Case Study **Open Access**

Walid Thabet[1,*], Jason Lucas[2], Sai Srinivasan[3]

Linking life cycle BIM data to a facility management system using Revit Dynamo

DOI 10.2478/otmcj-2022-0001
Received May 12, 2020; Accepted June 16, 2020

Abstract: Facility management (FM) requires multidisciplinary activities, and thus has extensive information requirements. Much of that information is created during the design, construction, and commissioning phases of a project. Providing the owner of a facility with usable life cycle asset information after construction has been a challenge to the industry. Traditional methods of manually inputting data into FM systems are time consuming and error prone. Various automated approaches and workflows continue to be developed to respond to specific owner needs. This research developed a unique workflow that uses Dynamo within Revit to automatically extract asset management data from the model and export the data to a proprietary format required by the facility owner. The formatted spreadsheet allows for direct linking of the data to the owner's FM system, hence eliminating time wasted in manual data entry and avoiding missing any maintenance cycles that would result if the FM system is not populated with critical information in a timely manner. This article utilizes a case study approach to demonstrate this novel Dynamo workflow. The required case study asset data identified and captured include asset groups, their properties and attributes, and corresponding metadata. A basic three-dimensional representation of the facility and all its equipment are modeled in Revit and asset data are input to corresponding model elements. This article also describes the complexity of the owner's proprietary information needs and the resulting automated workflow that extracts and exports data from Revit into an Excel format that can directly link into the FM system.

***Corresponding author: Walid Thabet,** 410A Bishop-Fabrao Hall (0156), Department of Building Construction, 1345 Perry St, Virginia Tech, Blacksburg, VA 24061, USA, E-mail: thabet@vt.edu
Jason Lucas, 2-136 Lee Hall, Nieri Family Department of Construction Science and Management, Clemson University, Clemson, SC 29640, USA
Sai Srinivasan, Breigan Concrete Constructors, Raleigh, NC, USA

Keywords: facility management, BIM, life cycle, Dynamo, data

1 Introduction

Historically, the capital facility industry (nonresidential, non-transportation) in the United States has issues with data efficiencies. It has been estimated that the fragmented nature of information exchange and management causes an annual loss of US$15.8 billion in the United States alone (Gallaher et al. 2004). Within these enterprises, 80–85% of the capital project dollars are consumed by costs associated with facility operations and maintenance (Marchese and Rudderow 2013). Creating more effective and efficient maintenance and management process can have a great impact on the industry in terms of cost savings (Salmon 2013).

Most owners address the issue of data management for facility operations and maintenance in one of the three broad strategies (Di Iorio 2013). First strategy involves directly referencing original project documents stored in various formats such as two-dimensional paper or electronic documents (unstructured data and not searchable). Second strategy involves the use of paper-based or electronic spreadsheets with structured data that have been manually extracted from drawings. The third strategy is to employ technology-based solutions, such as computerized maintenance management system (CMMS), which is typical for larger owners with complex facilities. Data are often input manually into the CMMS.

This study is part of a large university system's effort to adopt processes that allow for the use of building information models (BIM) throughout the life cycle of a facility to better support the operational needs of facility managers during operations and maintenance. Traditional methods to support documentation of life cycle information needed for asset management were manual in nature and time consuming to complete as the information tended to be fragmented and often incomplete. Specifically, this article examines the efforts to move from a manual

process of data manipulation to a semiautomated process that allows for transferring owner-specified data that were captured in the BIM to the CMMS. A case study of a mixed-use educational building is used to demonstrate the customized workflow.

1.1 Automating data transfer

Research has been conducted in how to move away from manual-based system and utilized systems such as Building Information Models for Facility Management (BIM-FM). In theory, BIM is capable of supporting FM operations with success. BIM provides an opportunity for facility owners to improve productivity, support proactive decision-making, and reduce costs (Bercerik-Gerber et al. 2012). Sadeghi et al. (2018) looked at BIM as a method to capture and store information for FM, supporting a decision support framework with the motivation of moving toward a data-driven process to overcome challenges of lacking timely and accurate information for facility operations. Chen et al. (2018) developed a framework utilizing an as-built BIM for creating automated work maintenance schedules by mapping facility data stored in the model through a developed IFC (Industry Foundation Classes) extension. The ultimate goal was time saving for performing maintenance activities. Beach et al. (2017) proposed a framework of utilizing federated models with an overlay condition to allow for transparency of data exchange and to maintain the ownership of the data in an attempt to address legal organization issues commonly found with a collaborative BIM process. This allowed for using the data within the bounds of the controls that are set up for the project. However, despite these efforts to directly link BIM to FM practices, Edirisinghe et al. (2017) argued that there is a gap of capable applications to fully support BIM during practice of FM operations. Edirisinghe et al. (2017) believed that further investigation is needed into productivity gains from working within a true BIM environment for FM versus the investment that is required earlier in the process. In practice, BIM is explored more as a method to support FM, not be fully integrated into the FM operations (Edirisinghe et al. 2017). Some of this is attributed to the shortage of BIM skills in the FM industry and the need for open systems and improved standards to support BIM-FM (Kassem et al. 2015).

The value of supporting FM with BIM, as opposed to fully utilizing BIM-FM, is the available life cycle data in the model. Data retrieval within a centralized BIM model and with enhanced collaboration in a BIM-based process can be of tremendous benefit because it allows for increased utility and speed of data usage by FM personnel (Parn et al. 2017). This has led to many efforts looking at utilizing BIM for FM in a support capacity for gathering and transferring information from the design and construction phases and ultimately linking the data into a CMMS system for operations and management. Heaton et al. (2019) developed and utilized a BIM extension to support asset management by collecting relevant information to support each asset within the BIM for easier transfer of data when the model was turned over to the owner. Other examples utilize a proposed framework to enhance the success of using Construction Operations Building Information Exchange (COBie) for life cycle information change (Alnaggar and Pitt 2018) and developing a workflow to export data from Revit using the COBie extension into a CMMS (Pishdad-Bozorgi et al. 2018). The latter limited the workflow to COBie compliant data in Revit and had other documentation uploaded directly into the CMMS after the project was turned over to support non-COBie data.

1.2 Challenges for BIM-FM

One often overlooked challenge in many research efforts is how to link BIM data to existing systems and tools that are used in practice (Miettinen et al. 2018; Sabol, 2008; Dixit et al., 2010). This can be especially true with larger owners of multiple facilities who have invested in data management infrastructure that has not yet come up with an integrated BIM solution and cannot simply switch to BIM-based FM or CMMS system. Borhani et al. (2017) worked within the confines of a large institution to identify a method for exchanging information from a BIM to an asset management system. The data were extracted from the BIM and mapped utilizing a data management tool and validated in the form of a specified Excel spreadsheet to be imported to the system.

Additional challenges lie within the lack of open systems and standardized data libraries that can be used to bridge between BIM and CMMS technologies and the need for rigorous BIM specifications for modeling requirements by the owner (Kassem et al. 2015). Some of this is attributed to owners having customized needs for managing a facility. Each facility is diverse and unique, and the organization's requirements to manage buildings also differ based on the need (Dias and Ergan 2016). This diversity inhibits the development of a standardized data structure as owners commonly have customized needs and are best suited to understand what information should be contained in them (Keady 2013). There is

a gap between what is typically included in a design and construction model and what is useful for FM (Halmetoja 2019). There is often a miss-match between what information providers think is valuable and what is actually needed to manage the facility (Wijekoon et al. 2018). This requirement needs to be better defined which requires appropriate planning to ensure a successful use of BIM to support FM.

Heaton et al. (2019) stated that having stakeholder engagement and the right personnel in the loop when determining the information needs is critical. Those involved in setting up an owner's data requirements need to understand the processes for managing the facility helps to identify the information that is needed to support those processes (Heaton et al. 2019; Cavka et al. 2017). Many issues from the project-delivery process when trying to support FM with BIM data extend from the exclusion of FM professionals in earlier project-delivery phases. Quality control of data needs to be an ongoing process. Lack of quality control can be a cause for potential problems and can arise with the slightest human errors (Pishdad-Bozorgi et al. 2018). Success of data in terms of quality, completeness, structure, and consistency is essential for successful data transfer to the owner (Alnaggar and Pitt 2018).

Documenting all available information just to "have everything" is not a viable solution either (Wijekoon et al. 2018). To help prevent the owners from requesting the collection of all data for the sake of having it, there are efforts that exist to help owners better identify what information they need. Cavka et al. (2017) created a framework for owners to identify the information that they need to support life cycle operations activities and understand how they map to FM database systems in an owner's customized processes to support their specific information needs. Additionally, Hosseini et al. (2018) worked on identifying and classifying key information that supported BIM for FM by creating a taxonomy of categories of information to help owners and contractors more easily identify the types of information that the user needed for managing the process.

1.3 COBie for FM data collection

One standard that has been developed is the COBie. COBie is commonly sought after for assisting in the transfer of data throughout the project life cycle. Systems have been developed to help manage the COBie data in model form and transfer that data for use in FM (Alnaggar and Pitt 2018; Pishdad-Bozorgi et al. 2018). Parn et al. (2017) identified that many practitioners did not receive COBie well with a perception of it as a "one size fits all." Many professions blindly adopt COBie in its entirety without truly identifying which information they need (Cavka et al. 2017). When the specifications are not defined and left to the designers to determine, they tend to incorporate more than what is needed while also missing some information that would otherwise be valuable to the owner (Parn et al. 2017). Many of these perceptions can be attributed to the lack of education and training on how to utilize COBie (Alnaggar and Pitt 2018).

Even with property planning and specification by the owner, COBie can still present challenges. In large educational institutions and for other owners with diverse portfolios, it has been found to be difficult to adopt COBie for the fact that information in these organizations is managed by many subgroups that have domains and information territories (Anderson et al. 2012). Often, the use of COBie requires a redesign of traditional processes of data preparation and exchange between construction and FM teams (Alnaggar and Pitt 2018). This can be especially difficult for owners who have long-existing workflows in place that now need to be adapted to fit within COBie (Cavka et al. 2017).

Additionally, if the facility manager has data requirements that are in other formats other than COBie, the data need to be segregated and planned for in another way (Alnaggar and Pitt 2018). Pishdad-Bozorgi et al. (2018) addressed this by keeping track of non-COBie data outside of the specification process and uploading into the CMMS system after the model was turned over. Parn and Edwards (2017) proposed a plug-in extension to COBie to better integration of sematic data linked to 3D objects; however, this was still designed to fit the client's needs and would require additional development of standard FM practices to be truly generalizable to the industry.

