Creative Construction Conference 2012
June 30 – July 3, 2012, Budapest, Hungary

BUILDING INFORMATION MODEL VISUALIZATION THROUGH JAVASCRIPT-ENHANCED 3D-PDF INTERFACES

Lars Oberwinter
Vienna University of Technology, Vienna, Austria, Oberwinter@industriebau.tuwien.ac.at

Iva Kovacic
Vienna University of Technology, Vienna, Austria, ikovacic@industriebau.tuwien.ac.at

Abstract

Evolving capabilities of CAD software solutions for the building industry have led to a considerable gain of importance of Building Information Modeling techniques over the last years. Being connected by parametric data interfaces such as the IFC standard, all disciplines begin to make use of the evolving opportunities of bidirectional data exchange. However, merging the multitude of complex geometrical and factual Information contributed by the individual stakeholders into an integrated digital building model obviously leads to a variety of challenges concerning the file sizes, structural organization and file handling, especially in large-scale-projects.

In this paper, we propose a method of using Adobe's 3D-pdf engine along with embedded, JavaScript-based interfaces to easily access and handle building information models. To guarantee the persistent and easy accessibility of all relevant information for every party involved, a method was developed to explicitly gain an individual scope of information from the overall data structure and then gather it into a custom made 3D-pdf. By also enhancing the basic 3D-pdf functionality trough customizable, JavaScript-based features, we affirm that even the most complex 2D, 3D and 4D BIM content can be handled and navigated easily by any amateur. Additional automated texturing, lightning and mapping functions in combination with the Acrobat's native functions for altering the building models display provide an eclectic range of high quality graphical outputs far beyond the opportunities of common IFC-viewing-software.

Proposing enhanced 3D-pdfs as an interdisciplinary communication interface, we aim to offer any stakeholder of a buildings planning and construction process a slim and easy to use tool to access, visualize, comment and communicate even the most complex BIM content. Throughout the building's life cycle, it also empowers both building owners and facility managers to inquire any desired building information by simply opening a single pdf file in Acrobat Reader.

Keywords: 3Dpdf, BIM, interactive, interface, visualization.

1. INTRODUCTION

Ascending requirements on the verification of a building's static, thermal, ecologic and economic performance throughout its lifecycle have led to a considerable increase of complexity in the planning process. (Dossick, Neff, 2010; Cicmil, Marshall, 2005) In addition, new design tools and fabrication methods empower modern architects to explore all new levels of geometric complexity in their formal vocabulary. To cope with the variety of challenges in realizing such complex structures in an interdisciplinary planning process, new tools and working methodologies evolved from the conventional CAD-based instruments and principles of operation.

Building Information Modeling (BIM) tools, based on parametric object databases, are about to oust classic two- or three-dimensional CAD programs by offering an enormous facilitation in many areas of the planning process. Just to name a few, BIM affords slimmer and easier-to-use data structures, improved abilities to hierarchically organize building components, dynamic object display modes for all planning phases and –disciplines and a more reliable interdisciplinary coordination of the ubiquitous continuing modifications (McGraw Hill, 2010). Also, by embedding and linking all parametric alpha-numeric information into the individual objects commonly represented by a 3D entity, it is possible to gain individual scopes of data from these entities and recombine them in a multitude of different, multi-platform interpretable file formats.

Nevertheless, the wide variety of heterogeneous software solutions used within and between the different disciplines along with different drawing conventions and scales as well as individual levels of geometric detail lead to a considerable amount of challenges concerning the abilities of technical data exchange interfaces. Indeed, holistic interface approaches such as the Industry Foundation Classes (IFC) are capable of both transferring and displaying complex interdisciplinary geometry and alphanumeric data (Building Smart, 2012). Beside from their main purpose of actual interdisciplinary data transfer, IFC data models can be explored and estimated by a variety of software solutions, ranging from simple IFC viewers such as the Nemetschek- or the DDS-CAD-IFC-Viewer to advanced model-checking and –analysis tools such as the Solibri Model Checker (IFC Viewer 1,2,3, 2012). These programs make use of the hierarchic IFC data structure and allow users to easily navigate through all building components by offering options to hide or display component groups such as stories and building element types.

