Lightweight Capture of As-Built Construction Information

E. East
Engineer Research and Development Center, Champaign, IL, USA

N. Nisbet, J. Wix
AEC3, London, UK

ABSTRACT: Discussions of Building Information Modeling (BIM) often center on commercial software and building geometry. Such discussions typically ignore the need for information capture and exchange during construction. This paper identifies several patterns of information exchange needed for the capture and delivery of construction handover data. A prototype tool set, bimServices, has been developed to support of these information exchange patterns. The paper also includes specific examples of how the prototype tool would be used to capture and transmit as-built construction information.

1 INTRODUCTION

The use of BIM during construction has focused, for the most part, on the visualization and manipulation of the geometric model elements. Anecdotes of large commercial contractors indicate that they often create a BIM for complex areas within a given project to resolve design discipline coordination problems prior to construction. Manipulation of geometric objects in three dimensions prior to construction reduces on-site conflicts due to hard and soft collisions. A leading U.S. trade magazine, Engineering News Record, reports that one hour spent in coordination activities results in ten hours of saved field re-work [Post 2009]. Adding of the time dimension to BIM allows the creation of so-called 4D models. The objective of such models is to reduce on-site conflicts due to trade scheduling and or movement of products in material storage yards. The practical use of 4D models is most highly visible on time-constrained premier projects, such as China's Olympic stadium, where the project team uses technology to links building elements with construction sequence [Tekla 2009].

The use of commercial software to display geometry in static and time-sequenced approaches is the most easily adopted aspect of BIM by commercial software firms and users of their products. These models are typically created by one "BIM jockey" at the prime contractor's office whose job is to present the model and identify problems that are resolved to update the design. Ideally, these changes are fed back into the BIM model and result in the production of improved construction contract documents. Today's efforts require a single party, typically the architect, to be responsible for the creation and publication of a model. Such "point applications" of BIM are far from the vision of models that capture the entire transformation of owner functional requirements throughout the project life-cycle. The life-cycle perspective is a key tenet of buildingSMART and large public owners such as the Corps of Engineers who are adopting open standards approaches for BIM exchange.

The American Institute of Architects recently released its document describing the expected evolution of design practice called the Integrated Design Practice [AIA 2009]. In that document, AIA acknowledges that while the design executive maintains overall project leadership, there are parts of the design that are led by appropriate consultancies. AIA's position demonstrates that there will not be a single point of BIM authorship during the design phase. At different points in the project, different team members have different levels of responsibility for those aspects of the design. This organizational model implies that the idea of interoperability must be transformed from a single unified model that provides all required information to all needed stakeholders into an idea of contracted information exchanges.

For many practitioners today, there is a perceived correlation between the question of proprietary technology and the definition of information exchanges. It is assumed that a single proprietary software stack will provide a seamless transmittal of information through the entire set of project stakeholders. This assumption is incorrect.

Regardless of the technology used, if information is to be shared, the precise definition of the content and timing of that deliverable is critical to the successful use of the information by others. Thus the essential issue with BIM is not really one of technology, but one of defining the specific requirements for information exchange.

For many public owners, as well as owners interested in maintaining control of their facilities information, the use of proprietary data formats and media is problematic. Over time, proprietary data formats are superseded by later non-compatible versions and storage media becomes obsolete. Owners interested in ensuring competitive markets and maintaining control of their facilities' information require open standards. The commonly acknowledged open standard for building information is the Industry Foundation Class (IFC) model. IFC, however, only provides a starting framework for the definition of information exchanges. Like the ASCII character set, the IFC format only provides the letters or building blocks for the information to be exchanged. The contents of any given exchange must still be specified.

In a world where different stakeholders (including those without direct contractual obligations) provide information that should be captured as part of the life-cycle history of the project, a precise definition of information exchange content, timing, and format is needed. Fortunately, in the construction phase, the specification of what information is to be provided when is not a new problem. Today's construction contracts already detail a myriad of deliverables ranging from pre-construction meeting minutes at the start of a project, to daily reports and jobsite photos, to a listing of spare part suppliers at the conclusion of a project. The job at hand is to transform current, largely discarded and usable, document-based information deliverables into open computable information exchange standards.

