–260.

[33] H. V. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi, C. Yu, Making database systems usable, in: SIGMOD Conference, 2007, pp. 13–24.

[34] T. Jayram, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, H. Zhu, Avatar information extraction system, IEEE Data Engineering Bulletin 29 (2006) 40–48.

[35] C. Yu, H. V. Jagadish, Schema summarization, in: VLDB, 2006, pp. 319–330.

[36] A. Webster. Semantic Spatial Interoperability Framework: a Case Study in the Architecture, Engineering and Construction (AEC) Domain. Master's thesis, Department of Computer Science, University of British Columbia, 2010.

[37] J. Zhang. Evaluations on XML Standards for Actual Applications. Master's thesis, Department of Computer Science, University of British Columbia, 2008.