ISSN 2658-5553

BIM by using Revit API and Dynamo. A review

Divin, N.V.[1]*

[1] Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation

* nikitad93@mail.ru

Keywords:

Building Information Modeling, Automation, Analytics, Visual scripting, Application programming interface, Revit API, Dynamo, Python

Abstract:

Ways of automation of work with Autodesk Revit software package are considered. The user can automate complex or repetitive operations in BIM software. It is possible to create plugins by coding for working with the Revit API and visual programming in the Dynamo module. Tasks for the formation and export of databases for interaction with sensors and electric motors can be realized through Arduino controllers.

1 Введение / Introduction

Появление технологии BIM (Building Information Modeling - Информационное моделирование зданий) в корне изменило процесс проектирования зданий и сооружений, предоставив возможность создавать единую базу данных на основе информационной модели[1], [2], [3], [4]. Это поспособствовало повышению эффективности строительной отрасли за счет облегчения взаимодействия между участниками проекта, уменьшения количества коллизий и работы по исправлению и корректировкам [5], [6], [7].

BIM можно рассматривать как результат эволюции технологии CAD (Computer-Aided Design - Системы автоматизированного проектирования) [8], а эволюция никогда не стоит на месте. В данный момент программы, реализующие BIM-технологию, продолжают совершенствоваться разработчиками: создаются новые инструменты, под постоянно обновляющиеся с развитием технологий задачи пользователей. В частности ведется работа над дополнениями, которые позволяют экономить время путем автоматизации повторяющихся операций.

Однако применение средств по автоматизации работы программ является процессом, требующим определенных навыков, в частности навыков программирования, зачастую отсутствующих у проектировщиков в области строительства, что вызывает сложности при освоении технологии и инструментов автоматизации. Понимая это, разработчики программного обеспечения (ПО) произвели интеграцию визуального языка программирования, или же программирования при помощи сценариев, в BIM программы[9], [10]. Данное решение в результате позволило инженерам, не способным писать коды, создавать алгоритмы – последовательности из сценариев, интерпретированных в виде связанных узлов, содержащих определенный набор команд, которые автоматизируют работу с инструментами программы-основы или же с загружаемыми инструментами.

Это упрощение вовлекло в процесс адаптации программы к современным задачам и создания новых инструментов значительное количество пользователей, которые в результате развития своих навыков алгоритмизации и отдельного изучения базовых принципов программирования могли в дальнейшем перейти к созданию кодов для работы с информацией, находящейся внутри и вне модели. Данный метод бывает необходим при более сложных задачах, стоящих перед проектировщиками, когда большую часть алгоритма можно представить в виде последовательности сценариев, но работу на определенном шаге рациональнее (или принципе возможно только так) выполнить путем написания кода на языке программирования, предоставляющего более широкие возможности по работе с информацией. Для подобных задач

разработчики ПО предоставили возможность кодирования внутри специальных узлов визуального языка программирования.

Как итог на данный момент существует несколько способов автоматизации работы BIM программы, предназначенных для разного уровня сложности поставленной задачи и подготовки пользователя, поэтому каждый из данных способов востребован. К этим способам относится:

1. написание кода на языке программирования в отдельной программе с последующим созданием плагина и загрузкой его в BIM-программу;
2. создание алгоритма на языке визуального программирования внутри BIM-программы с помощью узлов (в том числе программирования на языке Python в специальных узлах);

Целью данной статьи является попытка проанализировать результаты автоматизации работы программного комплекса (ПК) Revit его пользователями, и обозначить какие именно задачи могут решать различные способы автоматизации.

2 Способы автоматизации в Revit / Automation Methods in Revit

В качестве программы, реализующей технологию BIM, рассмотрен ПК Revit, так как данный программный комплекс наиболее распространен и имеет интерфейс прикладного программирования Revit API, за счет чего позволяет интегрировать плагины созданные на любом языке программирования, поддерживающем платформу.NET, таком как VB.NET, C#, Python и других. Также данный ПК имеет модуль для визуального программирования Dynamo, позволяющий работать с моделью в реальном времени. Кроме того открытое сообщество пользователей и разработчиков, постоянно создает новые пакеты инструментов и обменивается ими в сети, что позволяет начинающим пользователям получать наработки более опытных и использовать новейшие инструменты, которые возможны на определенном этапе развития технологии.