The issue of open standards information exchange is starting to be addressed by a variety of stakeholders primarily through the buildingSMART organization. Using the Information Delivery Manual process (IDM), user requirements may be objectively captured and mapped into the IFC model to produce Model View Definitions (MVD). MVDs define the subsets of the IFC model that apply to specific information exchange requirements. Once MVDs have been defined, the problem of how to exchange such information becomes critical. Since today's information exchange platforms are expensive and require significant training they are not likely -- with the notable exceptions of elite projects -- to be used at the construction trailer.

2 OBJECTIVE

The objective of this paper is to outline a set of building information server tools that can be used to capture pre-defined batch and transactional information deliverables commonly encountered at the construction site. These tools are characterized as "lightweight" because commercial BIM authoring software and Model server technologies are not needed to interact with the information content of the building model.

3 BACKGROUND

The Construction-Operations Building Information Exchange (COBIE) specification identifies content of information delivery required by specific project team members during design, construction, and commissioning [East 2007]. During design stages, the required information is found within BIM software, so the expected deliverable will be provided through the COBIE Model View Definition [Wix 2008]. Since the vast majority of construction contractors cannot be expected to develop BIM expertise at the job site, construction contractors can extract and view the information needed for construction handover using inexpensive, ubiquitous spreadsheet technology.

Commissioning agents are also able to enter maintenance plans and system instructions in the COBIE spreadsheet format. As the COBIE spreadsheet is completed, links to BIM object models are maintained by the references found in the individual worksheets. At the conclusion of construction, COBIE data may be imported directly to the facility manager's computerized maintenance management system or facility asset management system.

The template spreadsheet and example projects provided through the Whole Building Design Guide [East 2009] are not intended to be the ultimate means of transmission of COBIE data. These spreadsheets provide examples that quickly communicate the set of information needed to replace the cubic yards of paper documents currently provided at construction handover. As COBIE production and use are implemented directly within commercial software, use of the spreadsheet format for COBIE data will diminish. Table 1 lists the commercial software vendors who have demonstrated their product's compliance with either the production or consumption of COBIE test data [buildingSMART 2008, 2009].

Table 1. Software Meeting the COBIE Challenge

Product	Manufacturer	Phase
ArchiCAD	GraphiSoft	Design
Architect	Bentley Systems	Design
MicroMain	MicroMain	FM
OPS	Onuma	Multiple
Room Data	Project BluePrint	Construction
Revit	AutoDesk	Design
TMA	TMA Systems	FM
TOKMO	Tokmo	Construction
VectorWorks	Nemetscheck	Design

There are four formal exchanges of COBIE data during construction, as shown in Table 2. The first exchange is from the designer or owner to the contractor. This exchange contains the "as-designed" COBIE building information. The as-designed set includes, but is not limited to, the following objects: floors, spaces, zones, equipment, and systems. This set also describes the assets in the building. Space function and fixed/movable assets are identified in the as-designed set. The as-designed set may be used by the contractor as a batch import of the building information. The first COBIE deliverable supplied by the contractor is the "pre-built" model. The pre-built model is required at 70% fiscal completion of the project and includes the majority of manufacturer and related information provided as part of a submittal approval process. The "occupancy model" includes the 90% building information model including the list of values and other tagged items along with system operations and maintenance instructions. The as-built model contains the 100% building information model and is provided as part of the contract closeout documents.

Table 2. COBIE Construction Deliverables

Use	Name	Contents
Consume	As-Designed	Room, Equipment Types/List
Supply	Pre-Built	Mfg data, Warranty, Parts
Supply	Occupancy	Tagged Items, Commissioning Data
Supply	As-Built	Updated Data

The COBIE specification describes the timing and format of the needed deliverables, it does not provide means and methods to supply or consume the required information. The specification does imply, however, by the sequencing of the deliverable that the most effective way to capture the required COBIE information is to capture it as it is created during the project life cycle. Capturing the required information during the project is expected to result in a significant reduction in the cost of creating construction handover information. For example, documenting the manufacturer, model, and serial number of equipment as it is installed will eliminate the need for an expensive post-project "job crawl." Capturing construction submittals electronically can also fully eliminate the post-project effort currently needed to deliver mounds of paper to the owner at the projects fiscal completion.

