Creative Construction Conference 2019, CCC 2019, 29 June - 2 July 2019, Budapest, Hungary

Using Dynamo for Model-Based Delivery of Facility Asset Data

Walid Thabet^{a,*}, Jason Lucas^b

^aVirginia Tech, Blacksburg, Virginia 24061, USA

^bClemson University, Clemson, South Carolina 29634, USA

Abstract

Large facility owners are becoming increasingly aware of the value and advantage of utilizing design and build teams to capture and deliver facility asset data necessary to populate their Computerized Maintenance and Management System (CMMS) for efficient operation and maintenance of their facilities. Facility asset data can be delivered using various formats including a spreadsheet-based or a model-based deliverable. In a spreadsheet-based deliverable, asset data are captured and stored using a spreadsheet specifically formatted to allow the end user direct linking of the spreadsheet to the CMMS and automatic upload of the data. In a model-based deliverable, asset data is captured and loaded into a BIM model then transferred directly or indirectly from the model to the CMMS using various data exchange methods.

This paper discusses the indirect linking of facility asset data captured in a Revit BIM model to the facility management system through exporting the data to an external database. Dynamo, an open-source script-programming tool that works within the Revit environment, is used to extract and export the data from Revit to an external SQL database in a specific format and organizational structure that would allow for uploading of the data to the CMMS. Dynamo script (e.g. Python script) was used to export certain data parameters in a specific order and format. Exported asset data parameters and values are saved to the SQL database and linked to the CMMS to support operations and maintenance.

The paper uses a case study approach to illustrate the implementation of Dynamo to a renovation project for a large academic institution. Asset data for the project is captured from project plans and submittals and loaded into a Revit BIM model. The Dynamo script is tested to verify the export of data in the required format.

© 2019 The Authors. Published by Budapest University of Technology and Economics & Diamond Congress Ltd.

Peer-review under responsibility of the scientific committee of the Creative Construction Conference 2019.

Keywords: Asset data; Revit, Dynamo; SQL (standard query language); CMMS (computerized maintenance and management system)

1. Introduction

The capital facility industry continues to suffer from data inefficiencies. The fragmented nature of information exchange and management during design, construction and through facility management causes an estimated annual loss of \$15.8 billion (USD) in the U.S. alone [1]. Efforts focusing on efficient processes and workflows that utilizes BIM continues to be developed to define specific owner needs, capture and transfer life cycle facility data to the owner's computerized maintenance and management systems (CMMS) [2, 3]. Yet research still recognizes that there continues to be gaps and challenges to directly link BIM to FM practices and systems used [4,5]. [6] Proposed a BIM-FM holistic process workflow to define, capture and transfer lifecycle data to a computerized maintenance management

*Corresponding author: thabet@vt.edu

systems (CMMS). The proposed process comprises of six basic steps: identify need, specify data requirements, define format, define transfer mechanism, develop standards, and update CMMS database. The information needs analysis, data requirements, defined handover specifications and format were discussed as part of a case study [7].

Three main methods for data transfer were explored as part of the research investigation and include: (1) utilizing IFC models and schedule data exported to Navisworks and Data Tools to allow for an as-built federated model that can hold structured and unstructured data, (2) the use of the Archibus Extension in Revit to map parameters directly into Archibus, and (3) developing custom script programming within Dynamo to export Revit parameters and required data into the required format so it can be directly imported into the CMMS, AiM. The Navisworks process, though allowing for a complete model with links to structured and unstructured data still required manual manipulation of exported spreadsheet data to transfer to the CMMS. The Archibus Extension option provided a valuable insight but was deemed by the owner to not be a viable option because it would require them to switch to Archibus to manage all facilities.

This paper summarizes the process of using Dynamo to export data from the BIM model (Revit) to an external .csv file. This data file can be converted to an Excel file format or linked to an SQL database to transfer the captured facility information to the CMMS.

2. Case study overview

The case study project consisted of the renovation of mechanical and electrical systems for a 35,000 sqft science and research facility building in an academic institution. Table 1 summarizes asset group information for the project. Asset group name abbreviations used by the academic institution are adopted from the US CAD Standard. Table 2 summarizes twenty (20) pieces of equipment assets captured that are required by the owner to track and maintain. Equipment names are listed as it appear in the plans. Table 2 also shows the parent/child relationship, and the floor/room where each equipment is located.

Table 1. Asset groups considered in the science building renovation.

Asset Group	Description	Asset Quantity	Mechanical	Electrical
AHU	Air Handling Unit	2	X	
ERU	Energy Recovery Unit	1	X	
FAN	Fan	4	X	
FCU	Fan Coil Unit	9	X	
HTR	Heater	4	X	
HUD	Humidifier	1	X	
P	Pumps	4	X	
SEP	Separator	2	X	
TNK	External Tank	2	X	
EMER-ATS	Emergency Transfer Switch	2		X
EMER-GEN	Emergency Generator	1		X

Table 2. Equipment general information.

