Building Information Modeling and Virtual Reality: Workflows for Design and Facility Management

Zhan Yang, Karen Kensek

University of Southern California, Los Angeles, CA

ABSTRACT: Building information modeling (BIM) is used throughout a building's lifecycle from design to operations and maintenance (O&M). Virtual reality (VR) can be used for real-time simulation of a user's presence in a 3D interactive environment. Three workflows of BIM based VR were developed and applied towards design review and facility management (FM) for a patient room: (1) cinematic VR; (2) Revit + Unity; and (3) Revit + Fuzor. In the first workflow, a VR film was made with Revit, Maya, Mettle Skybox, Handbrake, Redshift, and After Effects to present the project through an Oculus Rift HMD for a semi-interactive immersive VR experience. In the second workflow, parameter data including the equipment's manufacturer, cost, and website was added to a project file in Revit. The geometry was exported to Unity using Maya as a go-between. The data was extracted from the Revit schedule to a text file. In Unity, the data was parsed based on a unique identifier and linked back to the equipment through C# scripting based on a panel based user interface of Unity. Through customized programming, the outcome was a room in VR where equipment information appeared on canvas panels. The third workflow used Revit, a database of the equipment in Excel, Fuzor, and a customized link written in Python using the Fuzor Application Program Interface (API). Fuzor was chosen as it is a game engine based solution specifically tailored for the building industry. An XML tool was developed to link an Excel sheet to the Revit and Fuzor models. This tool has immediate application potential because the Excel file can be any kind of database in a real project like a maintenance or repair logging schedule.

KEYWORDS: Building Information Modeling (BIM), Virtual Reality (VR), 3D Game Engine, Integrated Facility Management (IFM), Design Review of Healthcare Facility

INTRODUCTION

Virtual Reality (VR). Virtual reality (VR) is a virtual environment generated as a 3d digital model and accessed through multiple hardware platforms including stereoscopic goggles. A viewer's actions in the real world are tracked and reflected into that simulated 3D environment (Brouchoud, 2016). VR offers a visualization method to help people intuitively understand the environment in a fully immersive way (Donalek et al., 2014), with little effort (Arch Virtual, 2014). Although VR's characteristics are usually based on a single user's experiences, recently there has been environments created for collaboration and cooperative tasks (Peters et al., 2016). VR can be used to facilitate users to be more aware of a situation (Endsley, 1995), add interactivity for simulating real world, and enrich media content (Klein and Militello, 2001). Immersive collaboration in VR applications can help achieve a high degree of communication and collaboration among multiple players via text chat, voice communication, and interaction with shared components. There is a match between these VR characteristics and the nature of the architecture, engineering, construction (AEC) industry. Nevertheless, cumbersome headset and isolation from the real world are the two big hurdles in VR to be tackled in the future (Virtual Xperience Inc., 2017).

3D Game Engine. A game engine is a computer game application that includes elements such as 3D rendering, physics, collision, graphical user interface, artificial intelligence, sound, and event management (Eberly, 2007; Fritsch and Kada, 2004). A game engine enables the user to manipulate the virtual environment to achieve real-time control with more advanced software allowing multi-user collaboration. Unity is a commonly used 3D game engine. Its asset store, containing a library of high quality assets (pre-made objects, materials, animation, special effects), reduces development time and cost (Gaudiosi, 2016). Designed for the AEC industry specifically, Fuzor is a game engine allowing VR functionality and bi-directional synchronization with Revit. It provides a Revit add-on to convert the Revit model into a Fuzor model directly. Any change made in the Revit model can be automatically updated to Fuzor model and vice versa. Apart from the model, there is no need to export and import the BIM data of parameter information between the Revit and Fuzor (Kalloc Studios, 2018). The Fuzor API is the internal protocol that can be used for websites and applications to launch, query, and send commands to the Fuzor; this can also be regarded as an expanded user interface for Fuzor (Kalloc Studios, 2018). The REST (Representational State Transfer) API is the API using HTTP requests to get, put, post, and delete data (TechTarget, 2018). Fuzor REST API is one part of the Fuzor API. Although there are limitations in the Fuzor API and difficulty in use of Unity, the use of an existing software program like Unity or Fuzor makes the development of a workflow relatively easier (Du et al., 2018).