4 INFORMATION EXCHANGE METHODS

Assuming that COBIE information can be captured as the project progresses, several classes of information exchange during construction may be identified. These exchange classes are summarized in Table 3.

The first of these is the batch consumption of COBIE as-designed information. The as-designed information is that COBIE spreadsheet provided to the construction contractor by the owner or designer at the start of the construction contract. When this information is received by the construction contractor, two types of triggers may be invoked. Prior to the acceptance of any building information for use, the contractor will want to check the information to make sure a correct file has been received. Since the majority of contractors will not have the time or expertise needed to check the incoming data file, a triggered method that must be included in any information exchange tool is to automatically verify the incoming information against the specification of the format for that model.

Another case that should be considered when evaluating the as-designed model is the case where the as-designed model changes over time. This complicates the batch import process because in addition to checking the accuracy of the information format, changes made from previous models, must be clearly identified. As a result, if the model being imported is an update of an existing model, then there must be a triggered method that will identify all changes from the previous model. Understanding which objects are allowed to be overwritten and which objects can simply be added requires considerable effort due to the possible linkages between objects that must be created and removed as a side effect of even small changes to the model.

Requiring a user to track all of these relational links, or understanding how to check the links that were changed by BIM authoring software is beyond the capacity of the typical set of workers at a construction trailer or back office. If information exchanges are to be effective, tools must hide the complexity of object modeling from the users. One approach toward this end is to provide simple interfaces into the building model through commonly used software such as spreadsheets.

After construction starts, information must be captured in a way that supports the production of the COBIE pre-built model required at 70% earned value. Information captured during the project leading to this milestone may be supported by one of several types of transactional methods. The first type of transaction would be to update existing information in the COBIE model. Providing the make, model, and serial number for a specific piece of equipment would be a simple example. Adding information to the building model through COBIE is another needed transaction. An example is the addition of valves and other “tagged” items to the installed equipment list. These tagged items identified during construction, but not specifically called out in a design schedule are, according to standard construction contracts, required as part of the construction handover data set.

Table 3. Transactions' Functional Requirements

Function	Example of Use
Query	Find Object and Related Objects
Trigger	Verify against specification
Trigger	Verify against previous model
Add	Equipment Serial Number
Add	Tag and Room Number
Update	Change Several Objects
Delete	Remove Object and Links

Another significant set of transactions that must be made against the model are those that reflect construction change orders. We must assume that change orders could affect any of the information contained within the COBIE model. Given the tightly linked nature of the data in the COBIE data set, the typical construction quality control or production officer will need assistance to understand the full set of linked changes resulting from large change orders. A model query tool should be provided that identifies the set of specific objects to be reviewed and shows the set of related information that must be updated to ensure that the model remains consistent. An additional function, object deletion, will also be needed for the series of discrete transactions needed to document change orders. Once the user has identified the appropriate changes, the changes must be made to the model and the changes verified against the previous version.

5 LIGHTWEIGHT MODEL SERVER

5.1 *Assumptions and Pre-requisites*

Information about assets begins to be identified during early design stages and undergoes a series of augmentations and transformations during the project life cycle. The overall process sequences may vary in different contexts based on specific contract arrangements, but, the required information exchanges can be identified and decomposed. The goal of a building asset model is to deliver a single set of information based on the compilation of many discrete information exchanges.

The buildingSMART IFC schema provides a common neutral data format for the exchange of building handover information. The IFC (ISO-16739) common neutral schema can exist alternatively as an Express (ISO-10303) schema or as an XSD (XML schema definition). While the expressive power of Express is superior to XML it is assumed that for well specified exchanges required in contracts that the dual expression of the IFC model can provide a transparent exchange between files in the Step Physical File Format (SPFF) or the eXtensible Markup Language (XML) format.

While the majority of larger BIM models will eventually be delivered in IFC, this research will take advantage of the transformation between SPFF and XML as the means to simplify the tools needed to interact with the data within the model. Use of XML significantly simplifies the use of data originally created in IFC since XML Cascading Style Sheets (CSS) allow large XML datasets to be displayed according to references between XML tagged information and HTML layout objects. By including XPath functions within the style sheet, information may be transformed from one XML structure to another and then displayed in that new structure.