2.1 Автоматизация при помощи кодирования / Automation using the coding

Базовым методом создания новых инструментов (автоматизации существующих) предоставляющим наиболее широкий выбор функций и команд остается кодирование [11], [12], [13]. Изначально разработчиками были созданы пакеты инструментов в виде закодированных команд, которые выполняли ту или иную операцию, например, создавали арматурные стержни в модели железобетонной плиты по заданным характеристикам. Интерфейс Revit API в процессе работы плагина считывал эту команду, идентифицировал ее и выполнял необходимые построения. К примеру, для преобразования некоторой линии в модели в арматурный стержень с заданными характеристиками, разработчиками была создана команда на языке программирования C#:

Rebar.CreateFromCurves(Autodesk.Revit.DB.Document, FamilyInstance, RebarBarType, RebarHookType)

Характеристиками, которые запрашивает команда (в скобках) является название проекта, в котором будет выполняться построение (Autodesk.Revit.DB.Document), вид арматурного стержня (прямой или имеющий определенную изогнутую форму) (FamilyInstance), тип арматурного стержня (характеризует диаметр, радиус загиба и материал арматуры) (RebarBarType) и тип загиба стержня в начале и конце (RebarHookType).

Данная команда отдельно выглядит вполне понятной для пользователя и быстро реализуется интерфейсом прикладного программирования. Однако чтобы вписать данную команду в программный код и обеспечить ее необходимыми корректными данными, извлеченными из модели, необходимо обладать навыками написания кодов, что не является обязанностью строителей-проектировщиков, поэтому это вызывает трудности. Изучение пользователем прикладного использования программных языков в подобных случаях встречает многочисленные сложности, которые делают невозможным применение данного метода повсеместно [14]. Однако при наличии навыков кодирования, этот метод носит наиболее результативный характер, как с точки зрения скорости выполнения алгоритма, так и вычислительной нагрузки на компьютер и сам программный комплекс.

2.2 Автоматизация в среде Dynamo с применением стандартных узлов / Automation in Dynamo using standard nodes

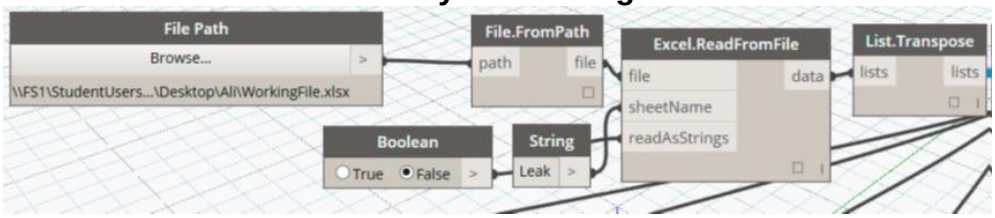

Рисунок 1. Пример алгоритма по извлечению информации из файла Excel в Dynamo [15]

Визуальные языки программирования не имеют столь высоких требований для работы с ними[16], [17]. Алгоритм реализуется при помощи соединенных в необходимом порядке узлов, которые интерпретируют определенные команды, заранее заложенные в них разработчиками. Эти команды могут выполнять построения в модели, расчеты, анализировать информацию, извлекать данные, как из модели, так и из файлов, находящихся на компьютере, а также выводить их обратно и делать многое другое[12], [18]. Иными словами данный способ предоставляет обширные возможности, которые могут быть достаточны для многих поставленных задач. При этом способ обладает доступностью для большинства пользователей, что привлекает к нему все больше людей[19]. Однако стоит отметить, что работа сложного алгоритма, насыщенного узлами в Dynamo, в особенности, когда происходит обращение к файлам вне программы, имеет более длительный и тяжелый характер, как для компьютера, так и самого программного комплекса Revit, чем работа такого же алгоритма, выполненного в виде плагина на программном коде. Это может являться проблемой, в особенности, когда происходит непрерывный обмен данными, например, с датчиками интенсивности солнечного света, размещенными на фасаде здания.