Asset Group	Equipment Name	Parent	Location
AHU	MAU-1 (Supply Air Handler)		Roof
AHU	EAU-1 (Exhaust Air Handler)		Roof
ERU	Energy Recovery Unit	EAU-1	Roof
FAN	Supply Fan	MAU-1	Roof
FAN	Exhaust Fan	EAU-1	Roof
FAN	F-1		Basement – Mech/Elec Rm 7
FAN	F-2	ERU	Roof
FCU	FC-1 through FC-8		Main Floor
FCU	FC-9		Basement
HTR	HUH-1		Basement – Mech/Elec Rm 7
HTR	ECH-1	ERU	Roof
HTR	ECH-2, ECH-3, ECH-4	MAU-1	Roof
HUD	HUM	MAU-1	Roof
P	HWP-1, HWP-2		Basement – Mech/Elec Rm 7
P	ERP-1	ERU	Roof
P	CWP-1		Basement – Mech/Elec Rm 7
SEP	AS-1		Basement – Mech/Elec Rm 7

SEP	AS-2	ERU	Roof
EMER-ATS	ATS-1, ATS-2		Basement – Mech. Equip. Rm
EMER-GEN	Emergency Generator		Basement – Mech. Equip. Rm

Each asset group has common and specific attributes required for managing and maintaining the asset post-construction. Common attributes are properties that are similar across the different asset groups. , Specific attributes are unique to the asset group. [7] Discussed the institution’s classification and definition of asset attributes and how they support facility management activities.

Common attributes comprises of twenty two (22) properties such as ASSET_TAG, ASSET_GROUP, MANUFACTURE _CODE, MANU_PART_NUMBER, SERIAL NUMBER, WARR_DATE_FR, etc. Specific attributes vary in number for each asset group. Table 3 provides an example of specific attributes for the AHU (Air Handling Unit) asset group.

Table 3. Specific asset attributes for AHU (Air Handler Unit) asset group

AHU Common Attribute		
Bas Point Address	Min. Outside Air	Economizer
Heating Source	Max. Outside Air	Economizer Type
Heating Capacity	Cooling Source	Total Static Pressure
Total Air Flow	Total Cooling Capacity	Type
External Static Pressure	Sensible Cooling Cap	

2.1 Case study model

The renovation project did not have a 3D model available. For the purposes of this research, three (3) basic 3D Revit component models were created: Architecture, Mechanical, and Electrical. The basic wall and room configurations for the spaces involved in the renovation were modeled in the architecture model. Only the assets that were under renovation were added to the mechanical and electrical models. As shown in Figure 1, Both the mechanical and electrical models were separately linked to the architecture model to allow for viewing their respective elements within the context of space and rooms defined in the architecture model. Once parameters and data were loaded into the mechanical and electrical models, the three models were linked together as an overlay reference using “Manage Link” in Revit. The purpose of developing three separate component models was to replicate the process of a typical project where each discipline consultant or trade contractor develop their own model.

Fig. 1. Linking the mechanical and electrical models to the architecture model.

Within each of the mechanical and electrical component models, stand-alone elements that did not have a parent-child relationship (e.g. boilers or fan coil units) were easily modeled from standard Revit family libraries. Elements that include other elements as an assembly with parent-child relationships, such as Air Handling Units (AHU) or Energy Recovery Units (ERU) with fans, had to be modeled such that it graphically appeared as an assembly and, at the same time, it allowed to define parameters for the assembly independent from the components. Modeling these elements as

Revit Assemblies were first considered. All components of an assembly were selected and combined using the Revit assembly tool to create the assembly as shown in Figure 2. This allowed assemblies such as AHU and its components to appear graphically correct and have their own independent parameters; parameters for the assembly and parameters for the components within the assembly (e.g. Return Fan). However, Revit does not distinguish between different assemblies and therefore it does allow for defining customizable parameters for two different assembly types. For example, an AHU assembly and an ERU assembly cannot have their own independent set of parameters assigned to them. Any unique parameter assigned to the AHU assembly through the parameter assignment process will be automatically assigned to all other assembly types defined in the model.

Fig. 2. Creating parent assembly by combining individual components in Revit.

An alternative solution was to use nested families. Families of components of an AHU shown in Figure 2, such as Filter Box, Supply Fan and so on, would be nested to another host component family such as Return Fan to create a group of nested components. Graphically this would accurately represent an assembly and components, however, there are issues with defining parameters of individual components nested to a host component; they do not translate through the host. Parameters of nested components need to be identified through the host element and would not be associated with the individual components.