Use of the extensible style sheet transformation (XSLT) (i.e., style sheet plus XPath programming) provides a common method for developers and users of building information to directly interact with their models in ifcXML format. The tools to execute the XSLT interactions and interpret the resulting files are included on virtually all computers since these functions also serve as the backbone for display of information provided through the World Wide Web. Microsoft's XML Core Services (MSXML) and the Firefox browsers both provide direct support for XSLT. Initially transforms will be created by experts but as this technology becomes used local technical staff at design, construction, and operations offices will be able to modify these sample transforms directly with BIM models in a variety of innovative and useful ways going well beyond what is available today.

Given the ingenuity born of necessity found at many construction field offices, it is likely that if given the definition of standard ifcXML BIM deliverables (such as COBIE) and the current availability of inexpensive commercial reporting tools used to shape and transform that information, using XSLT, there will arise a community of users developing and exchanging these XSLTs for their specific needs. The first author's personal experience with microcomputer user groups when desktop computers first came into widespread use showed that even with the limited hardware and coarse software of that time, innovative civil and mechanical engineers with real problems to solve were able to craft and exchange practical solutions to their own problems.

Of course there will always be times when highly geometric information will be needed to evaluate the placement of specific objects or the location of components within large rooms. It is also assumed that there will continue to be a wide range of tools that are publicly available. Links to many of these products may be found through the "software downloads" page of the ifcwiki [IFC 2009]. One notable recent entrant into the list of free software for viewing IFC files is the Adobe Acrobat viewer.

5.2 *Overall Description*

The "bimServices" application has been designed as a handful of simple command line programs. Each program is intended to effect a single task. Together these programs offer sufficient flexibility to meet the immediate needs of those receiving BIM models. The programs may be operated through the command line or by sequential operation of a planned set of individual operations using batch files. bimServices may be used directly by knowledgeable construction users, by those who received pre-scripted information exchanges for specific types of transactions, and may also be directly incorporated as a "black box" into business process services looking to add BIM transactional support to their existing suite of tools.

One of the most important features of the bimServices engine is that it accepts and generates IFC or ifcXML and cross-converts the two formats as specified. This ensures that the physical format of the delivered file (either SPFF or ifcXML) is not relevant to the use of the file by means of XML transformations. This translation service is enabled by the "ifcEngine" program, created by the by the TNO Building and Construction Research Office, Delft, Netherlands.

bimServices matches the XSLT required to obtain and format model data with the native operating system or browser-based XML transformation engine through the "transform" program. The transformation accomplished does not require any external sources of information or any pre-conception about what the state of the model should be. The importance of this feature, often referred to as "late binding," is that the data model needed for the transformation is not tested until the transformation is run.

The alternative to the late binding approach, briefly considered by the authors, is "early binding." Early binding requires that specific data model be known by the program prior to the execution of any code related to that data model. Given the myriad of possible types of information transactions, transformations, and presentations that may be needed by BIM users, demanding a specific requirement of a BIM data set before it can be processed was an unacceptable constraint on the development of the light-weight BIM server.

5.3 *Transformations*

"Transform" is the bimServices tool that enables connections between IFC SPFF and ifcXML to be defined and managed. The application accepts an XSLT transformation file, appropriate "asset" file, and several additional command line parameters. bimServices provides a front end to support transformations between a variety of file types based on XSLT. Asset files might include, for example, structured spreadsheets and issue reports from model checking programs. The transformation between the IFC SPFF and XML (or XHTML) documents are illustrated in the following figures. Figure 1 shows a snippet of an IFC file containing a single object, one room within a fictional building. Figure 2 shows the resulting ifcXML code produced from the transform bimService.

```
#7356=IFCSPACE('03zmjESy97ZewfRSCN11i1',#7468
,'003','Lobby area that leads to male and female toilets',
'F321',#7225,#7206,'Lobby',.ELEMENT.,.NOTDEFINE
D.,$);
```

Figure 1. ifcSpace fragment of an IFC SPFF file

While some transformations will be required as part of contractually identified information exchange requirements, others may be made by local construction personnel to facilitate reporting information out of the BIM. Examples of such reports could include equipment schedules with links to approved submittals and shop drawings.