Along with these obvious advantages of the IFC standard there still remain quite a few challenges: The process of assembling integrated interdisciplinary building models requires a very high (and seldom reached) level of pertinent technical knowledge and detailed conventions between all stakeholders. (Sachs et al, 2010; Plume, Mitchell 2007; Arayici 2011) In large scale projects, the resulting IFC model file sizes can easily exceed a gigabyte, leading to a variety of inconveniences in terms of data transfer, viewing and estimating, especially on mobile devices. In addition, by far not all 3D CAD programs even offer an IFC interface and are only able to deliver geometric data such as the obj- or dwg-file format, which cannot (or only hardly) be assembled with IFC data structures. In addition, besides from the necessity of

installing an IFC viewing program, quite some experience in navigating and handling 3D objects in virtual space is required (but typically not reached) from users such as building owners and facility managers trying to explore the virtual model.

In this paper, we well illustrate a method of merging interdisciplinary 3D data content in heterogeneous file formats and generated by any 3D CAD- or BIM-Software, enriching the geometry with alphanumeric information to hierarchically structure and filter these data and finally assembling it within a slim and easy-to-use interactive 3Dpdf interface.

Once set up and customized for the data structures in an individual interdisciplinary project content, these 3Dpdfs can be updated within shortest time and enable any project stakeholder to visualize and explore just the individual scope of model information demanded within a customizable interface.

2. STATE OF THE ART

2.1 Basic 3Dpdf

Since the release of Adobe Acrobat 7.0 in January 2005, it is possible to display and navigate 3D content in the pdf file format (Adobe 3Dpdf, 2005). Since then, the technical abilities such as dynamic lighting- and shading-modes, model navigation, individual object display modes and many more along with the comparably tiny file sizes resulting from the U3D ECMA (Ecma International, 2007) technology encouraged many branches of industry to make use of this file format for their 3D model presentation and documentation. Nevertheless, 3Dpdfs never really made their way into the building industry – for a variety of reasons:

While small objects, in scales between a mobile phone and a piece of furniture, usually get along with the basic display- and navigation methods offered by Acrobat Reader, taller and more complex building models require advanced features to be easily navigated and examined. Even though Acrobat basically offers a way to navigate, select, hide and show a model's components in the so called model hierarchy tool box, the lack of customizable structure in this list of model entities usually gives users a hard time trying to organize their model display, especially because typically there are hundreds of single objects in a file.

In addition, the later on ease-of-use in pdf of an imported CAD model depends a lot on the way the 3D geometry data were brought into the pdf. Every 3D file format comes along with different possibilities concerning the later on data organization in pdf: For example, converting 3D dwg data into pdf provides the file's layer structure within the Acrobat's model tree, but cannot transfer any object names, texture- or mapping information. Vice versa, importing classic visualization file formats such as obj or 3ds leads to a flat (and by the way not even alphabetically sortable) object list in Acrobat's model tree, making it almost impossible to quickly manage the content.

At the moment, the highest level of hierarchically structured 3D building models in Acrobat is reached when importing IFC data. These allow to navigate the model's hierarchy through stories, building components groups and individual objects with unique IDs. Nevertheless, file sizes are comparably big and there is no (standard) way to embed textures and mapping

information, which leads to a rather poor model appearance. In addition, the aforementioned rather humble spread of programs even being able to produce decent IFC data leaves little room to actually consider IFC the standard case file format for building geometry content at the moment.