Another complication, as with any spreadsheet, is that COBie in a spreadsheet view presents large amount of facility data in tabular formats across many worksheets. This presents usability challenges such as visualizing the extent of the content, understanding data dependencies within and between the workbooks, and memory overload from the sheer volume of numerical and text-based data (Yalcinkaya and Singh 2019). Even though the data model extensions exist to manage the data in a three-dimensional (3D) view, the interconnection of data can cause confusion and lead to insufficient COBie data (Alnaggar and Pitt 2018). These model extensions are not without issue either. Pishdad-Bozorgi et al. (2018) found interoperability issues when importing COBie compliant data into a

compliant CMMS that produced errors because of special formatting requirements. At the beginning of this research study, COBie was examined and presented to the owner as a potential starting point. However, since the owner had a well-developed process with specialized needs that COBie could not directly support and there was resistance for training the workforce on a new complex system, this led to the development of a simplified proprietary data classification system to suit the owner's needs and work with their existing workflow.

1.4 Current study

The study presented in this article is part of a larger research effort by an educational institution to develop a more efficient workflow of documenting asset-related information throughout the project life cycle for use during FM. The educational institution whose processes are being studied has adopted the use of AiM by Assetwork as their CMMS. Thabet and Lucas (2017a) described in detail the manual process used to collect data for the CMMS. FM personnel were reviewing equipment schedules at the end of construction and commissioning and manually inputting the data into a spreadsheet that would then be manually input into the CMMS. The main challenges included missing data, incomplete data, and a long lead time between the commission of the building and usability of the information within the CMMS. The original process was very manual and included many unstructured data sources for populating asset information in the CMMS. Relying on unstructured information leads to a lack of data readiness which extends the gap between project completion and when the information can be used, whereas structured data can lead to efficiencies in FM and reduction of time where data becomes usable within a CMMS (Fallon and Palmer 2006). By creating a method for structuring the data throughout the life cycle of the project, the owner would be able to populate the CMMS with relevant data in a more timely fashion, preferably automatically. The overall research methodology is presented in Figure 1.

Phase 1 of the research included an information needs analysis to identify the data required to support asset management practices. Existing practices and workflows were examined to identify the needed data. A list of mission critical assets was identified. Data needs for each of these assets were also determined. These assets and related data were documented with the creation of a BIM-FM Guidelines that outlines the requirements for collecting data throughout the design and construction phases of the project. The goal of the BIM-FM Guidelines was to provide an understanding, at the life cycle level of the project, of what data were needed so it can then be properly structured and more efficiently input into the CMMS. Phase 1 is a prior research discussed in Thabet and Lucas (2017a).

Phase 2 of the research was to identify methods for documenting the information that were compliant with the guidelines. A case study approach was used for buildings both under renovation and as new construction. Alternatives were explored for documenting information within the Revit environment. These alternatives are discussed in detail in Thabet and Lucas (2017b). The data were collected from multiple project deliverables including contract documents and submittals. The data collected were validated by the organization through an iterative interview process. This allowed for a BIM that contained correct data in the appropriate format to be used in phase 3 of the current study. The case study information used in the current study is discussed in Section 2.

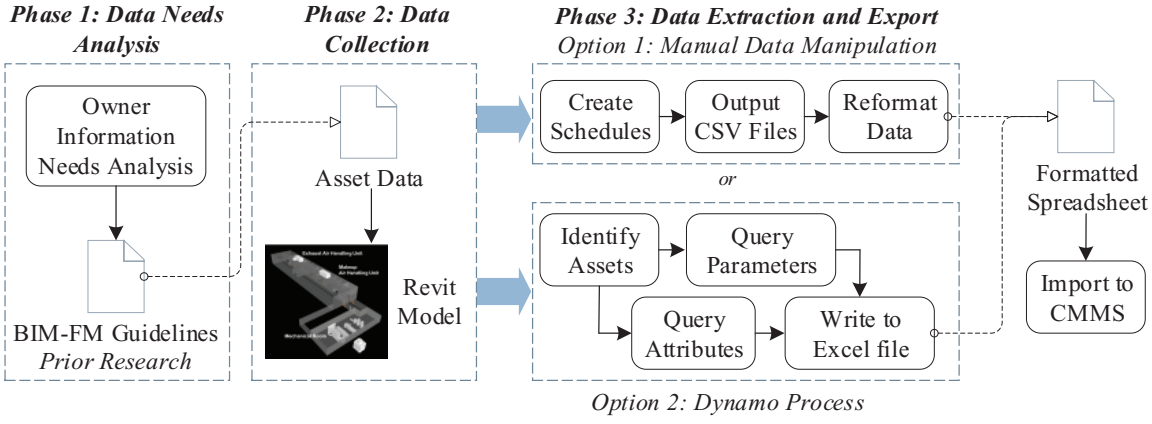

Fig. 1: Research methodology.

Phase 3 included an exploration of how to complete the workflow and allow the asset data collected within the model throughout design and construction to be incorporated into the CMMS. Direct export from the BIM to the CMMS was not available due to the defined processes and systems used by the organization. The first option for populating data within the CMMS included a manual manipulation of spreadsheet that consisted of extracted data from schedules and data tables within Revit. The CMMS, to support the needs of the owner, requires the data to be formatted in a specific format. The manual manipulation of this spreadsheet was time consuming and could lead to errors, therefore a customized process utilizing Dynamo and Python scripting was explored. This process allows for life cycle asset data to be extracted from a Revit model and used directly within the CMMS. Automating this process allows for both efficiencies in data usability and reduction in unintended error possible by human manipulation of the data. Dynamo was used because of its ability to create a customized workflow of the data while automating the otherwise manual process. The script created within Dynamo allows for a saving time in the data transcription process. The created process was validated by comparison of the intermediate steps and final implementation of the data within the CMMS system between the manual and automated processes. This article documents the Dynamo process used for the information extraction process.

2 Case study project and FM information needs

A mixed-use multistory educational building that included offices and laboratories was used for this case study. Using the project plans and submittals, 33 pieces of equipment were identified which belong to 11 asset groups. Table 1 summarizes the asset groups and equipment names. Equipment name abbreviations adopted by the academic institution are based on the US CAD Standard (NIBS 2019). Equipment names are listed as it appears in the plans. Table 1 also shows parent/child relationships and the floor/room where each equipment is located.

Each asset group has common and specific attributes that are used for managing the asset post-construction. Thabet and Lucas (2017a) discussed the institution's classification and definition of asset attributes and how they support FM activities. Two types of attributes are defined, namely common attributes, also known as properties, and specific attributes. Common attributes are similar across the different asset groups, while specific attributes are unique to the asset group.

There are 22 common attributes (properties) for any asset group. This includes attributes such as ASSET_TAG, ASSET_GROUP, MANUFACTURE _CODE, MANU_PART_NUMBER, SERIAL NUMBER, WARR_DATE_FR, etc. Specific attributes vary for each asset. Tables 2 and 3 illustrate

Tab. 1: Asset groups and equipment 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
TNK	PET-1		Mech/Elec Rm 7
TNK	PET-2		Roof
EMER-ATS	ATS-1, ATS-2		Basement—Mech. Equip. Rm
EMER-GEN	Emergency Generator		Basement—Mech. Equip. Rm

Tab. 2: Common asset attributes for MAU-1 (AHU asset group)

Common attributes	Common attribute value	Data source/remarks
ASSET_TAG	00441-AHU-RT-0001	VT Asset Tag Naming Convention
ASSET_GROUP	AHU	VT Asset Groups List
DESCRIPTION	MAU-1—MAKEUP AIR HANDLING UNIT	SHEET M1.2
LOCATION_CODE	ROOF	SHEET M2.6
DATE_PURCHASED		
LONG_DESC	AIR INTAKE SECTION W/FILTERS (100 PERCENT OUTDOOR AIR HOOD), ENERGY RECOVERY COIL SECTION (GLYCOL LOOP), 24" ACCESS SECTION, STEAM PREHEAT COIL SECTION, 24" ACCESS SECTION, CW COIL SECTION, PLENUM SUPPLY FAN, HUMIDIFICATION SECTION FOR DISPERSION MANIFOLD, 48" PLENUM W/ TWO BOTTOM DISCHARGE OPENINGS. OVERALL UNIT DIMENSIONS APPROX. 27'–6"L ´ 7'–10"W ´ 5'–8"H. APPROXIMATE UNIT OPERATING WEIGHT 8,000 LBS; OMNICLASS23: BUILT UP ROOFTOP AIR HANDLING UNITS	OMNI CLASS 23
LOCKOUT	Y	SUBMITTAL EAU_ MAU_050216—p12
SERIAL_NO	K16E36891	Bldg. Walkthrough
MANUFACTURE_CODE	TRANE	SHEET M1.2
MANU_PART_NUMBER	CSAA030UB	SUBMITTAL EAU_ MAU_050216—p37
PROCEDURE_YN	Y	SUBMITTAL EAU_ MAU_050216
PARENT_ASSET_TAG	N/A	
YEAR_INSTALLED	2017	
PART	12 ´ 24 2" Pleated Media MERV 8	SUBMITTAL EAU_ MAU_050216—p35
QUANTITY	3	
PART	20 ´ 20 2" Pleated Media MERV 8	SUBMITTAL EAU_ MAU_050216—p35
QUANTITY	8	
PART	BX-98 Belt	
QUANTITY	2	
PART	12 ´ 24 12"Cartridge 95% MERV 15	SUBMITTAL EAU_ MAU_050216—p35
QUANTITY	3	
PART	20 ´ 20 12"Cartridge MERV 15	SUBMITTAL EAU_ MAU_050216—p35
QUANTITY	8	
WARR_DESC	1 YEAR PARTS AND LABOR; 5 YEARS PARTS ONLY for ECONOMIZER	SUBMITTAL EAU_ MAU_050216—p3, 11
WARR_DATE_FR		
WARR_DATE_TO		
LOCATION	BLDG441—BASEMENT & MAIN FLOOR	M2.4, M2.5 AND M2.6
USAGE_FACTOR	50, 50	
CLASS_STANDARD	OMNICLASS, Master Format	OMNI CLASS 23, Master Format 2016
LEVEL_1	23, 23	OMNI CLASS 23, Master Format 2016
LEVEL_2	33, 74	OMNI CLASS 23, Master Format 2016
LEVEL_3	25, 16	OMNI CLASS 23, Master Format 2016
LEVEL_4	11	OMNI CLASS 23, Master Format 2016
LEVEL_5	13	OMNI CLASS 23, Master Format 2016
LEVEL_6		

an example list of attributes and values collected for the asset MAU-1 (Makeup Air Handling Unit) that belongs to the asset group AHU (Air Handling Unit). The tables also list the source contract document where the information was captured. Missing attribute values could not be found or are not applicable.