A crucial idea behind this work is that access to and use of BIM data should not require special purpose BIM software and on-site trained staff typically available only to elite construction offices and projects. Transformation of XML information into HTML provides a critical link in making BIM information transparent to end users. Figure 3 shows an example IFC file whose exhaust fan equipment schedule has been extracted using an XML transform and displayed in HTML format. Note that the transform captures not only the equipment name and type but also its location and associated properties.

```
<IfcSpace id="i7356">
  <GlobalId>03zmjESy97ZewfRSCN11i1</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" ref="i7468"/>
  </OwnerHistory>
  <Name>003</Name>
  <Description>Lobby area that leads to male and
      female toilets</Description>
  <ObjectType>F321</ObjectType>
  <ObjectPlacement>
    <IfcLocalPlacement xsi:nil="true" ref="i7225"/>
  </ObjectPlacement>
  <Representation>
    <IfcProductDefinitionShape xsi:nil="true"
      ref="i7206"/>
  </Representation>
  <LongName>Lobby</LongName>
  <CompositionType>element</CompositionType>
  <InteriorOrExteriorSpace>notdefined
    </InteriorOrExteriorSpace>
</IfcSpace>
```

Figure 2. Transformed ifcXML ifcSpace fragment

Since common office tools such as spreadsheets, word processing files, and smart forms are based on underlying XML schema, transformations from models to standard office documents is also possible. Figure 4 shows a snippet of an IFC file that was transformed into an XML spreadsheet that meets the COBIE spreadsheet requirements. Notice that the transformation includes formatting that will be helpful to transmit the importance and use of the information to the user at the construction trailer.

Of particular interest in Figure 4 are the columns identified as "ExternalSystemName" and "ExternalNameID." These columns specify the exact software used to create the model and the Globally Unique Identifier (GUID) that relates the data in the XML spreadsheet back to the original building model. Since the worksheet is the Space worksheet, we also know that the IFC entity to which the GUID is applied is the ifcSpace object.

A critical concern of this research is to ensure that information may be transformed from the building model, updated, and seamlessly integrated back into the model whenever required. Such "round tripping" has been proven not to be effective in geometry-based information exchanges such as those supporting the IFC Coordination View. Explicit entity mappings and maintenance of the GUID are the keys, in bimServices, to accomplishing the ability to provide successful round tripping.

For example, when comparing the contents of a floor of a building in the new file the comparison program will begin by reviewing any added or deleted spaces when compared to the original file. Where spaces are found in both models, the algorithm will then start with the first space and investigate any changes within each space.

Maintenance Facility Type, Component and Attribute Schedule

For types 23-75* and components EF*.

23-75 00 00 Climate Control (HVAC) : HVAC System Components and Equipment

Component	Description	Location	Area Served	Make	Model	Power (HP)	Voltage (V)	Weight (lbs)	Controlled by	CFM	S.P. In. W.G.	Accessories
EF-1	Exhaust Fan	Prefab Fluid Storage Bldg.,	High Bay.	Greenheck.	SB-2L30-7.	0.75 (HP) .	208/3/60 (V) .	125 (lbs) .	co sensor, on/off switch, and thermostat.	8000.	0.125.	.
EF-2	Exhaust Fan	Prefab Fluid Storage Bldg.,	High Bay.	Greenheck.	SB-2L30-7.	0.75 (HP) .	208/3/60 (V) .	125 (lbs) .	Co sensor, on/off switch, and thermostat.	8000.	0.125.	.
EF-3	Exhaust Fan	Communication room,	Electrical Room.	Greenheck.	CSP-A510.	217 (W) .	115/1/60 (V) .	36 (lbs) .	Thermostat.	350.	0.25.	.
EF-4	Exhaust Fan	Recessed Maintenance Area,	Recessed Maintenance Area.	Greenheck.	CSP-A710.	325 (W) .	115/1/60 (V) .	36 (lbs) .	Co sensor.	250.	0.625.	.
EF-5	Exhaust Fan	Maintenance Bay,	Parts Washer.	Nederman.	N16.	0.5 (HP) .	115/1/60 (V) .	xx (lbs) .	Light switch.	500.	3.75.	.
EF-6	Exhaust Fan	Maintenance Bay,	Welding Bench.	Nederman.	N16.	0.5 (HP) .	115/1/60 (V) .	xx (lbs) .	Light switch.	500.	3.75.	.
EF-7	Exhaust Fan	Recessed Maintenance Area,	Toilet, Jan., Comm.	Greenheck.	CSP-A710.	325 (W) .	115/1/60 (V) .	36 (lbs) .	.	210.	0.625.	.
EF-8	Exhaust Fan	Prefab Fluid Storage Bldg.,	High Bay.	Greenheck.	SB-2L30-7.	0.75 (HP) .	208/3/60 (V) .	125 (lbs) .	Co sensor, on/off switch, and thermostat.	8000.	0.125.	.
EF-9	Exhaust Fan	Electrical Room,	Office/ Kitchen.	Greenheck.	SP-B150.	129 (W) .	115/1/60 (V) .	10 (lbs) .	On/off switch.	150.	0.125.	Integral Ceiling Grill.