2.2 Customized Templates and JavaScript-Control

Aside from the basic tool sets offered by Acrobat to alter a models display settings, Adobe enables users and developers to create custom sets of additional functions providing a build-in JavaScript functionality (Adobe JavaScript, 2012): On different document levels, this JavaScript engine allows not only to create automated model interaction processes (such as hiding all elements of a certain layer, material or name), but also provides the connection to Acrobat's interactive Elements such as buttons and form field elements like drop-down-lists, checkboxes and so on. Even dynamic Flash menus can be used to interact with the 3D content through Action Script (Adobe Actionscript, 2012). In this way, complex model interaction (e.g. isolating a single building story containing hundreds of objects in a model without layer hierarchy) can be carried out within a glance by the click of a button – given that the model's objects are prepared to be controlled by such scripts.

Through template files containing a place holder for the 3D content and a set of interactive elements such as buttons enriched with JavaScript commands to control the model content, it is possible to quickly convert individual geometry files into such interactive 3Dpdf. Specialized software applications such as Deep Exploration (Right Hemisphere, 2012), 3DPDF Converter (Tetra4D, 2012) or Publisher3D (QuadriSpace, 2012) offer tool sets for template design, 3D data conversion and script embedment in one package. This allows users to quickly (and even batch-) create 3Dpdfs from initial CAD geometry without having to worry about the technical details, once the template setup is completed.

Still, in defiance to the enormous variety of opportunities of such JavaScript empowered pdf templates, in practice they are rarely pushed to the full extent of their functionality: Most 3Dpdf templates are confined to offering graphical masks with a few buttons to solve very basic Acrobat operations such as jumping to the next view or change the model's shading- or lightning mode. Commonly, also two-dimensional document content can be altered in such masks, for example to hide or show text fields containing project information or navigation instructions.

The reason for this rather low rate of capacity utilization in common pdf templates seems to be the aforementioned challenges concerning the CAD data structures: Both the 3D content's structure and the scripts have to be perfectly concerted, which requires a high level of conventions in organizing and structuring CAD data: Object names, layer structures, materials and so on have to be perfectly optimized for a given template's script functionality. Probably for this reason, most of the technically advanced examples of 3Dpdfs that can be found on the web are originating in the field of mechanical engineering, where large-scale companies with tight and database-powered CAD structures make use of this technology.

2.3 3Dpdf in the BIM context

In the building industry, especially in the context of today's highly complex interdisciplinary planning process, the heterogeneous CAD software structures used by the stakeholders involved in a specific project lead to a variety of challenges concerning the organization, structuring and synchronization of the data structures (Owen et al, 210) Commonly, each planning discipline is using its own specialized software, layer structures and conventions such as drawing scale or level of detail. In addition, by far not all planning disciplines producing 3D content use BIM-powered software with according data hierarchies or interfaces such as IFC and hence are only able to deliver geometric content structured by file- and layer structures. For these reasons, merging interdisciplinary planning content into an integrated model often leads to a considerable amount of inconveniences in both producing and handling such files: Hundreds of referenced drawing layers, scale conversions and altering levels of detail, huge file sizes, just to name a few.

In fact, some of the common BIM software solutions already offer build-in interfaces to directly produce 3Dpdf-files either through natively build-in functions or through third party plugins: Nemetschek Allplan (Nemetschek, 2012) for example offers opportunities to directly save planning content as 3Dpdf, producing files with embedded texture- and mapping information, a convenient data structure represented in Acrobat's model tree and some basic pdf templates to embed the 3D content into layouts with additional 2D content. QuadriSpace offers Autodesk Revit (Autodesk, 2012) users a 3Dpdf converter plugin with comparable functionality (QuadriSpace, 2012). In fact, these 3Dpdf interfaces in BIM software solutions not only demonstrate the software producer's awareness of the obvious potentials of this powerful file format – they also deliver a very acceptable basic solution for the users of these BIM programs to quickly produce decent 3Dpdfs.