Integrated Facility Management (IFM). According to the International Facility Management Association (IFMA), facility management is a profession that encompasses multiple disciplines to ensure functionality of the built environment by integrating people, place, process and technology (Professional Constructor Central, 2012). Integrated facility management (IFM) is essential in the lifespan of a building project because operations and maintenance (O&M) in IFM occupies the longest period. More than 85% of the total cost can be attributed to maintaining the normal working condition of a building (Teicholz, 2004). When as-built drawings are not available, data re-entry may be necessary for facilities managers; fieldwork inefficiency can be improved up to 20% by offering suitable information support (Lee and Akin, 2009). Around 50% of the total task time is accounted by activities related to documentation including finding the appropriate paper work and then reading it (Teicholz, 2004). Both the processes are time consuming because the building operating data is often not digitally associated with the building model; static information including annotated drawings, manuals, and photos have normally been stored in the form of paper documents (Hou and Wang, 2011).

Design Review of Healthcare Facility. Healthcare facilities are specialized spaces with critical consideration for patient safety and comfort. The design needs feedback from clients, architects, engineers, technicians, facility managers, healthcare staff, and even future patients. This can be difficult to accomplish as some of the stakeholders mentioned previously have different skills in interpreting architecture drawings. Yet, still, design review is important (Fu and East, 1999; Shiratuddin, 2009). Healthcare quality including the staff fatigue and effectiveness in delivering care, patient safety, and stress and recovery is strongly related to the physical environment (Ulrich et al., 2004). Creating a better physical environment can be enhanced through experience based design (EBD) for the initial and detailed design reviews in providing better solutions for visualization, information management, and collaboration. EBD improves feedback quality in design reviews. It can help reveal architectural errors, find incompatibilities between an equipment location and other physical constraints, and establish better design solutions in the early project stage. Done digitally (for example in a VR environment), it might be able to reduce costs versus building a full-size mockup of a room, which is a standard practice for expensive and complex rooms. For example, there are certain tasks that can be benefited from EBD: assessing the mobility of equipment and activity limitations of occupants, configuring indoor facilities like patient beds to different forms, and inspecting indoor dimensions. Through simulation, one can also evaluate architectural characteristics for infection control and compare different light sources (Kumar et al., 2011; Dunston et al, 2007).

Integration. BIM enabled FM has been utilized to enhance the communication in a more intuitive presentation by combining the model and data together after supplementing the FM related information in addition to the original BIM database. The addition of a multi-user environment via the game engine has solved the shortcoming of the BIM based VR solution which is lack of the communication (Shi et al., 2016). Another researcher has developed BIM to VR data synchronization with custom programming (Du et al., 2018). Training and design review for healthcare facilities can also be benefited from game engine based VR environment (Kumar et al., 2011). In addition, simulation and training purposes can both be satisfied by a special framework integrating BIM and the game engine (Wu and Kaushik, 2015). Based on the integration of all these elements in the literature review, three workflows using BIM and VR were developed for the main work of this paper, which were cinematic VR, Unity based interactive VR, and Fuzor based interactive VR.

1. WORKFLOW 1: CINEMATIC VR

Autodesk Maya was used as the main platform for integrating all the video ingredients, starting with basic models exported from Revit and ending with a cinematic VR file (Fig. 1).

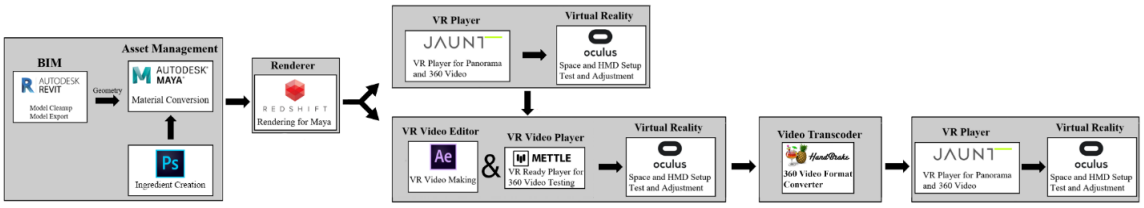
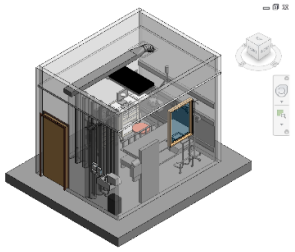