Finally, though not graphically the best option, families representing the different components of were modeled in the same geometric space representing the assembly as shown in Figure 3. For example, a non-descript block is modeled to represent the location and size of an energy recovery unit (ERU) assembly. This modeled object can then support the attributes and parameters associated with the ERU as the parent asset group. Within the same space, families for detailed components such as fans, pumps, heaters, etc., were placed in the model. Each component being their own family were able to hold the parameters of the child asset.

Fig. 3. Modeling parent-child components in Revit.

2.2 Populating model with required asset attributes and data

To load the common and specific attributes into the mechanical and electrical Revit models, a shared parameter file was created using the Shared Parameter Manager™ module (BIM Management Suite), a 3rd party plugin for Revit from CTC (www.ctcexpresstools.com). CTC BIM Management Suite contains add-ons that work within Revit's

modeling environment to help manage data. The Shared Parameter Manager™ allows the user to easily create and group parameters and load them into the Revit model. The user can then apply the parameters to the appropriate categories, families and assemblies of objects. The plugin is much more robust than using the standard Revit tools to define parameters and allows for batch loading and assigning data to the Revit objects, saving time and minimizing errors. To input parameter values to the different elements in the model, the Spreadsheet Link™ module of the CTC BIM Project Suite was used. Specific categories and parameters are identified within the Spreadsheet Link tool by the user. The user can then enter the values for the parameters within the tool in a user friendly and familiar spreadsheet view. This allows for quick and efficient entry of the data. As soon as it is entered and input into the Spreadsheet Link tool the parameter is updated in the model. The process for loading the shared parameters into the model and inputting values into the model was discussed in further detail in a prior stage of the research [9].

Once parameters and data are loaded into the mechanical and electrical models, a single ‘Integrated’ or group model was created with all the parameters and data attached. This four (4) step process is summarized in Figure 4.

Fig. 4. Linking and binding component models to create an integrated model.

3. Capturing and exporting model data using Dynamo

To transfer the asset data captured in the BIM model to the CMMS platform used by the academic institution, the data had to first be exported to a .csv (Comma Delimited Spreadsheet) file that is then linked to the CMMS. had to export specific format for that data is required. The .csv spreadsheet file must be structured in a specific format to allow the CMMS to recognize the data embedded in the file. The spreadsheet file comprised of six (6) sheets defined in the order listed below:

1. General: lists general properties for each asset.
2. Attribute: lists specific attributes for each asset.
3. Parts: lists replacement part(s) number and quantity.
4. Warranty: lists asset warranty information (description, start and finish dates).
5. Locations Served: lists the locations served by the asset and corresponding usage factor (%) of asset.
6. Classification: Omniclass and Master Format classification numbers for each asset.

The overall process for the Dynamo script is shown in Figure 5 is completed in four (4) main steps. First, data in the “General”, “Parts”, “Location”, “Warranty”, and “Classification” tabs are queried, captured and exported (steps 1, 2, and 3). Second, data in the “Attributes” tab (steps 1, 4, and 3) are queried, captured, processed then exported. The reason the “attributes” tab is processed separately is because the format of the data in that sheet (tab) is structured differently than the other tabs; hence requires additional code.

Fig. 5. Dynamo script workflow.

The following two sub-sections provide examples of the Dynamo script used in steps 1 and 2 in Figure 5 above.

3.1 Identify model elements (step-1)

Figure 7 illustrates step-1 script required to identify all model elements under the different asset groups required under this study and need to be tracked (total of 22 elements). The script in Figure 6 shows three (3) main sub-steps:

Fig. 6. Identifying model elements.

- 1.1 The nodes “Categories” and “All Elements of Category” access all the family instances of a particular category. For the test case, Mechanical Equipment and Duct Accessories were the category names within Revit that contain all the assets that are being tracked. The “List.Join” node compiles lists from each of the inputs that can be queried in future parts of the process. The “All Elements of Category” identifies all elements in the model of the specified categories.
- 1.2 To limit the elements in the list, the parameter “ASSET_TAG” is checked for those assets that have a value. As part of the data documentation and model creation phase, every asset that is being tracked is provided an ASSET_TAG at the time of creation so it can be tracked. The “Element.GetParameterValueByName” and “!=” nodes allow for filtering the list. The Element.GetParameterValueByName outputs a list of ASSET_TAG values. The != node is then used to compare the output list of variables (the X input) to a specific value (Y input). The Y input in this case is set to null (“”), or an empty string. The List.FilterByBoolMask compares the two lists of

elements and masks the elements that have a null value for the parameter. The result is a list of only the elements that have a defined asset tag.