Nevertheless, these possibilities for 3Dpdf creation are commonly confined to the actual native model elements of the respective BIM software. Third party and 'non-BIM' 3D geometry imported into a BIM software, let's say MEP model elements delivered in dwg format brought into an architectural BIM-model, will cause many inconveniences when being exported into 3Dpdf: Imported geometry naturally doesn't contain the same grade of parametric information as a program's native objects and will hence cause troubles when being exported through an interface developed and optimized to handle these native elements. Besides from this, only few BIM programs even offer decent ways of integrating and managing complex 3D content from other stakeholders, not even taking into account further export-matters.

Hence, it seems that at the moment, the possibilities of producing integrated and well-structured 3Dpdfs containing all multi-party geometry and offering the conveniences of well-prepared functional templates is restricted to scenarios, in which all stakeholders are using the same (BIM-) software platform. In this paper, we will demonstrate a new method of merging, optimizing and filtering multidisciplinary geometry content for visualization purposes regardless of the way and software it was created in.

3. PROCEDURAL METHOD

3.1. Automated reorganization of CAD data through MAXScript

To be able to (automatically) merge multi-platform generated model geometry along with the desired alphanumeric data later used to structure it, Autodesk 3DsMax was chosen as the software chassis to solve all primary data transformations. 3DsMax not only offers a wide variety of data interfaces to import all common file formats for three dimensional geometry - through its build-in script language MAXScript (Autodesk 2012) and its wide variety of functions for scaling, transforming, optimizing, renaming, texturing and mapping geometry data, it is possible to automate any individual operation needed to create an appealing and highly functional interdisciplinary model as the basis for the sub-sequent transformation into 3Dpdf. In addition, any kind of ascii-based information stored in text files or charts can be interpreted by MAXScript, allowing to gain any kind of alphanumeric information and to apply it onto the models components.

As MAXScript also allows users to batch all of 3DsMax's file operations such as opening, importing and exporting files from and into a given folder structure, it is possible to differentially process many files by executing a single operation. This is essential because in the planning process, many disciplines creating 3D content still don't use programs with a build-in story-structure but rather work in single files per story. Due to the possibilities of creating graphical user interfaces in MAXScript, all of the functionalities to be described can be gathered in a compact toolbox and easily be run as a plugin on any 3DsMax workstation. The developed file- and folder-picking functions allow operation on any given folder structure.

Making use of these basic 3DsMax functionalities, our MAXScript-based tools are able to quickly collect all planning content from a given (but customizable) folder structure, perform the requested transformations and optimizations and then freely recombine the merged data into new file structures later to be converted into 3Dpdf.

3.2. Data Merging

In the building industry, the different planning disciplines and according companies commonly use their own individual drawing standards in terms of layer structures, working scales and levels of detail. Typically, when being brought together in a project, the stakeholders agree on basic conventions such as their 2D drawings' origin and a project's relative z-point-zero to make data exchange easier. Nevertheless, the aforementioned heterogeneous software structure, the according disparities in the individual data structures and –interfaces and a considerable lack of pertinent user knowledge commonly lead to the use of the least common nominator in terms of geometric data exchange. In addition, especially when talking about non-BIM software solutions without a story hierarchy, it is common practice to construct 3D-elements based on a discretionary z-point-zero in a single file per story, leading to problems when being merged into integrated models with an absolute z-point-zero.

To cope with these challenges, the developed toolbox contains a bundle of functions for automatically scaling and positioning geometry objects properly during the import process into the integrated model. To be able to batch-apply a specific scope of operational functions on a set of files, the tool is designed to work on (customizable) folder structures: In this way, every file contained in a specific folder will be processed by the algorithms designated for this folder. Given that the individual file names additionally contain previously defined information about the building story they belong to (e.g. *Story_01*), it also possible to apply individual functions per file such as the altering z-offset.