Most attribute values were determined from the plans and submittals. A building walkthrough allowed verification of captured values and determination of some of the missing fields for the case study project.

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

AHU common attribute	Common attribute value (MAU-1)	Source/remarks
Bas point address	CMMU1SAF	Bldg. WALKTHROUGH
Heating source	BOILER	SHEET M1.2
Heating capacity	13,500	
Total air flow	13,330	SHEET M1.2
External static pressure	1.5	SHEET M1.2
Min. outside air		
Max. outside air		
Cooling source	CHILLER	
Total cooling capacity	704.09	SUBMITTAL EAU_ MAU_050216
Sensible cooling cap	455.52	SUBMITTAL EAU_ MAU_050216
Economizer	Y	
Economizer type		
Total static pressure	5.006	SUBMITTAL EAU_ MAU_050216
Type	RT (Roof Top)	SHEET M1.2

The renovation project did not have a 3D model available. For the purposes of this research, three 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. 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. Parameters and data were loaded into the mechanical and electrical models using a third-party plug-in for Revit from CTC (www.ctcexpresstools.com) and the three models were then 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 develops his own model.

The BIM-FM standards for the research institution under consideration require that data captured in the BIM model must be exported in a proprietary format to an Excel spreadsheet to allow for direct upload of the data to their FM system. Properties and attributes must be placed in six separate sheets (tabs), in the exact order shown, within the Excel spreadsheet: General, Attribute, Parts, Warranty, Location Served, and Classification. Figure 2 shows how the properties and attributes are distributed.

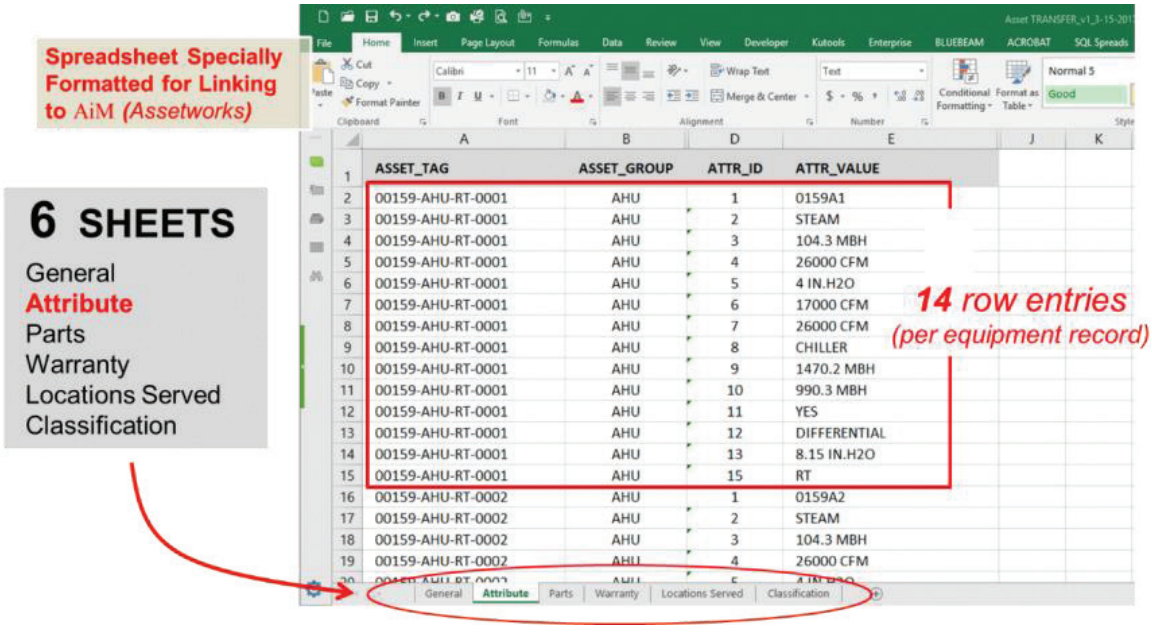

Fig. 2: Allocation of asset properties and attributes in the formatted Excel spreadsheet.

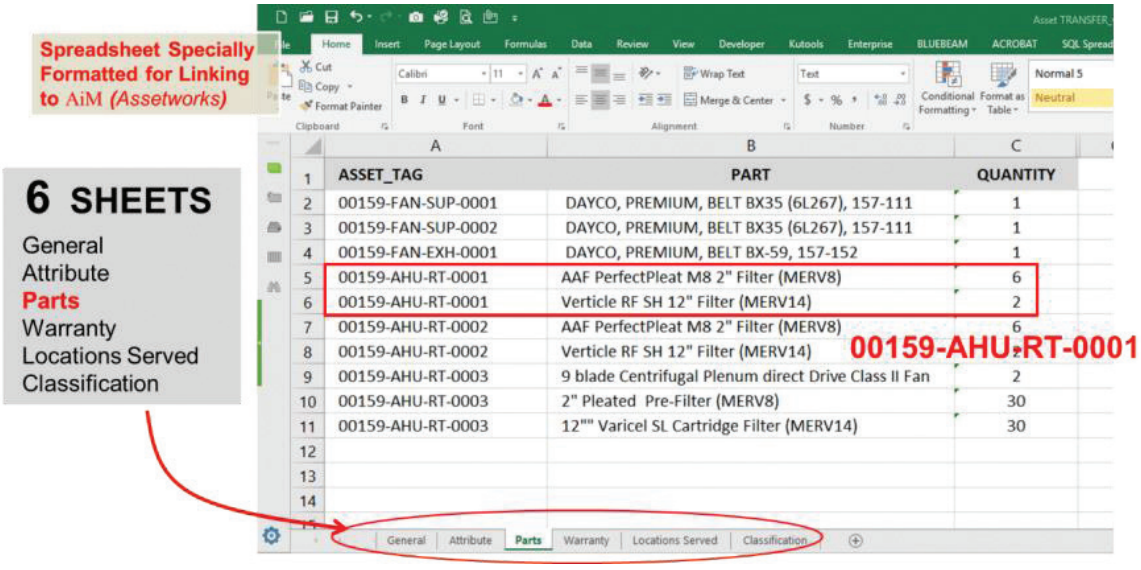

Fig. 3: Example of the "Parts" tab.

The figure also highlights how the attribute data for a given asset (e.g., AHU-1) are represented.

Figure 3 provides another example of how the data should be formatted in the Parts tab. Each required part and its specified quantity for each asset is represented in its own row in the format shown. For example, for AHU-1 there are two parts required with quantities specified. The parts data for AHU-1 are represented in two rows in the spreadsheet. Other property data (General, Warranty, Location Served, and Classification) follow a similar format.

Given this proprietary formatting requirements for the data spreadsheet, a custom novel solution was needed to extract and export the data from the BIM model and create the required spreadsheet. No current workflows and methods available could support such format. Using Dynamo programming and Python script, a novel workflow is developed and described in Section 4.

3 Data transfer mechanisms

Thabet et al. (2016) proposed a BIM-FM holistic process workflow to define, capture, and transfer life cycle data to a 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 by Thabet and Lucas (2017a). The first case study utilized a paper-based spreadsheet method of documenting the required data types and validating the identified information needs by using a newly constructed classroom facility. The first case study focused on the documentation of five mission critical assets (Air Handler Units, Fans, Emergency Transfer Switches, Boilers, and Emergency Generators). A second study explored a BIM-based documentation approach for the same defined assets in the same building (Thabet and Lucas 2017b). The documentation methods looked at the data capture through creating custom parameters in Revit and directly inputting values into Revit schedules as well as the use of third-party plug-ins for managing parameters and values. The study resulted in the use of Pentaho, a data mapping tool, to connect schedule data from Revit that was exported into a CSV file and manually reformatted to AiM from AssetWorks. The result of these case studies ended in a preliminary draft of the BIM-FM Guidelines that defined the data documentation requirements.

The purpose of the current phase of the research is to identify advanced methods for electronically transferring the captured data from the Revit model directly into AiM with minimal manual manipulation of the data. Three main methods for data transfer were examined as part of this current phase of research 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 in the required format so that they 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 not to be a viable option because it would require them to switch to Archibus to manage all facilities. Lessons learned were documented to inform the development of a future plug-in that will allow direct data exchange between Revit and Assetworks AiM, the CMMS system used by the academic institution.

The Dynamo process was chosen as the tool to be explored by the research work presented in this article.

4 Proposed novel workflow using Dynamo for extracting and exporting asset data

Dynamo, an open-source graphical programming tool within the Revit modeling environment (Dynamo 2017), was used to extract data from the model. Dynamo allows for incorporation of predefined scripts, called nodes, to be connected together to perform custom operations within the Revit model environment. In addition, custom scripts using Python language are incorporated. In order for the facility data collected in the model to be usable for the CMMS platform, a specific format of data extraction is necessary. Dynamo allows this customized data extraction and eliminates the need for manual data manipulation otherwise required to format the data into the spreadsheet form required for connecting the data to the CMMS.

The overall process for the Dynamo script shown in Figure 4 includes two parts: collecting and writing the data for the asset data included on the "General," "Parts," "Location," "Warranty," and "Classification" tabs (steps 1, 2, and 3) and then documenting the "Attributes" tab (steps 1, 4, and 3). The Attributes tab requires special formatting and unique code, whereas the other five tabs use a similar process. A summary of the steps are as follows:

Step 1: The elements within the model associated with a specific asset type are identified.
Step 2: Parameters for each of the assets in each group are extracted into formulated lists.
Step 3: Data are written into an Excel spreadsheet to be interpreted as a.CSV file type.
Step 4: Attribute data are queried based on individual elements and the separate lists are transcribed and reformatted before being written to Excel.

4.1 Step 1: Identifying model elements

Model elements representing the required assets to be exported are identified. Figure 5 illustrates the required Dynamo nodes to identify the family instances within Revit that represent these assets.