- 1.3 The resulting list from the List.FilterByBoolMask can be used to identify parameter values of a specific element but does not work on type parameters. The FamilyInstance.Type node is used to identify the type parameters of the element. This is required for identifying parameters that are assigned to the element type, such as the “ASSET_GROUP”.

3.2 Querying model parameters (step-2)

The objective of the script for step-2 shown in Figure 7 is to query model parameters in the “General”, “Parts”, “Location”, “Warranty”, and “Classification” tabs and create appropriate lists of those values that can be written to the .CSV file (completed later in step-3). The script in Figure 8 shows two (2) main sub-steps:

- 2.1 Identify the values of specific parameters for each tracked element. The “Element.GetParameterByName” node utilizes two inputs. Depending on the parameter type the “element” input is a list. The “ParameterName” input is a string of the parameter’s name. The result is an indexed list of all parameters of the queried elements.
- 2.2 Each output of the “Element.GetParameterByName” node is then added to a list using the “List.Create” node. These are created by the tab of the csv spreadsheet and organized in the required order.

Fig. 7. Identifying parameter values.

4. Conclusion

The developed process allows for collecting facility management related data throughout the project lifecycle within a Revit model and automating the process of exporting that data for use during facility management. The process helps the owner address the issues of timeliness and efficiency of information transfer to make data usable at the end of the construction process. The data exported through this process can be directly imported into the current CMMS system of the owner. This process replaces a tedious manual process and cuts down on time and potential for error during the data transfer process.

To validate the developed process the output spreadsheet was checked for usability with the CMMS and compared to a manually developed spreadsheet. The data values that were extracted were validated as correct and extraction process took a fraction of the time to complete.

At this time, a limitation exists in that the designed Dynamo approach only allows for data transfer in one direction. Data is extracted from the BIM model and is used to update the CMMS however data changed in the CMMS is not reflected back into the BIM. This would have to be done manually or through further developing the coding. The authors believe that additional development can allow the data exchange process to be bi-directional and will explore it as part of future research. Future research will examine the utilization of a SQL database and Dyanmo programming to serve as a bidirectional link between Revit and the CMMS. SQL will remove a current manual step of importing the spreadsheet as the CMMS works directly off a SQL database. This will require slight programming change but allow for maintaining the geometric representation of the building with updated asset information. This would also serve as a continuous record. This can also then link to other unstructured data into the model such as owner's manuals and relevant PDFs. These forms of unstructured data are not currently addressed in the developed process. The current development is also limited to the current assets that were used in the case study, however, the framework for extraction can be used to quickly add additional assets that the facility owner will be tracking.

References

- [1] Gallaher, M. P., O'connor, A. C., Dettbarn, J. L., Jr., and Gilday, L. T. (2004). Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry (NIST GCR 04-867), National Institute of Standards and Testing (NIST), Gaithersburg, Md. <https://doi.org/10.6028/NIST.GCR.04-867>
- [2] Cavka, H.B., Staub-French, S., and Poirier, E.A. (2017) "Developing owner information requirements for BIM-enabled project delivery and asset management" *Automation in Construction*, 83(2017):169-183. <https://doi.org/10.1016/j.autcon.2017.08.006>
- [3] Sadeghi, M., Mehany, M., and Strong, K. (2018) "Integrating Building Information Models and Building Operation Information Exchange Systems in a Decision Support Framework for Facilities Management", *Construction Research Congress 2018*, ASCE Reston VA, 770-779. <https://doi.org/10.1061/9780784481295.077>
- [4] Edirisinghe, R., London, K.A., Kalutara, P., and Aranda-Mena, G. (2017) "Building information modelling for facility management: are we there yet?" *Engineering, Construction and Architectural Management*, 24(6):1119-1154, <https://doi.org/10.1108/ECAM-06-2016-0139>.
- [5] Miettinen, R., Kerosuo, H., Metsala, T., and Paavola, S. (2018) "Bridging the life cycle: a case study on facility management infrastructures and uses of BIM", *Journal of Facilities Management*, 16(1):2-16, <https://doi.org/10.1108/JFM-04-2017-0017>.
- [6] Thabet, W., Lucas, J., and Johnston, S. (2016) A Case Study for Improving BIM-FM Handover for a larger Educational Institution, in *Proceedings of Construction Research Congress 2016*, ASCE, Reston, VA. DOI/10.1061/9780784479827.217
- [7] Thabet, W. and Lucas, J. (2017a) Asset Data handover for a Large Educational Institution: Case-Study Approach, *J. Construction Engineering and Management*. 143(11): 05017017. DOI/10.1061/(ASCE)CO.1943-7862.0001389
- [8] Thabet W. and Lucas, J. (2017b) A 6-Step Systematic Process for Model-Based Facility Data Delivery, *Journal of Information Technology in Construction (ITcon)*, 22: 104-131. <http://www.itcon.org/2017/6>.