Figure 3. Exhaust Fan Schedule

SpaceID	FloorID	SpaceFunction	ExternalSystemName	ExternalNameID	SpaceNumber	SpaceName	Description
i7356	F1	F321	ArchiCAD 8.0	03zmjESy97ZewfRSCN11i1	3	003	Lobby area that leads to male and female toilets

Figure 4. Transformed MS Excel XML ifcSpace Fragment

5.4 *Comparing*

Conventional file comparison tools are useless for the purposes of comparing building model files because such files provide only structural not semantic, comparisons. bimServices provides a recursive comparison tool that uses IFC spatial and containment hierarchies to effect a meaningful and systematic comparison. By iterating down from the ifcProject object into the spatial hierarchy, each object and its required, optional, and filled properties are compared. Figure 5 displays a text version of the results obtained when comparing two IFC files. Such a text file, or reports transformed from such a file, provides the user a precise and trustworthy evaluation of changes between two model files.

The model hierarchy is used as the basis for the comparison for two reasons. The first is that this path through the model will typically have only incremental changes across different versions of a single project. bimServices is not intended to provide ad hoc comparisons of building models from different projects. Secondly, the IFC implementation of spatial hierarchy is relatively reliable across both software vendor and modeler during the life-cycle of a given project.

To find a match between objects there are several rules that interrogate the model for matching entities. The first, and always most reliable method is to match by GUID. Failing that, entities of the same type, with different GUIDS but matching names will be compared. Finally, if there is only a single instance of a given type in each model, those two instances will be compared.

```
Comparing Bce31.ifc with Bce32.ifc
Reading : Bce31.ifc buildingSMART STEP
Reading : Bce32.ifc buildingSMART STEP
Matched = Default Project, -, Project
Related : ..IsDecomposedBy
Matched = ....Default Site, -, Site (
Related : ......IsDecomposedBy
Matched = ........Default Building, -, Building
Related : ..........IsDecomposedBy
Matched = ............, -, BuildingStorey
Related : ..............ContainsElements
Matched = ................Wall-138, -, WallStandardCase
Related : ..................IsDefinedBy
Matched = ....................WALL, -, PropertySet
Related : ......................HasProperties
Matched = ........................WALL, -,  , ComplexProperty
Related : ..........................HasProperties
Matched = ............................INFO,  PropertySingleValue
Changed < ............................NominalValue Wall-138
Changed > ............................NominalValue Wall-138 A
Finished comparing Bce31.ifc with Bce32.ifc
```

Figure 5: Comparisons Detect Meaningful Changes

5.5 *Merging*

The control of transactions requiring information to be merged from an external source back into a model is a critical aspect of bimServices. First, it is important to state that since bimServices is, by definition, late binding, control of data are to be accepted at which point must be determined by the user accepting the delivered information. To promote user confidence in accepting such deliverables, a clear roadmap of contracted information exchanges must be created, such as that provided by the COBIE project, and comparison and checking routines that highlight variations or violations in file content or structure must be accomplished prior to the merging event. Only after reviewing the changes and/or errors should the user proceed with merging. Once the process of information exchange is well defined and has been used several times with confidence, simple file merges can be quite effective in supporting construction management transactions occurring at the jobsite on a daily basis.