Figure 1: Workflow 1 of cinematic VR

All the information panels were made from Adobe Photoshop (PS). Redshift within Maya was responsible for rendering all the frames, followed by making a linear video from Adobe Effects (AE). AE was used to combine all the frames into a video with another round of rendering. AE was also for creating effects for better transitions between frames. The frame output from Maya could be rendered in either mono or stereo mode, the stereo mode being especially time-consuming for rendering. A single frame was rendered first. The test procedure

was conducted via Jaunt Player (shown at the top right corner of the workflow diagram). Mettle, an AE plugin, was used to play the VR video after connecting to the Oculus Rift. It was also used to preview the video from AE before the final AE rendering. Jaunt Player is a standalone VR player and can also be used to play and present the final VR video either through Oculus Rift terminal or computer screen.

A generic model of the patient room from MILSTD1691 was used (Fig. 2). The data in the Excel database was gathered from the Internet for simulating a real application scenario of IFM. Later, this Excel data could be replaced with real IFM related data like a maintenance schedule, replacement list, warranties, O&M manuals, training videos, and change orders that are not standardly within a Revit model. Test data was applied to three pieces of equipment in the research model: Name, Manufacturer, URL, and Cost of the cart, stretcher, and light (Fig. 3).



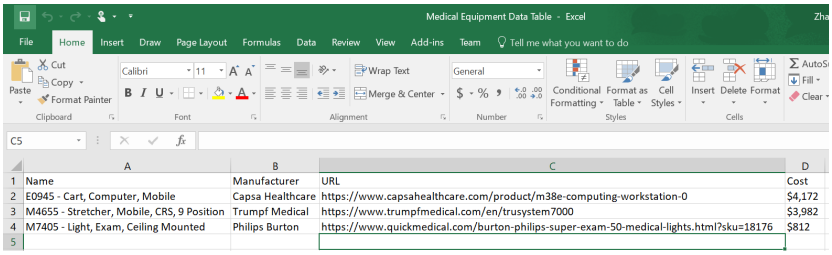


Figure 2: The modified Revit model of a patient room based on the EPS OOHR Template (Military Standard 1691, 2017)
Figure 3: The Excel database of the three pieces of equipment within the research model

After adding materials, establishing lighting, adding pan and render cameras for key frames, creating additional frames for scene changes, and setting up surrounding environment for viewing, the final footage was rendered. The frames were automatically linearly interpolated within Maya. There was a total of 1400 frames rendered via Redshift, with a resolution of 3840 × 1920 dpi. Adobe After Effects, using Mettle SkyBox VR, was used to composite the whole image sequence and produce the final rendering (Fig. 5). Mettle was also used to preview the VR video before the final rendering process was initiated. After transcoding the final raw video file through HandBrake, the final video was played via Jaunt Player. The information panels for the three pieces of data associated with the objects and the space limitation detected appeared in the final video (Fig. 4). Currently, this type of VR in the form of the linear film cannot achieve real-time interaction.