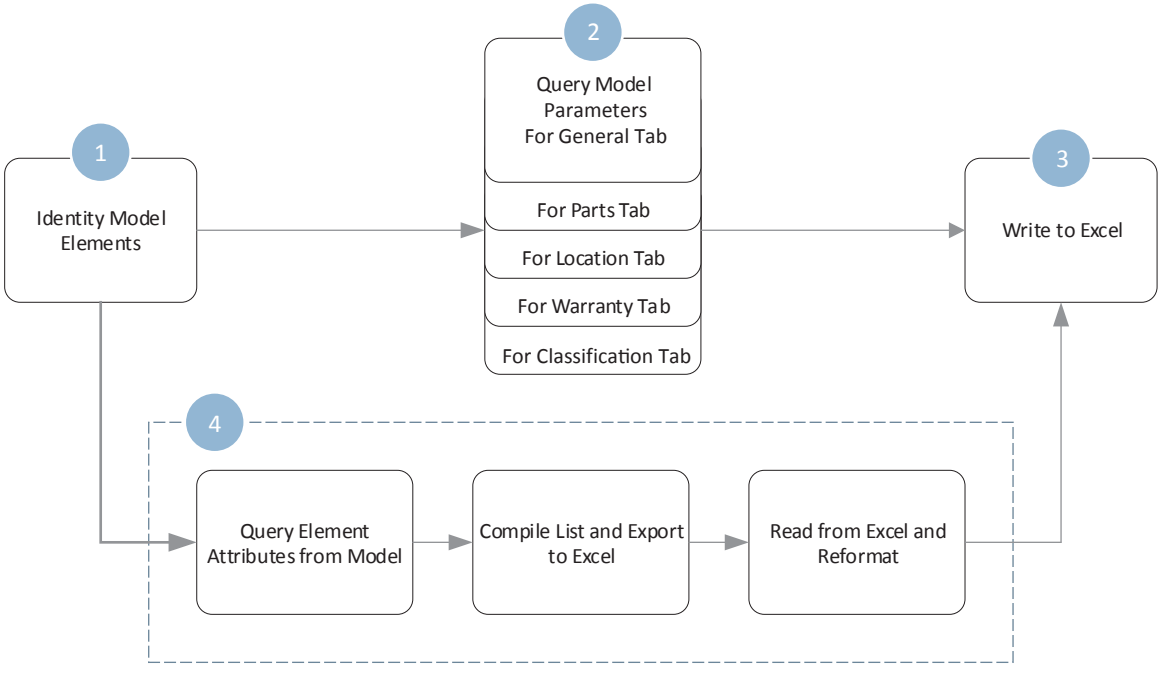

Fig. 4: Summary of the proposed Dynamo workflow.

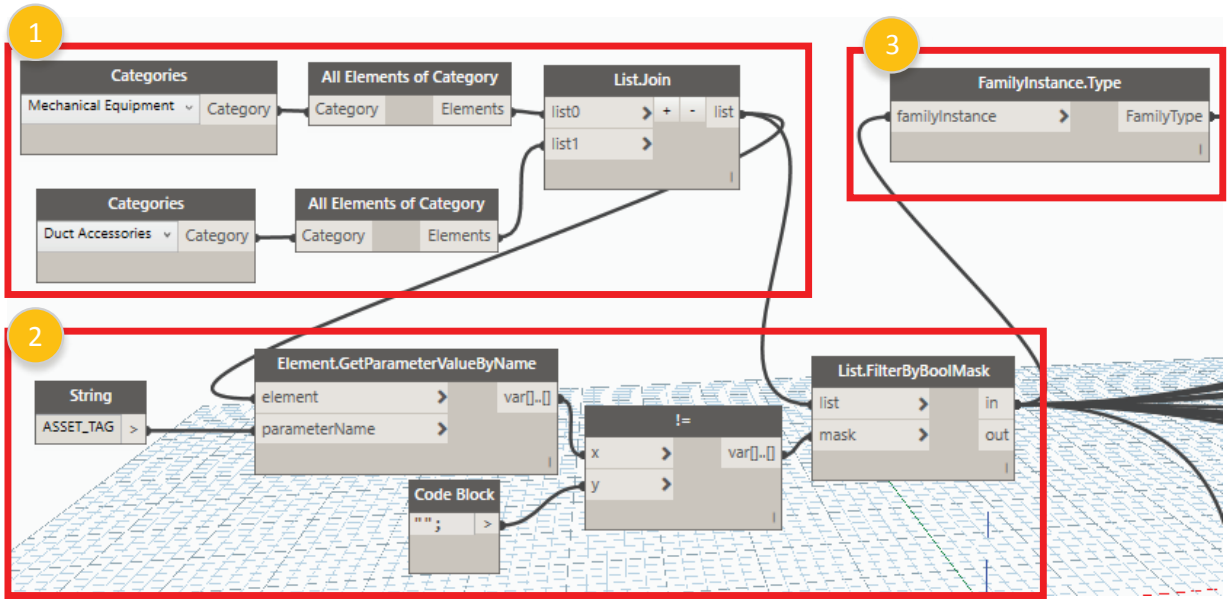

Fig. 5: Step 1: Identifying model elements.

With reference to Figure 5, the following substeps summarize this process:

1. The nodes "Categories" and "All Elements of Category" access all family instances of a particular category. For the case study, *Mechanical Equipment* and *Duct Accessories* were the required categories that contain the targeted assets. The "List.Join" node compiles lists from each of the inputs that can be queried in future parts of the process.
2. Identify only the elements representing assets being tracked. The "All Elements of Category" identifies all elements in the model in one of the two categories chosen. To limit the elements in the list to the required assets, 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. 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) with 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.
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." The different inputs required for each parameter are identified in Table 4 and consist of the output list from either the "List.FilterByBoolMask" or "FamilyInstance.Type" nodes.

4.2 Step 2: Querying model parameters

The objective of this step is to identify the parameter values and create appropriate lists of those values that can be written to the proprietary spreadsheet CSV file in step 3. Step 2 comprises two substeps as shown in Figure 6. The first is to identify the values of required parameters for each tracked asset. The "Element.GetParamaterValueByName" node utilizes two inputs. Depending on the parameter type, the "element" input is a list as the result of the sources identified in Table 4. The "ParameterName" input is a string of the parameter's name. The result is an indexed list of all parameters of the queried elements. Each output of the "Element.GetParameterValueByName" node is then added to a list using the "List.Create" node (Figure 6, substep 2).

4.3 Step 3: Writing data to Excel

The General, Parts, Location, and Warranty tabs require a similar workflow and the same Dynamo nodes to query the

Tab. 4: Parameter types and corresponding input list

		List used as input	
Sheet	**Parameter name (input)**	**List.FilterByBoolMask**	**FamilyInstance.Type**
General	ASSET_TAG*	X	
	ASSET_GROUP		X
	DESCRIPTION	X	
	LOCATION_CODE	X	
	DATE_PURCHASED	X	
	LONG_DESC		X
	SERIAL_NO	X	
	LOCKOUT		X
	MANU_PART_NUMBER		X
	MANUFACTURE_CODE		X
	PROCEDURE_YN		X
	PARENT_ASSET_TAG	X	
	YEAR_INSTALLED	X	
Classification	ASSET_TAG*	X	
	CLASS_STANDARD		X
	LEVEL_1		X
	LEVEL_2		X
	LEVEL_3		X
	LEVEL_4		X
	LEVEL_5		X
	LEVEL_6		X
Location	ASSET_TAG*	X	
	LOCATION	X	
	USAGE_FACTOR	X	
Parts	ASSET_TAG*	X	
	PART		X
	QUANTITY		X
Warranty	ASSET_TAG*	X	
	WARR_DESC		X
	WARR_DATE_FR	X	
	WARR_DATE_TO	X	

*ASSET_TAG appears at the top of each list; when output to Excel, it is only queried once.

data. The Classification tab, because of the way it is formatted, requires a custom Python script to allow for arranging the data in the format required by the spreadsheet.

Once a list is created for all required data in the six spreadsheet tabs, it needs to be formatted and written to Excel (Figure 7). This workflow comprises of two substeps. The first substep uses the "List.Transpose" node to transpose the data to be written horizontally. The second substep utilizes the "Excel.WriteToFile" node to write the data to the spreadsheet. This node requires inputs from the file path to write the data, the sheet name where the data is to be written, the column and row numbers to start writing the data, the list of data, and if data should be overwritten—meaning clearing the sheet if it already exists or simply replacing the values in the cells at the starting part identified. This process is applied to all four tabs: General, Parts, Location, and Warranty. Figure 7 illustrates an example using the "General" tab.

4.3.1 Writing data to Excel for the "Classification" tab

The OmniClass parameter value for each asset consists of a series of integer numbers defined in the Revit model in a single field. For example, 23.27.19.15 is the OmniClass value for an asset represented in the model. This value needs to be separated into five separate integer numbers when exported to the Excel spreadsheet. The Python script within a Python node shown in Figure 8 is used to create custom functions to transform the one value Omniclass entry in the model to four separate integers to export to the spreadsheet.