As an example of how such a merge of construction data might occur, consider that model information is to be changed at the construction trailer using XML-based applications such as MicroSoft Excel. In the case of COBIE an example of this type of merged data includes manufacturer names, product models, serial numbers, tag numbers, product attributes, and approved procurement documents. Merging of information added during construction back into the model occurs through the mapping of the IFC model to specific worksheets and maintaining the GUID once an item is created in the model. The business case supported by bimServices is obvious: capturing BIM data in simple tools and merging the information back into the model can eliminate the end of project survey currently required to identify equipment location and stop virtually all end of project document reproduction charges.

5.6 *Filtering*

In particular a BIM will contain a large amount of information not relevant to specific use cases. For example the IFC 2x3 Coordination View includes pipes and elbows for the purposes of checking plumbing design for collisions against work designed by other disciplines. From the point of view of the asset manager, the exact geometric location of these pipes is not relevant. Confidentiality may also drive the release of some information from a model and not other information. Users operating within a building may not want to provide, for example, room names or room function codes that identify satellite storage of hazardous materials. One of the reasons given for the lack of widespread acceptance of IFC files has been, anecdotally identified as the large size of files stored in SPFF or ifcXML. The size of files is the direct result of not having specified the filtering needed to extract only that portion of the model appropriate to a given topic.

Filtering has been included in bimServices to reduce a model to those parts that are of interest for a specific transformation. The filter tool takes two or more parameters defining the object types and their relationships that are to be removed. Once established, the filtering rules can be reused within any similar process during design iterations or in future projects. In the future more complex rules will be embodied in a configuration file, as again the rules may become very specific and complex for a particular exchange requirement.

5.7 *Testing*

bimServices' final component is a testing application. The test is conducted by applying a set of user-defined constraints against a specified building model file. These constraints may be created by knowledgeable users through an XML spreadsheet developed in a very specific format. Each constraint references the function name(s) needed to implement that constraint and any parameters or inputs needed from the model to feed those functions. Once the testing tool is started, these constraints are processed by the referenced algorithms and evaluated across the model hierarchy. The testing component has been applied to evaluate internal conformance against open information exchange standards such as COBIE, Spatial Compliance Information Exchange, and Coordination View files.

The constraint rules included in exchange standards such as COBIE force, for example, names of spaces within a given building to be unique, every piece of named equipment to be located within a valid space, and equipment types to be identified for each individual piece of equipment. The value of such rules is to ensure that the form of the information delivered is consistent with the specifications for that deliverable. This type of consistency testing forms a critical quality control tool needed for software certification as well as for the acceptance of contracted information. Someone at a construction trailer who uses such a checking tool can be confident that they have a well formed building model, and may then start to check the contents against the specific contents expected for the building.

When checking building codes and regulations, the diversity of language and conflicting content in these documents poses a large problem for any checking tool. In addition, much contextual information needed to evaluate codes and regulations is not found in the building model. For example, the bimServices tool was used to evaluate the tabular properties of external doors, walls, and windows found in one public agency's design guide against building models. The difficulty was that the site that would identify the applicable table row was not contained within the building model. The ad-hoc provision of external data sets to enable specific types of checks, linked to the model at the top level, will help solve such problems over the long term. In the meantime, a test to identify the properties that must be known before the checker can run should be of great value.

Assuming that all the required information is provided for a specific rule, there is also a question of the extent to which the computer checker should complete calculations that may need to be accomplished (or checked) by the design professional. Consider the case of exit distances. An automated checker could, for example, conduct an exhaustive calculation based on the geometry of the model to determine the shortest actual path to an external door. On the other hand, an automated checker could, using only the information available in a very simple list of spaces and connecting doors, without any space geometry, determine the minimum number of doors between a given space and any exit. Rules based on such qualitative modeling could help guide designers to areas where there are likely to be problems and provide further evaluation. Such an approach may be better than providing a single "solution" because a designer will always know more about the project than does the rule engine.

It is likely that no single approach will satisfy all the types of checking that needs to be accomplished in a BIM. The provision of a lightweight engine whose constraints and evaluation functions can be changed by the users does open the possibility of engaging many more people in the development of such algorithms than are currently employed by model checking software firms and academics specializing in these areas.