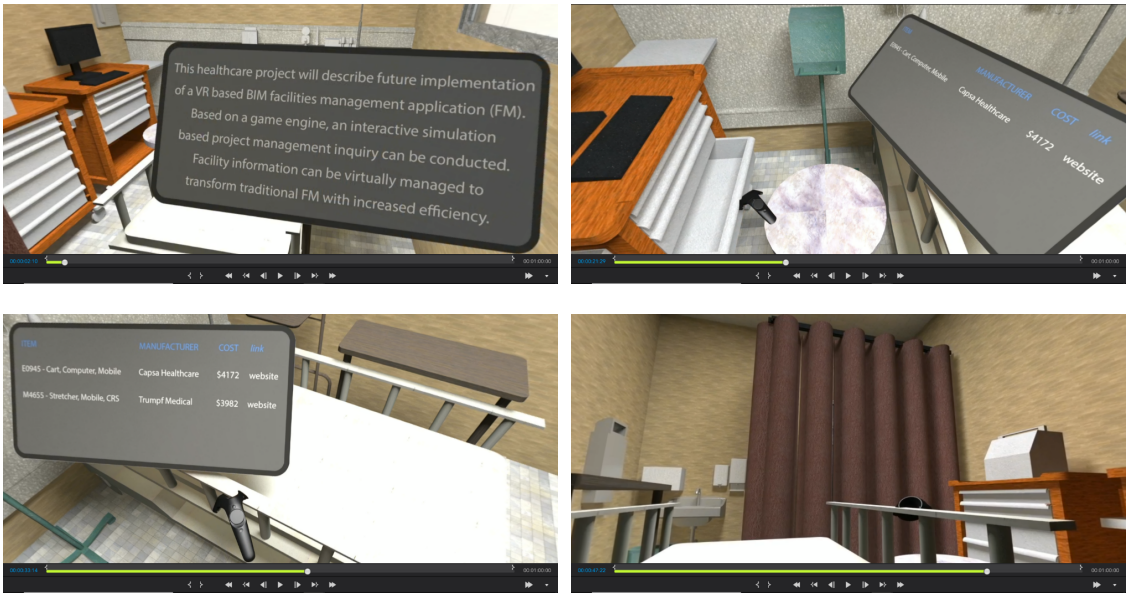


Figure 4: Four example frames of the final VR video played by the Jaunt Player

2. Workflow 2: Unity Based Interactive VR

The Unity based interactive VR environment was created for design review of an architectural space and for data visualization in an environment suitable for IFM (Fig. 5). Both the geometry and data attributes in BIM

were exported from Revit to Unity with Maya as the transition tool. VRTK (Virtual Reality Toolkit) and Steam VR were both necessary for this workflow. VRTK was a free tool used to facilitate the VR game development through Unity, allowing the players to play the developed game through a mouse and keyboard without needing the head mounted display (HMD); this can save time and effort in development. Steam VR was used to support the HTC Vive (a type of HMD) connecting to a PC.

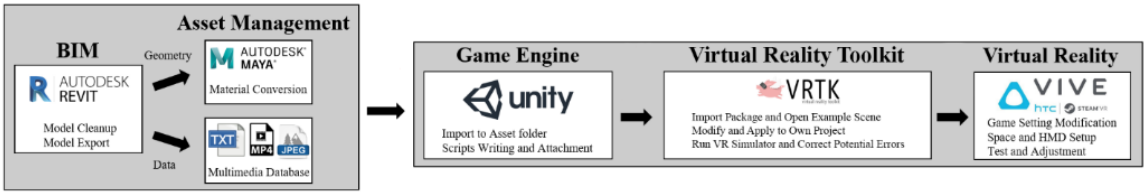
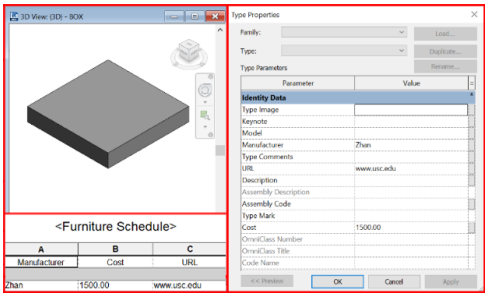


Figure 5: Workflow 2 of Unity based interactive VR

In addition to the geometry, Revit was also used to create a equipment schedule based on the family parameters. The schedule was then assembled into a database stored in the Unity Asset folder. A text file format was selected as the database format with a specific data structure including the semicolon as field delimiter and the quotation mark as text qualifier. A simple box furniture was created first for testing with the three family parameters of Manufacturer, Cost, and URL (Fig.6). Since the entire schedule contained the facility information of more than one piece of equipment and the display for any piece of the equipment should only extract its own specifically related information, the solution was to parse and extract the data from that schedule (Fig. 7). Since the name of each equipment in Revit was repeated twice in the name of the Unity model within the FBX group, the first five numbers were utilized as the unique marker instead of the whole name in the data parsing (Fig. 8).