Example scenario: An architect uses BIM software to create a ten story building model, let's say ArchiCAD (Graphisoft 2012). Accordingly, all elements in this model are placed on absolute heights. The MEP engineer uses non-BIM software (e.g. some AutoCAD 2004 add-on) to create a model of the ventilation- and heating elements, producing single files per building story and using the individual story's finished floor level as z-point-zero. For the merging, both offices export their geometry into the dwg file format and deposit it in a specific folder. First, the architect's building model is imported into max, containing all building stories in a single file. The developed merging tool now allows to batch-import the individual MEP story files from a given folder into the model, adjusts their drawing scale and offsets the individual elements in height to fit their particular story. All that is needed to be prepared before this operation is a csv-chart or txt-file containing all of the story names and their absolute level heights and the according file name syntax.

Trough this, the time consuming merging process of dozens of individual files from different interdisciplinary stakeholders in a large scale can be solved within shortest time. Thus, with just a few preparations, the assembly of an integrated building model containing all (geometric) information can be solved by the click of a button. Brought into a single file, the imported model elements are now ready to undergo further automated optimizations in the following steps.

3.4. Data Treatment

At the moment, 3DsMax unfortunately doesn't offer an opportunity to directly produce 3Dpdfs. Hence, a choice had to be made, which 3D data file type exportable by 3DsMax suits all the needs for the subsequent conversion into pdf through any third-party-software. After an intensive practical study, the Wavefront file format (.obj) proved to be the most powerful choice: Even though this format only allows to structure the contained 3D data through the object names (no layers or stories), it offers all other desired abilities for visualization purposes: Multilayer materials, mapping information and cameras can be stored in obj-files. Also, 3DsMax-made mesh-optimizations such as vertex- and normal-welding for optimized geometry display and file sizes produce good results when converted to obj and hence deliver ideal conditions for a conversion into 3Dpdf.

Due to the limited amount of characters in a single 3D object's name in obj files, it turned out helpful to encode all alphanumeric information placed in the object names. In addition, this makes a lot of sense for the subsequent object control through JavaScript in 3Dpdf. An

Exemplary encoded object name syntax could be "**GR**130_**MT**00035_**SY**05_**SE1**"

GRoup 130, MaTerial-ID 35, on StorY 5, Structural Element = yes

The group classification is usually based on the imported drawing's layer structure, and is translated based on a chart or text file on import. In this example, group 130 marks all inner bearing walls. The building story value can be applied either through automated checking and comparing an object's z-coordinate with a given story list containing the corresponding absolute story heights, or through implementing the desired part of a filename when importing single-story geometry files. Through marking an object as a structural element, it is possible to easily isolate all bearing elements through JavaScript in Acrobat later on. Material-IDs can be applied based on the 3DsMax material names, allowing advanced display opportunities in 3Dpdf later on. When importing dwg- or other files without embedded material information, materials can also be automatically applied based on layer names (and through corresponding, customizable translation charts).

Even though these four exemplary basic parameters already a allow a wide range of model-handling and –display conveniences in Acrobat, off course it is possible to add and encode any additional user defined parameters into the individual model nodes, such as desired construction time parameters for 4D construction phase simulation in pdf, object's surface areas for the use in facility management, room numbers, and so on. Anyway, in this case the effort depends a lot on the primary type of data – BIM or non-BIM. To enrich non-BIM geometry data (lacking unique object IDs) with additional parameters from charts or text-files, it is usually necessary to pre-structure these objects in corresponding layers delivering the parameter input information, again, through translation charts.

3.3. Gaining and filtering specific scopes of information

Once assembled into an integrated model, the adequately tagged model components are ready to be gathered into arbitrarily arranged new model compositions. Thus, it is possible to provide any specific scope of model information for any project stakeholder: Bearing elements for civil engineers, HVACR-elements in the context of architectural elements for MEP engineers and facility managers, furniture, plants and lights for the building owners. Specifically assembled models for the different construction companies (façade, carcass, earthworks, etc.) can also be assembled and exported within the click of a button.