The first five lines of the code are default lines written by Dynamo, and these lines import the data from the nodes into the variable IN. The variable IN is a list; for example, if there are two inputs to a node the first input will be accessed using IN[0] and the second using IN[1]. Working on input data directly can corrupt the entire

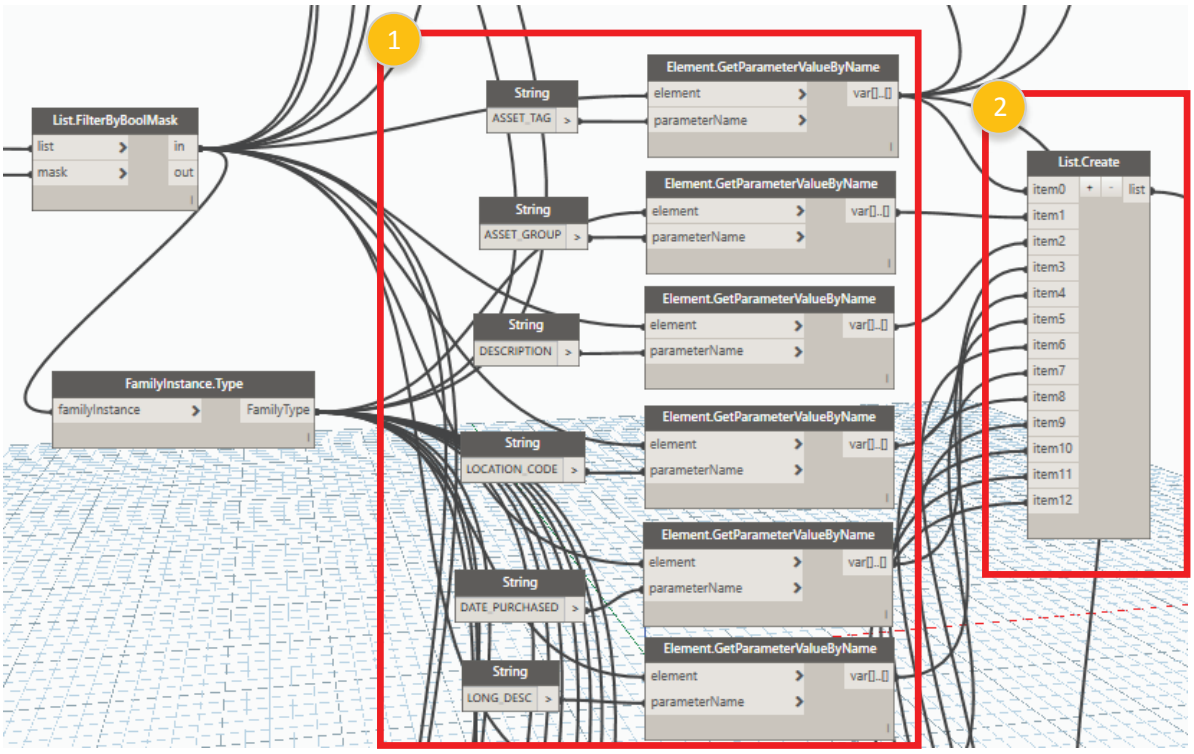

Fig. 6: Step 2: Querying model parameters.

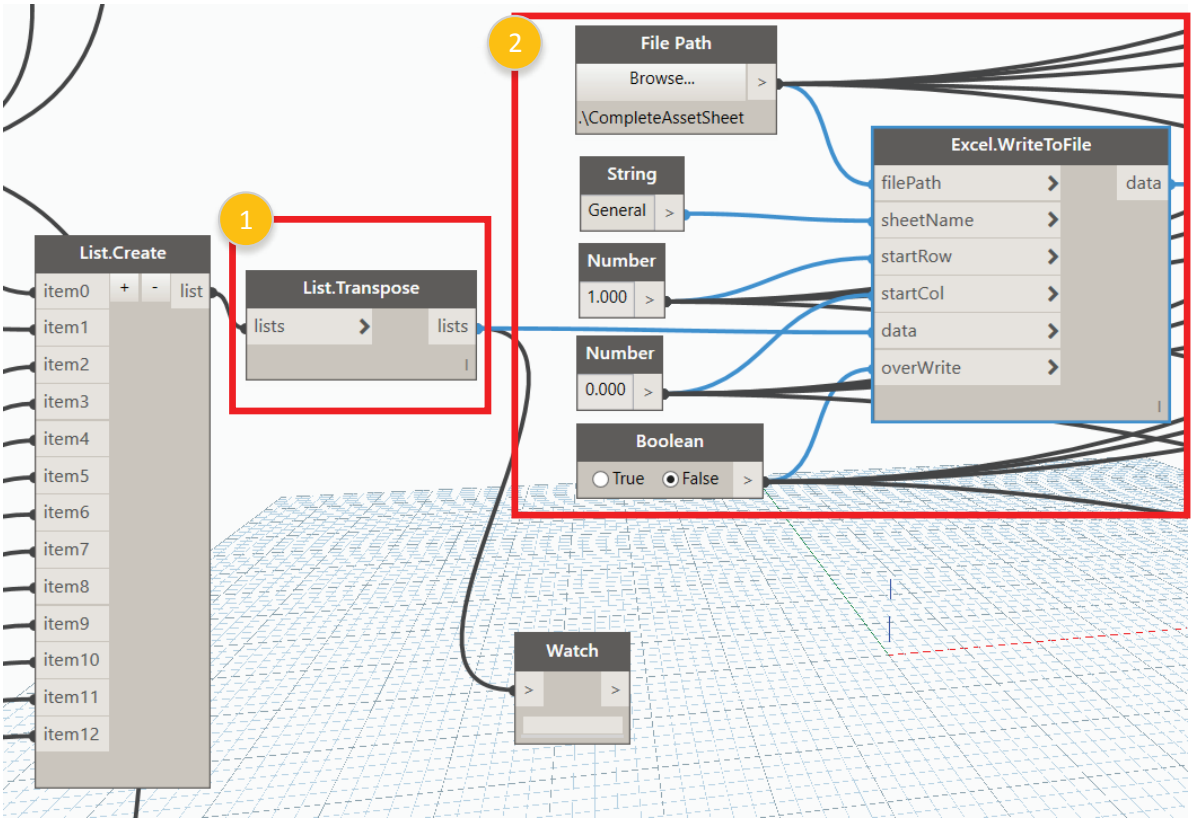

Fig. 7: Step 3: Writing data to Excel ("General" tab).

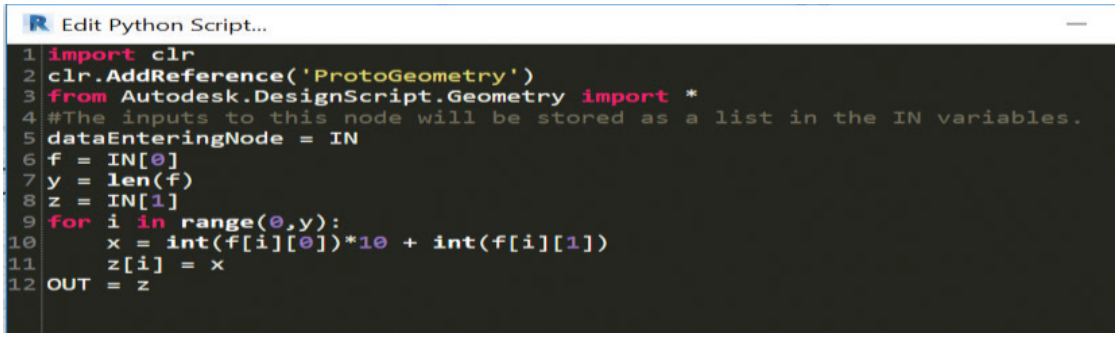

Fig. 8: Python code for converting OmniClass values.

code, so the input data are transferred into temporary variables for performing the operations. The variables f and z in lines 6 and 8 are used as temporary variables in this program. The "len" function in the Python script finds the length or number of elements in a list. The variable y in line 7 stores the length of the input variable which would be used in the next steps.

Line 9 is used to initialize the loop function. The loop function allows repeated execution of a command in a program. The number of executions is set by defining the range that in this case is identified by the length, or y variable. The code on line 9 executes from 0 to the point where value i matches that of y. For example, if the value of y is 2, the loop will be executed twice.

To understand how lines 10 and 11 in Figure 8 function, consider an example using two OmniClass numbers: 23.27.19.15 and 23.27.17.13. The script would store the data as shown in Table 5.

Now assuming the variable "f" has the list from Table 5 stored into it, Table 6 shows the query functions required to access each element in the list.

With the above example in consideration, a sample case is explained as follows:

Since there are two elements in the list, the value of y is equal to 2. The variables f and z have the values stored in the format shown in Table 5.

Case 1: for-loop when value of i = 0:

Line 10: x = f[i][0]*10 + f[i][1]

Since value of i is 0

Line 10: x = f[0][0]*10 + f[0][1]

Substituting values from the above table,

x = 2*10 + 3

x = 23

Line 11: z[i] = x

or

z[0] = 23

Hence the value in location 0 of list z is replaced with 23.

Tab. 5: Stored OmniClass values

Element index			
0		1	
Value index	Value	Value index	Value
0	2	0	2
1	3	1	3
2	.	2	.
3	2	3	2
4	7	4	7
5	.	5	.
6	1	6	1
7	9	7	7
8	.	8	.
9	1	9	1
10	5	10	3

Tab. 6: Query functions to access value

Query function	Value
f[0]	23.27.19.15
f[0][0]	2
f[0][1]	3
f[1]	23.27.17.13
f[1][4]	7

Thereby from a list of OmniClass categories we are able to obtain a list of level 1 values. By altering the location of the list, we can similarly obtain a list of other levels.

4.4 Step 4: Writing data to Excel for the "Attributes" tab

Writing asset data to the "Attributes" tab using the format described in Figure 2 is more complex than a simple linear export used for other tabs. The process first requires that

attributes data are exported to a temporary file and organized in separate tabs, each tab representing attribute data for a single asset group. Dynamo code is then used to consolidate the data creating one long list that is exported to the "Attributes" tab.

Step 4 comprises of four substeps illustrated in Figures 9 through 12:

4.1 Identifying all the assets within a specific asset group (e.g., TNK).

4.2 Extract attribute data for each group and place them in their own tab in a temporary Excel file.

4.3 Read the data back from the temporary file and consolidate into one continuous list.

4.4 Write the reformatted data back to the "Attributes" tab in the main Excel spreadsheet.

In step 4.1 (Figure 9), elements specific to an asset group are identified based on their "ASSET_GROUP" value. A list of the elements belonging to an asset group is generated. This is repeated for all the asset groups.