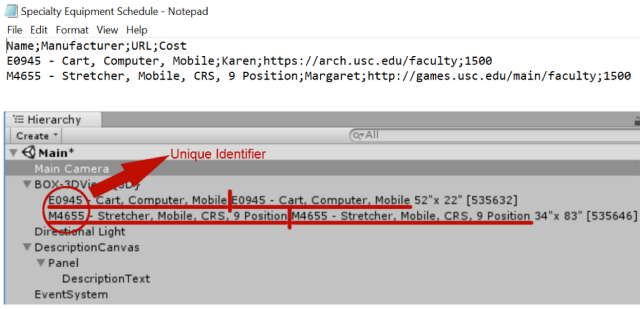


Figure 6: The Autodesk Revit interface showing the sample box furniture model with its type properties (left)
Figure 7: The exported specialty equipment schedule (upper right)
Figure 8: The names of FBX imported model pieces in Unity hierarchy window (lower right)

The VRTK class of C# code was used to convert the mouse input to the VR controller input, which could be "laser pointing" to use object or directly touching the object. Initially, the VR simulator was used to show the final VR environment, which could achieve almost the same effect as that of full VR except for the perception and the presence of the controller model in the scene.

This type of VR model could be used for both design review and O&M. In a complex project like a medical room, problems could be easily detected through VR simulation regarding space layout and utilization. The piece of equipment could be programmed to only perform as it would in a real-world environment with respect to gravity, collision, and movement (e.g. a door or a cabinet might only be able to open in a certain amount in accordance to the technical specifications). This part of the workflow was accomplished for a drawer in a cabinet showing it clashing with a chair when open. This was programmed in Unity, via VRTK SDK. However, the transfer from new Revit parameters (for example, the depth the drawer is physically possible to open) to VR object behaviors could not be easily automated and did not become part of this workflow.

The interactive data visualization for IFM was achieved using a C# script developed to place the data attributes from Revit into the VR environment (Fig. 9). In this script, the inheritance was essential to shift the mouse click control to VR control using VRTK. To be specific, the class of Information_Visualization was inherited from the class of VRTK_InteractableObject so that the behavior of VRTK_InteractableObject class could be reused, extended and modified by the new class of Information_Visualization (Microsoft, 2015).

Figure 9: The C# script for data visualization in VR

In the result, the related data was overlaid on top of the VR screen per laser point or controller touch (Fig. 10).

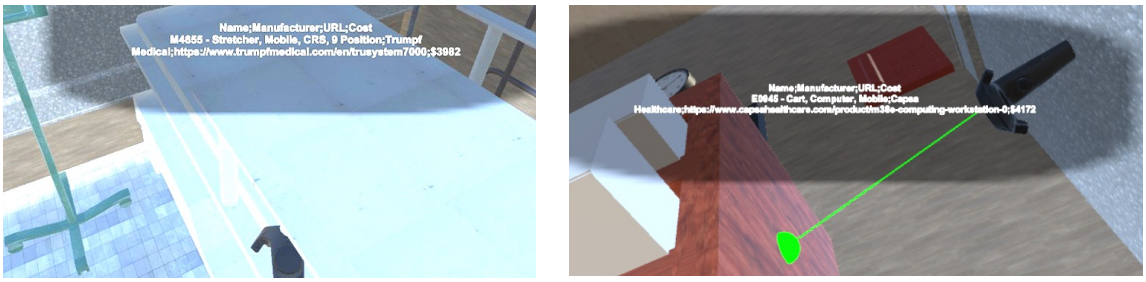


Figure 10: Sample data visualization in the Unity based interactive VR through HTC Vive

3. Workflow 3: Fuzor Based Interactive VR

In the third workflow, the inherent capability of Fuzor in displaying the Revit parametric information in VR was demonstrated and enhanced (Fig. 11). A tool was developed to automate the integration between the Excel sheet and the Fuzor model instead of using the internal Revit parameters as in the second workflow. This tool has potentially many applications in FM because the Excel file can be changed to an IFM database like a maintenance schedule or repair logging document. The tool is a customized Python script to parse the external data from Excel and execute the Fuzor API to add new parameters to Fuzor.