Afterwards, the exported obj-Files can be batch-converted into respective template-based 3Dpdfs by any pertinent software or in Acrobat itself. These templates then contain the JavaScript-based menus and functions developed for the needs of the individual stake holders and that operate based on the model components' name syntax.

Even though it is also possible to generate integrated 3Dpdfs containing all components, in practice it has shown that file sizes become quite large due to the geometry-intensive furniture- and MEP elements, causing a rather poor graphical performance on standard PCs in Acrobat. According to this, it is a good idea to export integrated models into single-story assemblies, leading to a well-balanced compromise between performance and fully displayable interdisciplinary content.

3.5. Interactive interface design in Adobe Acrobat

Once converted into 3Dpdf, the pre-optimized and tagged model components from the obj-files appear as a flat list of nodes in Acrobats model tree. Through JavaScript-functions working on either the document level or within the so-called 3D-annotation level, it is possible to access different properties of the individual model nodes. In the context of visualization, the most important properties are the node's visibility, material and display mode.

In looped functions, all model components can now be identified and accessed either through their entire name or through parts of it, which allows altering their properties either group wise or individually. For example, all objects with names containing the label "SE1" (bearing structure element) can be isolated through a looped function showing these and hiding all elements without this label. Another example would be to always display all elements containing a glass material ID as transparent, even when the scene display mode is switched to a state with no transparency intended, such as Acrobat's "illustration" mode.

To decode the syntax of object labeling in order to gain information on the type of an object for example when clicking on it, translation-functions can be implemented to display a "real world" object name such as "Fire Damper 01/35" instead of "GR711_MT16855_SY01_SE0" by depositing corresponding string arrays on the JavaScript level. Thus, not only the interdisciplinary communication is facilitated by unique and comprehensible object identification, it also offers opportunities to search, isolate or focus the camera to specific model components. Especially when handling more complex geometry such as MEP elements as a facility manager or highly detailed facades as a constructor, this can be very handy.

On the document level of a 3Dpdf, every type of Acrobat's interactive form fields (buttons, radio buttons, checkboxes, etc.) can be used to call any JavaScript function. For example, unchecking a checkbox called "Display Walls" could both execute the according JavaScript function and deliver the state value (on or off) into it. For extended options, for example to display the walls either transparent, on or off, radio buttons can be used accordingly. Additionally, it is possible to create such form fields (i.e. menu items in the user interface) based on the 3D content: A JavaScript detects based on the syntax of the object names, how many stories ("SY01","SY00",…) are in a model, an then creates an according set of buttons allowing to isolate, hide or show each of these stories.

4. CONCLUSIONS

Once decided on a target structure and syntax for classifying and naming a model's elements for a subsequent treatment in an interactive visualization interface, the challenge is to reduce the time-consuming processes of reorganizing and enriching the heterogeneous input data from all stakeholders. Employing 3DsMax as one of the market-leading and utterly capable visualization software to both collect all desired input data in (almost) every common 3D file format and from any given folder structure and then to label and classify all model elements according to the target syntax, it becomes possible to assemble integrated project models in shortest time. The only preparation needed to adopt these automation functionalities to a new project's stakeholder constellation is the customization of a single

translation-chart for every stakeholder's CAD structure and file system once at project start. Once brought together and labeled, the model components can now be automatically enriched by graphical information such as textures or mapping coordinates and any additional chart-based alphanumeric information. Based on the individual demands of all stakeholders, the model components can be assorted into new assemblies and afterwards be converted into 3Dpdf.

JavaScript-empowered pdf functionalities based on the imported model labeling syntax along with custom interactive interfaces assembled in pdf templates then allow users to explore the model content: Isolating, highlighting or hiding large groups of building components, floor levels or individual structural elements interdependently through scripted functions along with the Acrobat's basic functionalities to navigate and display models offer users an enormous range of opportunities to easily visualize any BIM content by just opening a pdf in Acrobat Reader. Besides, the estimated average file size for a single story, fully textured and mapped 3Dpdf in a large-scale project containing all interdisciplinary 3D content ranges between five and ten megabytes.