As shown in Figure 10, the output list of elements from step 4.1 is then used as an input to create the first

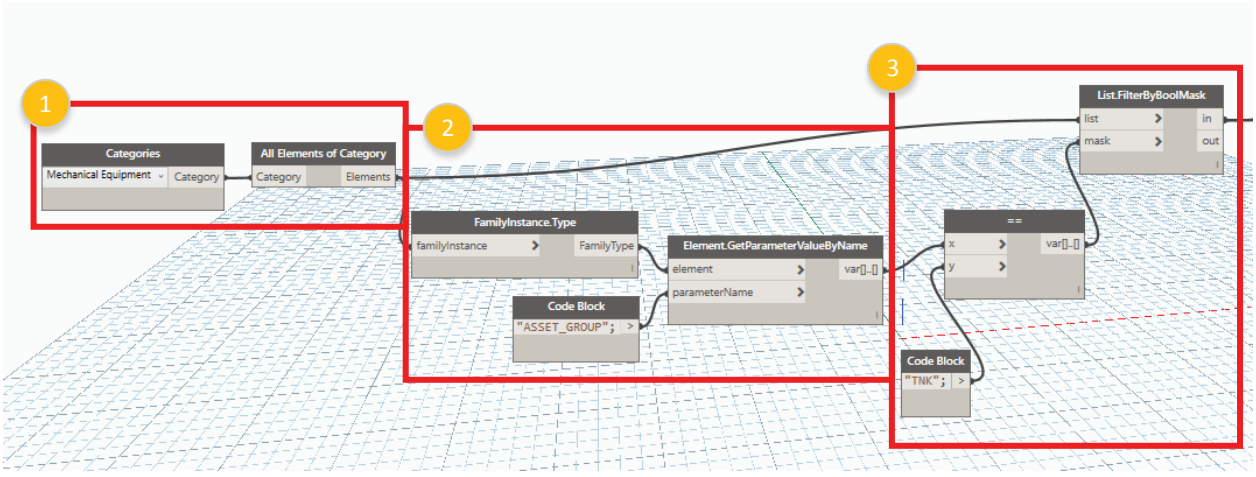

Fig. 9: Step 4.1: Access all elements of a specific asset group.

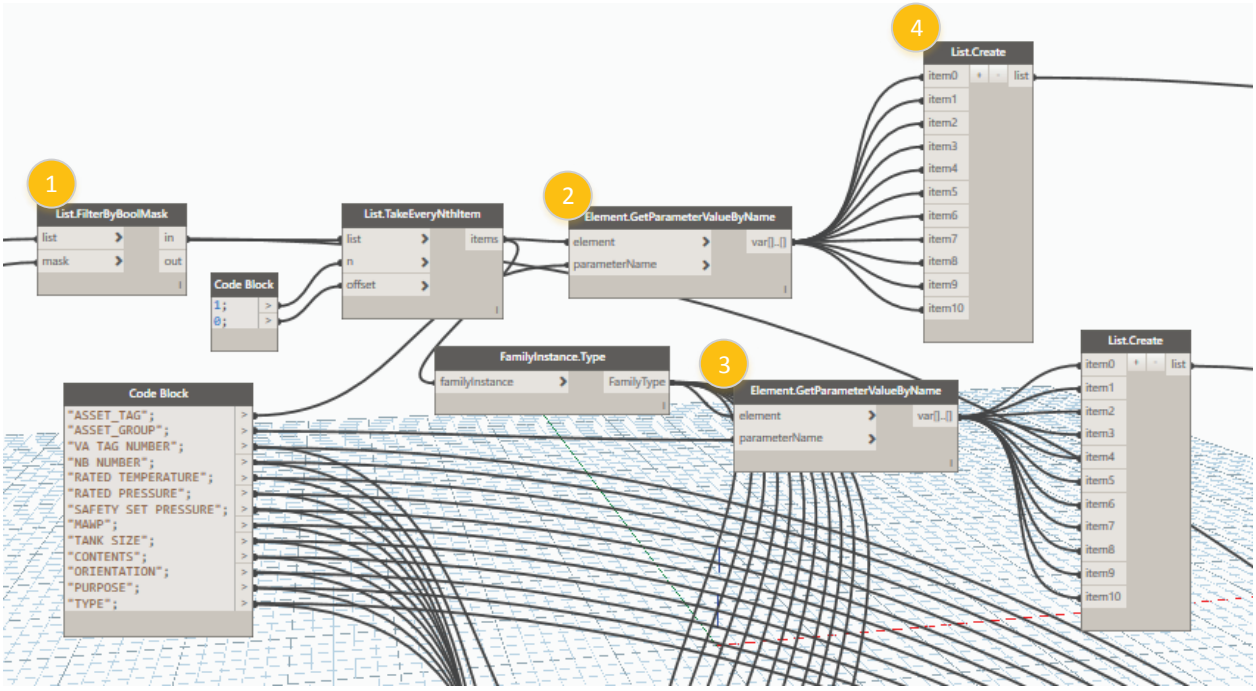

Fig. 10: Step 4.2: Creating column 1 (ASSET_TAG) and column 2 (ASSET_GROUP).

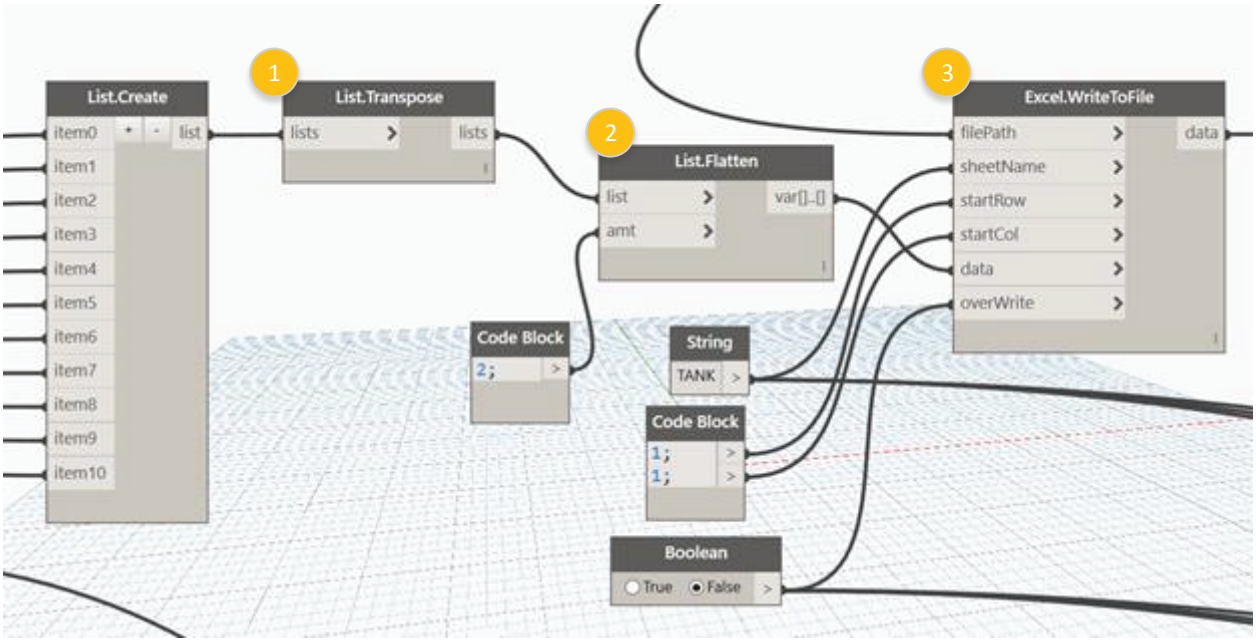

Fig. 11: Step 4.2: Transpose and write columns 1 and 2 to a temporary Excel file.

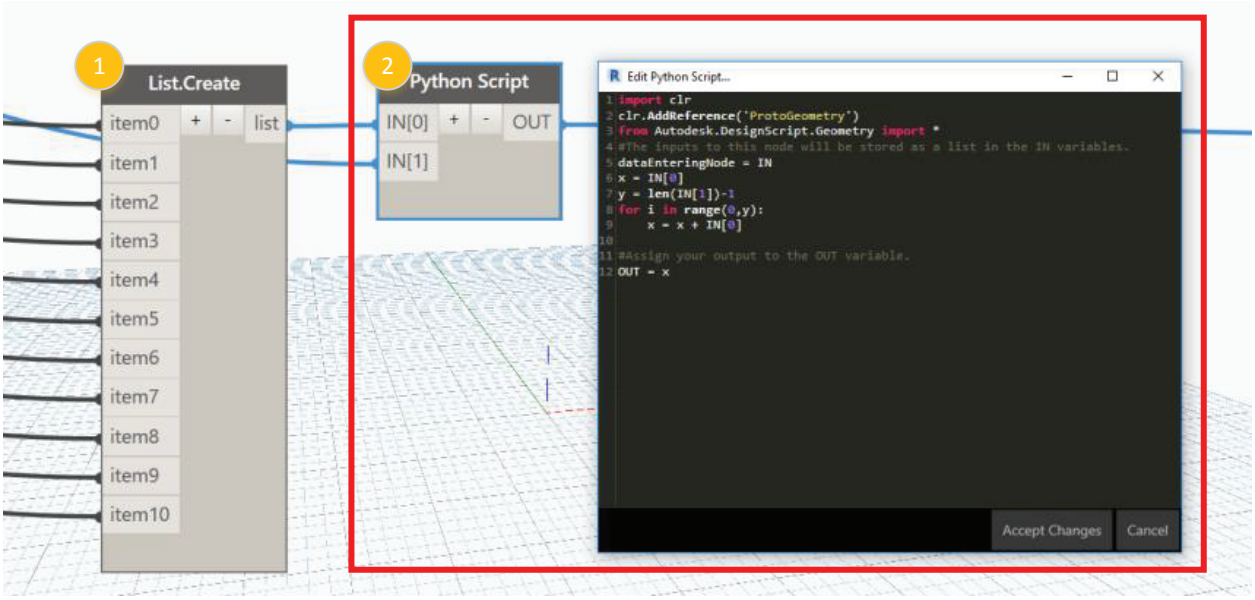

Fig. 12: Step 4.2: Column 3—Writing Attribute ID for each asset.

two columns in the "ATTRIBUTES" tab: ASSET_TAG and ASSET_GROUP.

The list of a specific ASSET_GROUP serves as the input for "List.TakeEveryNthItem" node with "n=1" and "offset=0." This simply means that it will look at one item at a time starting with the first (index 0) item and looking at every nth item (in this case 1—meaning every item). Second, a parameter name is added utilizing the "parameterName" input value of "ASSET_TAG" in forming column 1 (ASSET_TAG) of the temporary file by means of the "List.Create" node. Third, the value "ASSET_GROUP" is identified to form column 2 (ASSET-GROUP) of the Excel file by way of a separate list. The output is the value of the parameter. In the case "ASSET-GROUP," since it is a Family Instance, the "FamilyInstance.Type" node is used in the intermediate step. The list is cycled through by each element.