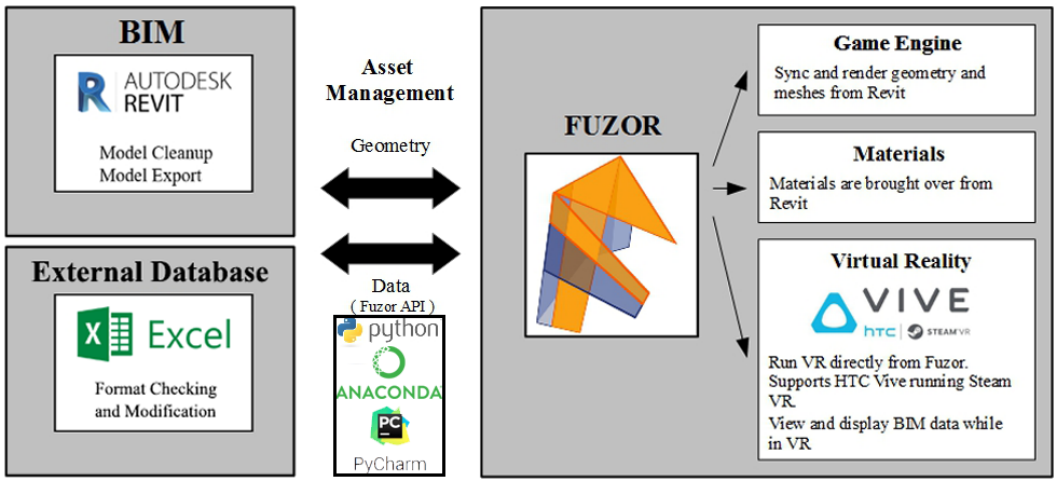


Figure 11: Workflow 3 of Fuzor based interactive VR

In the first test of workflow 3, materials were not yet assigned. Inherent in Fuzor and used with Revit were the following applicable features: any Revit family can be added, deleted, moved, and rotated; property data from the Revit family parameters can be visualized; dimensions can be measured in VR; and any issue can be

marked up in VR and saved so that later it can be seen by the other reviewers and modified accordingly (Figs. 12 and 13)

Figure 12: Fuzor VR inherent functionality examples
Figure 13: Close up of the object properties

Expanding the Fuzor's database through connecting to an external data warehouse for IFM involved creating a custom script. The database was the same as the original Excel datasheet with one difference - a new column for the object ID number was added for Fuzor API identification. The structure of the data was critical for parsing via the new script. The parameter label was at the top of each column: Name, Manufacturer, URL, and Cost. The parameter value was the cell value of the Excel whose value was determined by both the value of the first column of ID and the parameter label. Only one pair of the data could be added to Fuzor at a time. One pair of data consisted of a parameter label and its corresponding parameter value for each object identified by its unique identifier which was ID number for this case.

The Fuzor API is XML and Web based. It uses the uniform format of http://localhost:45190/query?parameter_list. One can query data from the Fuzor model using a "query" command (e.g. http://localhost:45190/selected). Another typical action performed is an "execute" where the API will be used to add parameters.

The process included two parts: extracting Excel data and exporting the data to Fuzor using REST calls. First, the object whose parameter was to be updated from the external Excel file was selected, followed by running the first Fuzor REST API. After running, all the information was shown in a new webpage in XML. The object ID was under the line of <name>__fuzor_objectid</name> that was also the first value in that structure. The next step was to run the second API in the format of http://localhost:45190/execute?createcustomparameter=<ObjectID>:<Parameter_Label>:<Parameter_Value>. For instance, if the cost information of that stretcher was the data to be added, the command was http://localhost:45190/execute?createcustomparameter=13931801347516435145:Cost:3982. After execution, the new BIM parameter was added automatically. All the steps were included in a Python script to automate the entire process of parsing key information out of the Excel document, handling the REST calls, and sending the information back to Fuzor in real time (Fig. 14).

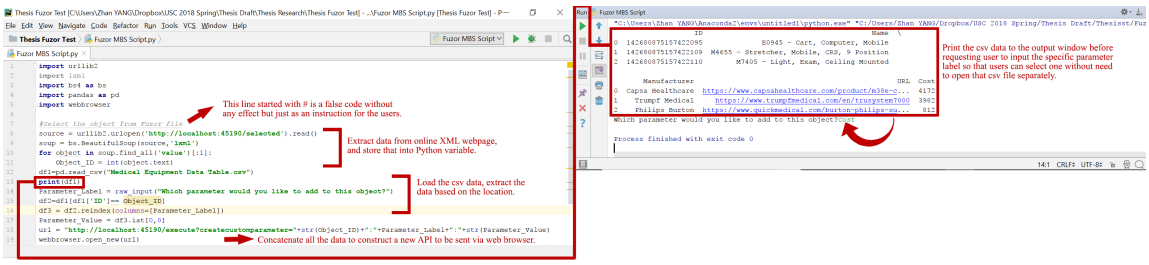


Figure 14: The Python script for automating extracting the Excel data and exporting the extracted data to Fuzor using REST Calls (shown in PyCharm)