By introducing this method of producing integrated interactive BIM visualization models, we aim to facilitate the access, visualization and communication of even highly complex geometric content in today's interdisciplinary planning process and would also like to illustrate a way to document these data for the use throughout a building's life cycle.

REFERENCES

Adobe 3Dpdf, 2005, Acrobat Release 7, Available at: www.adobe.com/aboutadobe/pressroom/pressreleases/200501/010505Acrobat.html [accessed 28 April 2012]

Adobe Actionscript, 2012, Actionscript in 2D and 3Dpdf. Available at: www.adobe.com/devnet/actionscript.html [accessed 28 April 2012]

Adobe JavaScript, 2012, JavaScript in 2D and 3Dpdf. Available at: www.adobe.com/devnet/acrobat.html [accessed 28 April 2012]

Arayici Y., Coates P., Kiviniemi A. , Koskela L. and Kagioglou M., 2011. BIM implementation and Adoption Process for an Architectural Practice. *FIATECH Conference,* USA

Autodesk, 2012, Revit Architecture, Available at: http://usa.autodesk.com/revit/architectural-design-software [accessed 28 April 2012]

Building Smart, 2012. Available at: http://buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary [accessed 28 April 2012]

Cicmil. S, Marshall D.,2005. Insights into collaboration at the project level: complexity, social interaction and procurement mechanisms. BUILDING RESEARCH & INFORMATION Vol. 33(6), pp. 523–535

Dossick C., Neff G. 2010. Messy Talk and Clean technology: Requirements for Inter-organiaztional Collaboration and BIM Implementation within AEC Industry. Working Paper Proceedings, *Engineering Project Organizations Conference*, Lake Tahoe CA

Ecma International, 2007, Standard ECMA-363 Available at: www.ecma-international.org/publications/standards/Ecma-363.htm [accessed 4 April 2012]

Graphisoft, 2012, ArchiCAD 15, Available at: www.graphisoft.com/products/archicad [accessed 28 April 2012]

IFC Viewer 1, 2012, Nemetschek IFC Viewer. Available at: www.allplan.net/cms/home/download0/downloadifcviewer.html [accessed 15 April 2012]

IFC Viewer 2, 2012, DDS CAD Viewer. Available at: www.dds-cad.de/123x5x0.xhtml [accessed 28 April 2012]

IFC Viewer 3, 2012, Solibri model checker. Available at: www.solibri.com [accessed 05 January 2012]

McGraw Hill Construction, 2010,' The Business Value of BIM in Europe', Available at: http://bim.construction.com/research/FreeReport/BIM_Europe [accessed 28 April 2012]

Nemetschek, 2012, Allplan 2012, Available at: www.nemetschek.eu [accessed 28 April 2012]

Owen R., Amor R., Palmer M., et al, 2010. Challenges for Integrated Design and Delivery Solutions. *Architectural Engineering and Design Management*, Vol. 6, pp. 232-240

QuadriSpace, 2012, 3D pdf converter. Available at: www.quadrispace.com/products/pages3d/index.htm [accessed 28 April 2012]

Plume J., Mitchell J.,2007. Collaborative design using a shared IFC building model—Learning from experience. *Automation in Construction,* 16 pp. 28 – 36

Right Hemisphere, 2012, Deep Exploration 3D pdf converter. Available at: www.righthemisphere.com/products/dexp [accessed 28 April 2012]

Sacks R. , Kaner I., Eastman M.C., Jeong Y-S., 2010, The Rosewood experiment — Building information modeling and interoperability for architectural precast facades. *Automation in Construction,* Vol. 19, pp. 419–432

Tetra4D, 2012, 3D pdf converter. Available at: www.tetra4d.com/3d-pdf-converter.html [accessed 28 April 2012]