Next, column 3 representing attribute id (Figure 12) is generated. This process takes a filtered list of elements

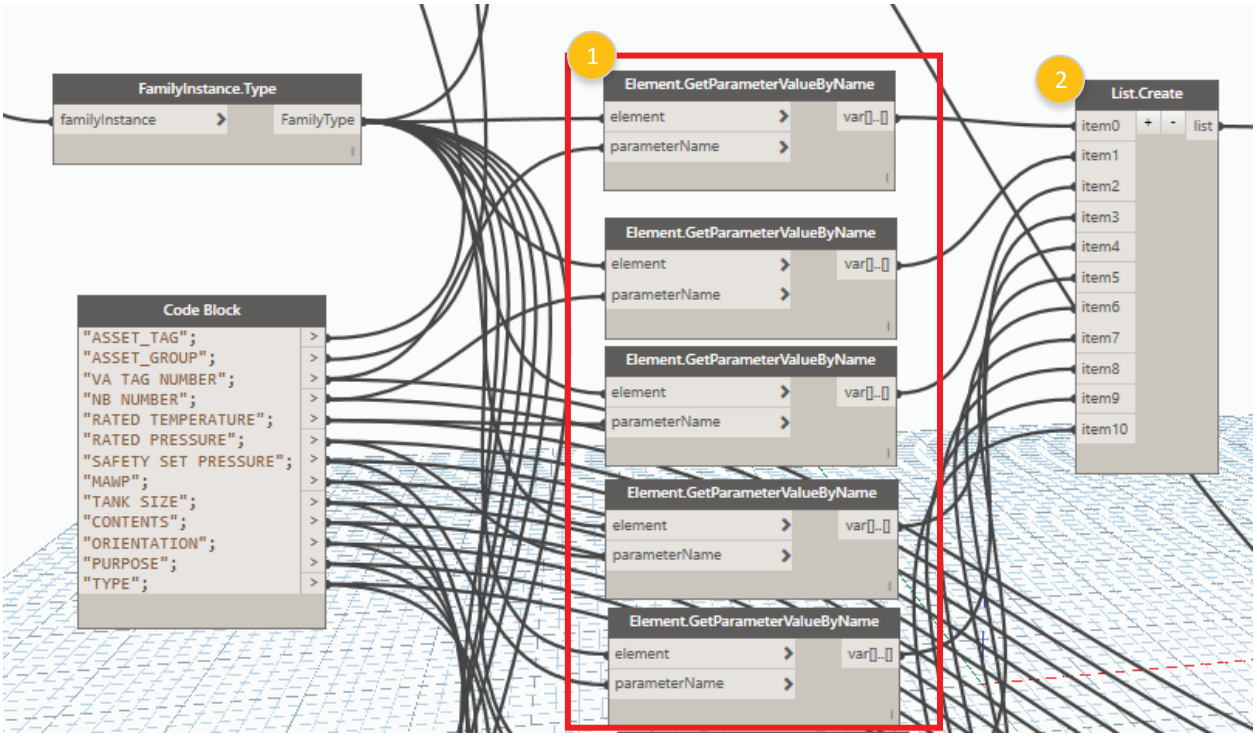

Fig. 13: Step 4.2: Column 4—Writing attribute values.

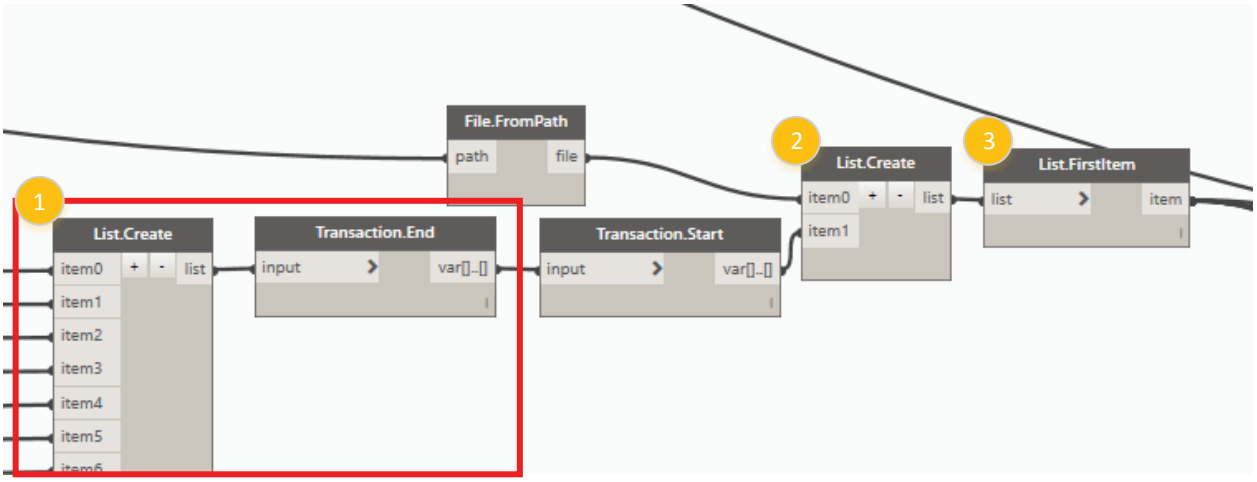

Fig. 14: Step 4.3: Preparing to read data from temporary file.

with the same value for the "ASSET_GROUP" parameter. Within the python script, the number of assets within the group is identified. Then, through the implementation of a loop, all of the attribute names for the asset group are written for each of the assets. The list of attribute names is then written to the temporary file.

Column 4 of the "ATTRIBUTE" tab holds the attribute values for each asset. Attribute values are the output of each node of "ElementGetParametersValueByValue" that has two inputs: the "ASSET_GROUP" value and a string representing the parameter name (Figure 13). From this process, a list of parameter values is created for each asset belonging to a specific asset group.

In step 4.3, data from all tabs in the temporary spreadsheet file are consolidated into one continuous list (Figure 14). This is done by feeding all data outputs from the "Excel.WriteToFile" nodes (from Figure 11) into a dummy list and ending the transaction with the "Transaction.

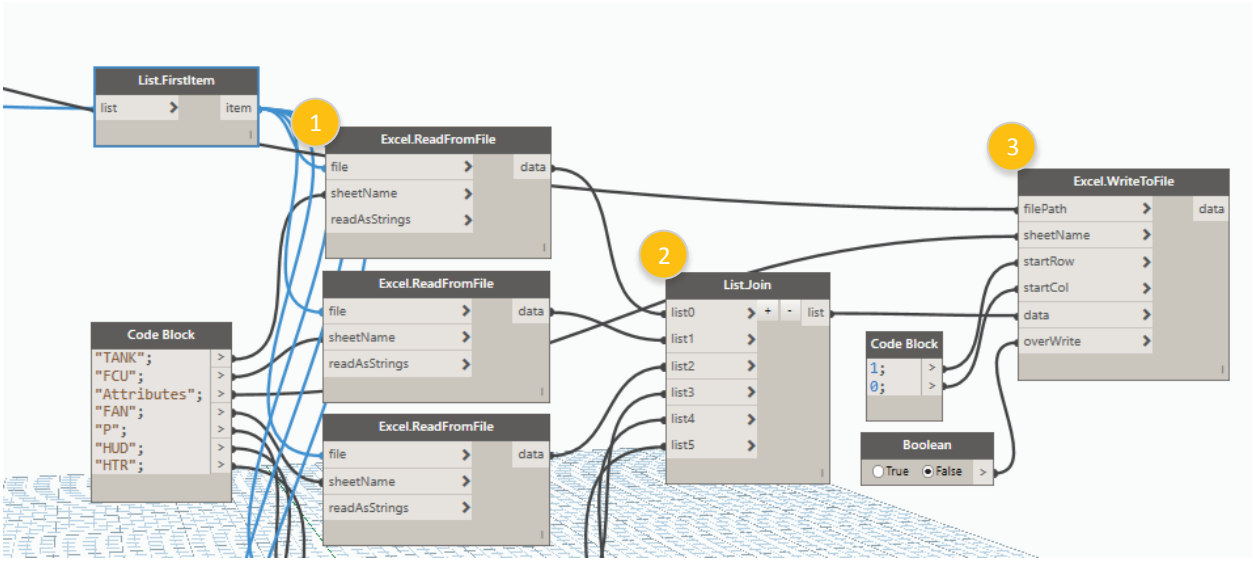

Fig. 15: Steps 4.3 and 4.4: Reading temporary data and writing to final file.

End" node. This allows Dynamo to make sure that all operations are complete. Once all items are fed into the list and the transaction ends, the "Transaction.Start" node can be used to start a new series of functions. This is then fed into a second dummy list that also includes the file path as the first item on the list. Before continuing, only the first item on the list is passed through. The output of the "List.FirstItem" node is the file path.

The file path output is used as the file input for "Excel.ReadFromFile" (Figure 15). Each tab from the temporary sheet is read and consolidated into a single list. The "ListJoin" function allows the data from multiple tabs within the temporary spreadsheet to be compiled into one master list. This allows the data to be written back to the Excel spreadsheet in the ATTRIBUTES tab. The output from the "List.Join" node is the data input for the "Excel.WriteToFile" node.

By the end of step 4.4, the Excel file is generated. Data can then be linked to the FM system using various methods including linking the data to the FM system's SQL Database (AiM) using tools such as Pentaho (www.pentaho.com).

5 Discussion

Facility managers continue to face challenges with identifying information in a timely manner once new construction is complete to properly maintain the buildings. Facility information delivered at the end of the project is often incomplete, inaccurate, or not readily available requiring significant resources and manpower to populate the FM system. To address the issues and to create a more efficient workflow, many owners have developed BIM to FM standards and guidelines to document life cycle information generated during the design and construction phases. While many of these standards define the information requirements for each owner, at present linking the information that is captured in a model, an excel spreadsheet, or any other format directly to the owner's CMMS remains a challenge. Many CMMS systems have different data linking requirements that may not be directly compatible with the format in which the data are captured. Therefore, a proprietary workflow may sometimes be required.

The facility department of the academic institution studied in this case study uses AiM by Assetworks as the CMMS system for tracking and managing their assets across their campus buildings. This system does not support direct linking of data from a Revit BIM model. Currently, and following the completion of new construction or renovation projects, the facility owner populates the spreadsheet manually for direct linking of the data to the CMMS. To allow for direct linking, the data must be organized in the spreadsheet using a propriety format with data distributed in six tabs as described in Section 2. Although a Revit model submission is not currently required by design and build teams performing work on campus, the facility owner plans to require this as a deliverable in the near future. To allow for moving the data from the model to the CMMS, the research team explored various approaches and workflows to move the data from the BIM to AiM.