The result of adding new parameters is shown in the red block, from which it can be found that the customized BIM parameters can also be added and deleted through the UI manually (Fig. 15). A new window within the yellow boundary is shown after clicking the Annotation button inside the window within the blue block. All the IFM related information can then be typed in (Fig. 16). This includes typical IFM data like the work order category, work description, and deadline. A markup image can be added as a separate attachment. With this information and the essential BIM data added previously, the basic IFM goal can be achieved.

Figure 15: The final result of the workflow 3. The yellow window can contain additional user added data.
Figure 16: Ability to add O&M data

The corresponding O&M scenario can be illustrated as follows. From all the markup images collected inside the Fuzor database folder, problems are identified for the specific equipment or its component. Customized parameters are used to supplement the necessary information for the life cycle of the project, important information for repair or maintenance can be retrieved (e.g. serial numbers, warranty information, and the O&M history of all the components inside the building). Then the work order can be established in the Annotation window, and the status of that work order can be constantly tracked within the Fuzor system until the issue is fixed. This scenario can be realized under the workflow described. If the level of development of the BIM is high enough, Fuzor can be also used as an advanced IFM tool with VR capability, and its model can be served as a living file which owns a precise snapshot of the full space.

DISCUSSION AND CONCLUSION

Three workflows are described for using VR for design review and facilities management: cinematic VR, Unity based interactive VR, and Fuzor based interactive VR. The first workflow creates an immersive experience for the user to comprehensively understand the environment. This workflow requires knowledge and access to many software programs including Autodesk Maya and Jaunt Player. Although no programming is needed, its development is time consuming and work intensive. While the final project provides only an animation path pre-determined for the user, it does allow for some interaction in the form of panoramic views. The lack of full interactivity in the VR environment is a weakness of this workflow.

The second and third workflows can both be used for both design review and facility management. Both workflows utilize a game engine based platform to achieve real-time control and accomplish interactive data visualization of the IFM data based on the existing BIM data. They can be used to transfer the needed information from the Revit model, in this example a patient room in a healthcare facility, to the VR model. Their main differences are the base software used. The Unity-based workflow allows for a high degree of customization and interactivity within the VR environment. However, this involves C# programming and the ability to use VRTK SDK. The third workflow with Fuzor, due to its bi-directional synchronization with Revit and easy access to a VR environment, has many features built-in that makes it easier for implementing a FM solution. However, the Fuzor API currently is relatively limited for developing customized applications.

Virtual reality provides an immersive environment. With additional data, it can also be used for better informed design and facilities management. Building information modeling is widely used in the building industry, and its files can provide both geometry and data. With some scripting, a game engine can be used to bring and link these together. Additional external data can augment original BIM data. Eventually, a complete database can be managed in the VR scene, which can benefit integrated facilities management. The proposed workflows can connect the 3d model and data more closely so that the virtual space can be transformed from a replica of a real project to a living project document suitable for both design and facility management.

ACKNOWLEDGEMENTS

Yang wishes to thank the other members of his committee Margaret Moser, Eric Hanson, and Chris Nguyen for their help in developing these ideas.

REFERENCES

Arch Virtual. 2014. *Learn How Arch Virtual Uses Unity to Visualize the Design Projects of Their Clients*, [online] Available at https://unity3d.com/showcase/case-stories/arch-virtual.

Brouchoud, J. 2016. *VR Device Options for Architects*, [online] Available at http://www.hypergridbusiness.com/2016/06/vr-device-options-for-architects/.

Donalek, C., Djorgovski, S. G., Cioc, A., Wang, A., Zhang, J., Lawler, E., Yeh, S., Mahabal, A., Graham, M., Drake, A., Davidoff, S., Norris, J. S., and Longo, G. 2014. *Immersive and Collaborative Data Visualization Using Virtual Reality Platforms*, 2014 IEEE International Conference on Big Data.

Du, J., Zou, Z., Shi, Y., and Zhao, D. (2018). *Zero latency: Real-time synchronization of BIM data in virtual reality for collaborative decision-making*. Automation in Construction, 85, 51-64.