Regardless of the issues found during the checking, bimServices transforms the results of these checks into an IFC file using ifcApproval objects. Supplementary information, such as document references can also be associated. These approvals can be loaded into design or contract management systems or formatted as reports. Examples completed by the authors demonstrate the automated checking of models and the subsequent generation of PDF Smart Forms which is, not surprisingly given the approach of the lightweight model server, just another XML file format.

5.8 *Availability*

The bimServices toolkit is provided free of charge through the IFC-BIM Forum (buildersnet.org/IFC-BIM/). Registration is required. Accounts are individually reviewed and approved prior to being given access with due consideration to site security. bimServices is provided without warranty, and no technical support is provided for the bimServices tools. The TNO ifcEngine component may be downloaded directly from TNO. The ifcEngine is free but comes with a not-for-profit license.

6 CONCLUSION

Construction teams are currently receiving significant value from the use of geometry-based building information. In addition to collision detection and trade sequencing, computable building information may be captured, transformed, and exchanged. While high-end centralized model servers have been touted as the most effective way to augment building information to capture construction information, this paper provides a brief example of a lightweight model server that can be used to display the contents of the models and allow information to be added to the model during the construction process.

The bimServices program demonstrates the value of the information content within a building information model by allowing a set of common transactions with information through a set of command line tools. These tools allow users to view building model data in commonly used formats such as web pages tables and update model information using office productivity tools, such as spreadsheets and smart forms available even on computers found at construction trailers.

The lightweight model server also expands the possible contributors to information filters and transformation programs that work with building information. The use of standard XSLT decreases the software, hardware, and training needed by those in the construction trailer and facility management office to directly access their data. By providing a user group for the exchange of these transforms potential innovations will not be limited to those directly engaged in the development of the IFC model.

bimServices provides a new paradigm for model servers. In this new paradigm, model servers are simple programs that perform simple functions. These servers may either be used directly or embedded within business process engines to enable secure workflow processing for the entire project team.

7 RECOMMENDATION

One of the most important features of the bimServices engine is its open framework. Given this open platform, innovative practitioners and students are encouraged to create their own transformations, filers, and rule checks for information exchange problems that they face. The authors would be interested in cataloguing such applications in future publications.

8 ACKNOWLEDGEMENTS

This work was supported through the Engineering Research and Development Center, Construction Engineering Research Laboratory in Champaign, IL under the project “Life Cycle Model for Resilient and Sustainable Facilities.”

Appreciation is expressed to TNO for the development and support of the ifcEngine component.

Appreciation is also expressed to the anonymous review who contributed to improving the quality of this paper.

9 REFERENCES

East, E.W., 2007. Construction Operations Building Information Exchange (COBIE): Requirements Definition and Pilot Implementation Standard, ERDC/CERL TR-07-30

East E.W., Nisbet N. 2008. Information Exchange Demonstration. buildingSMART Alliance US. cited 05 May 2009 buildingsmartalliance.org/news/20080723.php

East, E.W., 2009. COBIE Challenge. buildingSMART Alliance US. cited 05 May 2009 buildingsmartalliance.org/news/ nmft_cobie_2009.php

East, E.W., 2009. Construction Operations Building Information Exchange Format, *Whole Building Design Guide*. cited 05 May 2009. wbdg.org/resources/cobie.php

IFC 2009. Free Software, cited 15 may 2009 ifcwiki.org/index.php/ Free_Software

Nisbet N, Wix J and Conover D. 2008. "The future of virtual construction and regulation checking”, in Brandon, P., Kocaturk, T. (Eds),Virtual Futures for Design, Construction and Procurement, Blackwell, Oxfordshire

Post, N.M. 2009. “Modeling Pathfinders Impatient to have a much fuller digital toolbox,” Engineering News Record, 25 Feb 2009

Tekla 2009. Olympic Stadium, Beijing, China, Tekla Structuers, cited 05 May 2009. www.tekla.com/us/solutions/references/Pages/OlympicStadium.aspx

TNO 2009. TNO, IFC Engine DLL, cited 29 May 2009. http://www.ifcbrowser.com/

Wix, J. 2008. Construction Operations Building Information Exchange (COBIE) Industry Foundation Class Model View, ERDC/CERL CR-08-2