Due to the desire to have an automated process and the nature of the existing processes, Dynamo was explored as a means to create a customized workflow for the data. The research utilized a case study to present an automated solution for linking FM data from BIM to the AiM utilizing Dynamo within Revit. The research focused on developing a Dynamo work that would automatically generate the Excel spreadsheet with the required proprietary format. Once generated, the spreadsheet can be linked to AiM using the current workflow adopted by the facility owner.

6 Conclusion

The process that is discussed in this article to collect data from a Revit model and export them in a more automated processes for use for FM addresses issues of timeliness and efficiency of information transfer to support FM and the use of a CMMS. The process also allows the owner to maintain the use of its current CMMS system where other proprietary methods are built into software specific solutions. Tedious manual processes are replaced with automated solutions to help cut down on time and potential for human error in the process.

On completion of this work, we summarize some findings as follows:

- The results obtained in the output spreadsheet were checked. The format of the spreadsheet and values of the data extracted were validated to be correct.
- The workflow developed in Dynamo will work with any future asset identified.
- Manual manipulation of BIM data to populate the FM system is timely and subject to human error. The Dynamo workflow helps eliminate the errors and cut the time of the process to populate the FM system with data.
- Once expanded to include all assets, the customized workflow will allow for time savings of what used to take months now taking minutes to execute with the combination of collecting data throughout the life cycle and automating the data population process.
- Dynamo allowed for customizing the workflow to the organization's information needs with the need to reformat existing databases of information for other buildings they are managing. The current FM systems are still usable, meaning that the data for all of the buildings can be housed within the same system. If other technological solutions were adopted, the data would be held separately in two systems or data from the old system would need to be migrated to the new system.

One main limitation to the current approach using Dynamo is that the data transfer process is one-directional. Data extracted from the BIM model are used to update the CMMS. However, any changes made in the CMMS will not be reflected in the BIM model and have to be updated manually. The authors believe that with additional coding, the data exchange process can be bidirectional.

Future research will consider two other areas:

1. Examine the feasibility of modifying the Dynamo workflow to link the Revit BIM and the CMMS SQL database. This would provide a distinct operational advantage through providing a bidirectional link between Revit and the CMMS allowing for data transfer in both directions: model to CMMS and CMMS back to the model, hence allowing to maintain the geometric representation of the building with updated asset information. By keeping the data path open in both directions, the owner can have a record model that is up to date for future use, and they should see value in it.
2. Incorporate other record data information in the form of unstructured data that are currently not addressed.

Declaration of Conflicting Interests

The authors declare that they have no conflicts of interest.

References

Alnaggar, A., & Pitt, M. (2018). Towards a conceptual framework to manage BIM/COBie asset data using a standard project management methodology. *Journal of Facilities Management*, doi: 10.1108/JFM-03-2018-0015

Anderson, A., Marsters, A., Dossick, C. S., & Neff, G. (2012). Construction to operations exchange: Challenges of implementing COBie and BIM in a large owner organization. In: *Construction Research Congress 2012*, ASCE, Reston, VA.

Beach, T., Petri, I., Rezgui, Y., & Rana, O. (2017). Management of collaborative BIM data by federating distributed BIM models. *Journal of Computing in Civil Engineering, 31*(4), pp. 04017009.

Bercerik-Gerber, B., Jazizadeh, F., Li, N., & Calis, G. (2012). Application areas of data requirements for BIM-enabled facility management. *Journal of Construction Engineering and Management, 138*(3), pp. 431-442.

Borhani, A., Lee, H.W., Dossick, C.S., Osburn, L., & Kinsman, M. (2017). BIM to Facilities Management: Presenting a Proven Workflow for Information Exchange. *ASCE International Workshop on Computing in Civil Engineering*, doi:10.1061/9780784480823.007

Cavka, H. B., Staub-French, S., & Poirier, E. A. (2017). Developing owner information requirements for BIM-enabled project

delivery and asset management. *Automation in Construction, 83*(2017), pp. 169-183.
Chen, W., Chen, K., Cheng, J. C. P., Wang, Q., & Gan, V. J. L. (2018). BIM-based framework for automatic scheduling of facility maintenance work orders. *Automation in Construction, 91*(2018), pp. 15-30.
Di Iorio, C. (2013). BIM to FM – realities, goals, challenges and future. BIM MEP AUS.
Dias, P., & Ergan, S. (2016). The need for representing facility information with customized LOD for specific FM tasks. In: *Construction Research Congress 2016*, ASCE, Reston, VA.
Dixit, M. K., Venkatraj, V., Ostadalimakhmalbaf, M., Pariafsai, F., & Lavy, S. (2010). Integration of facility management and building information modeling (BIM): A review of key issues and challenges. *Facilities, 36*(7/8), pp. 455-483.
Dynamo. (2017). Dynamo BIM. Available at www.dynamobim.org [Retrieved on 12 December, 2017].
Edirisinghe, R., London, K. A., Kalutara, P., & Aranda-Mena, G. (2017). Building information modelling for facility management: Are we there yet? *Engineering, Construction and Architectural Management, 24*(6), pp. 1119-1154.
Fallon, K., & Palmer, M. (2006). *Capital Facilities Information Handover Guide, Part 1*. National Institute of Standards and Technology, Gaithersburg, MD.
Gallaher, M. P., O'Connor, A. C., Dettbarn, J. L., Jr., & 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.
Halmetoja, E. (2019). The conditions data model supporting building information models in facility management. *Facilities, 37*(7/8), pp. 484-501.
Heaton, J., Parlikad, A. K., & Schooling, J. (2019). A building information modelling approach to the alignment of organizational objectives to asset information requirements. *Automation in Construction, 104*(2019), pp. 14-26.
Hosseini, M. R., Roelvink, R., Papadonikolaki, E., Edwards, D. J., & Parn, E. (2018). Integrating BIM into facility management: Typology matrix of information handover requirements. *International Journal of Building Pathology and Adaptation, 36*(1), pp. 2-14.
Kassem, M., Kelly, G., Dawood, N., Serginson, M., & Lockley, S. (2015). BIM in facilities management applications: A case study of a larger university complex. *Built Environment Project and Asset Management, 5*(3), pp. 261-277.
Keady, R. (2013). *Equipment Inventories*. Wiley, Hoboken, N.J.
Marchese, J., & Rudderow, C. (2013). The Power of 3D: Using BIM for facility management. Available at www.areadevelopment.com/AssetManagement/Q2-2013.
Miettinen, R., Kerosuo, H., Metsala, T., & Paavola, S. (2018). Bridging the life cycle: A case study on facility management infrastructures and uses of BIM. *Journal of Facilities Management, 16*(1), pp. 2-16.
National Institutes of Building Science (NIBS). (2019). United States National CAD Standard – V6. National Institute of Building Sciences building SMART alliance. Available at: https://www.nationalcadstandard.org/ncs6/. [Retrieved on 7 August, 2019].
Parn, E. A., Edwards, D. J., & Sing, M. C. P. (2017). The building information modelling trajectory in facilities management: A review. *Automation in Construction, 75*(2017), pp. 45-55.
Parn, E.A., & Edwards, D. J. (2017). Conceptalising the FinDD API plug-in: A study of BIM-FM integration. *Automation in Construction, 80*(2017), pp. 11-21.
Pishdad-Bozorgi, P., Gao, X., Easman, C., & Self, A. P. (2018). Planning and developing facility management-enabled building information model (FM-enabled BIM). *Automation in Construction, 87*(2018), pp. 22-38.
Sabol, L. (2008). Building information modeling and facility management. In: *Proceedings, IFMA World Workplace*, IFMA, Dallas, TX.
Sadeghi, M., Mehany, M., & Strong, K. (2018). Integrating building information models and building operation information exchange systems in a decision support framework for facilities management. In: *Construction Research Congress 2018*, ASCE Reston VA, pp. 770-779.
Salmon, J. (2013). Built-BIM to FM, What Owners Want. Available at www.nosilos.com/J25.
Thabet, W., & Lucas, J. (2017a). Asset data handover for a large educational institution: Case-study approach. *Journal of Construction Engineering and Management, 143*(11), pp. 05017017.
Thabet, W., & Lucas, J. (2017b). A 6-step systematic process for model-based facility data delivery. *Journal of Information Technology in Construction (ITcon), 22*, pp. 104-131.
Thabet, W., Lucas, J., & 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.
Wijekoon, C., Manewa, A., & Ross, A. D. (2018). Enhancing the value of facilities information management (FIM) through BIM integration. *Engineering, Construction and Architectural Management*, doi: 10.1108/ECAM-02-2016-0041.
Yalcinkaya, M., & Singh, V. (2019). VisualCOBie for facilities management: A BIM integrated visual search and information management platform for COBie extension. *Facilities, 37*(7/8), pp. 502-524.

Dr. Walid Thabet is the William E. Jamerson Professor in the Department of Building Construction. He is a faculty member of Virginia Tech for more than 22 years and has been teaching and developing construction and construction management courses at the undergraduate and graduate levels. His current research agenda focuses on virtual design and construction (VDC), and construction innovation and BIM for facility management (BIM-FM). Thabet leads the Virtual Facilities Research Lab (VFRL), offering emerging CRE leaders hands-on experience in technology development and practical application as well as access to industry through both corporate and academic partnerships. Along with the academic impact that the VFRL has on curriculum and student support, the lab collaborates with industry partners on external projects. Recent projects involved collaborative research efforts with several A/E/C industry partners to expand knowledge of building information modeling and explore emerging software

tools and technologies including augmented reality technologies.

Dr. Jason Lucas is an Associate Professor of Construction Science and Management at Clemson University where he is involved in developing curriculum and teaching at both the undergraduate and graduate levels. He received his Ph.D. and master's degrees in construction from Virginia Tech and holds a Bachelor of Architecture degree from New Jersey Institute of Technology. Lucas's research interest includes the use of BIM to support life cycle information exchange, residential construction safety, and the use of emerging technology for education.

Sai Srinivasan completed his master's degree in 2018, with major subject construction management from Virginia Tech. He is always passion about application of technology in the construction industry and has worked on multiple projects during his college degree. For the past 2 years, he has been employed with Briegan Concrete LLC as a BIM coordinator where he models the concrete scope of the projects for shoring design, estimation, and scheduling in Tekla/Revit. He is currently working towards helping the company adopt cloud-based integration of three-dimensional models using tablets to streamline the flow of information from office to field.