Dunston, P. S., Arns, L. L., and McGlothlin, J. D. 2007. *An Immersive Virtual Reality Mock-Up for Design Review of Hospital Patient Rooms*, Purdue University.

Eberly, D. 2007. *3D Game Engine Design: A Practical Approach to Real-time Computer Graphics*, 2nd ed. San Francisco CA; Oxford, Morgan Kaufmann; Elsevier Science.

Endsley, M. R. 1995. *Toward A Theory of Situation Awareness In Dynamic Systems*, Human Factors: The Journal of the Human Factors and Ergonomics Society, 37(1):32–64, 1995.

Fritsch, D. and Kada, M. 2004. *Visualization Using Game Engines,* ISPRS Commission 5. Istanbul, Turkey.

Fu, M. C. and East, E. W. 1999. *The Virtual Design Review*, Computer-Aided Civil and Infrastructure Engineering, 14, 25-35.

Gaudiosi, J. 2016. *Why Valve's Partnership with Unity Is Important to Virtual Reality*, [online] Available at http://fortune.com/2016/02/11/valves-partners-with-unity/.

Hou, L. and Wang, X. 2011. *Experimental Framework for Evaluating Cognitive Workload of Using AR System in General Task*, in: Proceedings of 28th International Symposium on Automation and Robotics in Construction, Seoul, Korea.

Kalloc Studios. 2018. *Fuzor, Got AEC Apps: Design, Build, Succeed*, [online] Available at https://www.kalloctech.com/index.jsp.

Klein, G. and Militello, L. 2001. *Some Guidelines for Conducting a Cognitive Task Analysis*, Advances in Human Performance and Cognitive Engineering Research, 1,163-199.

Kumar, S., Hedrick, M., Wiacek, C., and Messner, J. I. 2011. *Developing an Experienced-Based Design Review Application for Healthcare Facilities Using A 3d Game Engine*, Journal of Information Technology in Construction, 16, 85-104.

Lee, S. and Akin, Ö. 2009. *Shadowing Tradespeople: Inefficiency in Maintenance Fieldwork*, Automation in Construction 18 (5) (2009) 536–546.

Microsoft. 2015. *C# Programming Guide*, [online] Available at https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/index

Military Standard 1691. 2017. *Product Search Page*, [online] Available at https://ms1691.facilities.health.mil/milstd1691/#/.

Peters, E., Heijligers, B., Kievith, J., Razafindrakoto, D, X., Oosterhout, R. V., Santos, C., Mayer, I., and Louwerse, M. 2016. *Design for Collaboration in Mixed Reality,* Technical Challenges and Solutions 2016; 978-1-5090-2722-4.

Professional Constructor Central. 2012. *IFMA 2012 – World Workplace and BIM*, [online] Available at https://buildinginformationmanagement.wordpress.com/2012/11/02/ifma-2012-world-workplace-and-bim/#.

Shi, Y. M., Du, J., Lavy, S., and Zhao, D. 2016. *A Multiuser Shared Virtual Environment for Facility Management*. Procedia Engineering 145, 120 – 127.

Shiratuddin, M. F. 2009. *A Framework for Design Review in a Virtual Environment: Using Context Aware Information Processing*, ISBN-10: 363917285X, ISBN-13: 978-3639172850, Publisher: VDM Verlag.

TechTarget. 2018. *RESTful API*, [online] Available at http://searchmicroservices.techtarget.com/definition/RESTful-API.

Teicholz, E. 2004. *Bridging the AEC Technology Gap,* IFMA Facility Management Journal, [online] Available at http://www.bricsnet.com/content/Teicholz.pdf 2004.

Ulrich, R., Quan, X., Zimring, C., Joseph, A., and Choudhary, R. 2004. *The Role of the Physical Environment in the Hospital of the 21st Century: A Once-in-a-Lifetime Opportunity*, funded by the Robert Wood Johnson Foundation.

Virtual Xperience Inc. 2017. *CBS – Virtual Reality Check*, [online] Available at https://www.virtual-xperience.com/cbs-virtual-reality-check/.

Wu, W., and Kaushik, I. 2015. *Design for Sustainable Aging: Improving Design Communication through Building Information Modeling and Game Engine Integration*. Procedia Engineering, 118, 926 – 933.