Также он не обладает всеми возможностями прямого кодирования, например, в нем невозможно задавать зацикленные структуры по перебору определенных значений, а набор стандартных узлов освещает только наиболее распространённые команды, что иногда является недостаточным. К примеру, представленная выше команда по преобразованию линии в арматурный стержень не имеет аналога в стандартных узлах Dynamo. Это означает, что рядовой пользователь, не знающий программирования, не сможет решить ряд задач, связанных с армированием.

Выходом из данной ситуации может являться обращение к сообществу пользователей Revit. Пользователи делятся своими наработками по различным направлениям автоматизации, создавая собственные инструменты [20]. При этом используется кодирование на языке программирования Python, код набирается в специальных узлах Dynamo, называемых «Python Script»[21]. В итоге пользователь без знаний программирования может скачать необходимый пакет инструментов, созданный другими людьми и решить поставленную задачу с помощью них, дополнив стандартные узлы.

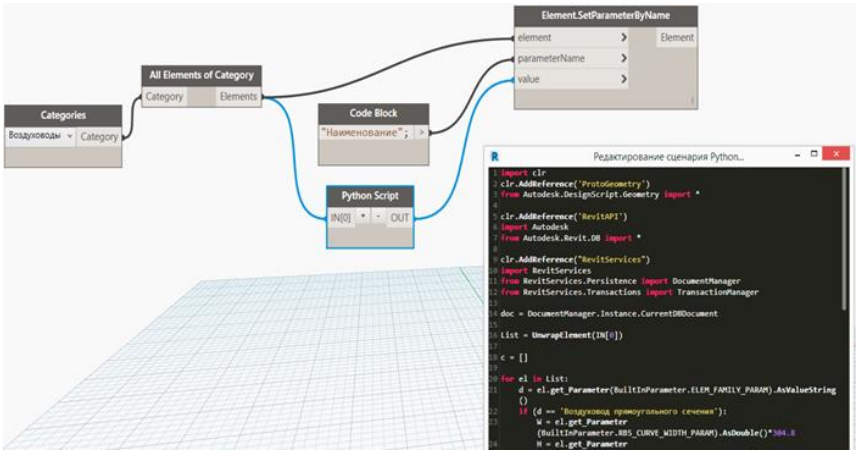

Рисунок 2. Пример алгоритма с применением узла «Python Script» в Dynamo [22]

Благодаря данному узлу возможно выполнять создание отдельных инструментов или же прописывать весь алгоритм в виде кода на Python целиком в узле «Python Script» в пространстве

Dynamo, создавая вспомогательные узлы лишь для ввода и вывода данных[23]. Фактически решение разработчиков предоставить возможность кодирования внутри модуля Revit, позволило заместить способ отдельного создания плагинов в специализированных программах, так как благодаря тому, что алгоритм создается в Dynamo, его работу с моделью можно проверять одновременно с написанием кода, а также использовать уже готовые узлы, для определенных этапов алгоритма, что является преимуществом. Однако так как данный способ основывается на работе в среде Dynamo, его работа также происходит медленнее, чем у плагина с тем же функционалом, созданного кодированием.

3 Обзор статей / Review of articles

3.1 Работа с обширными базами данных / Work with extensive databases

Проблема обмена данными между различными программными комплексами, а также в целом сложность вывода информации, описывалась ранее во многих источниках. Форматом, который должен реализовывать передачу данных между ПК является IFC (Industry Foundation Classes - отраслевые базовые классы) - это открытая и стандартизированная модель данных, предназначенная для обеспечения взаимодействия между программами[24,25]. Тесты многих исследователей показали слабую эффективность данного метода передачи[26–28]. Они зафиксировали значительное количество случаев искажения и потери информации, как на уровне модели, так и на уровне атрибутов элементов модели. В связи с этим некоторые пользователи разрабатывают свои алгоритмы по автоматизированному формированию файла с необходимой информацией, извлекаемой из модели, для передачи в другую программу.

3.1.1 Автоматизация путем разработки плагина / Automating with developed plugin

Пример решения данной проблемы описывается в статье [29], где на языке программирования C# создается плагин в программе Microsoft Visual Studio, выполняющий работу по созданию и обработке файла расширенной базы данных, который может являться заменой файлу IFC. В этом файле происходит запись последовательного изменения любой характеристики проекта и в результате есть возможность просмотреть историю изменений определённого элемента или же модели в целом. Также данный файл расширенной базы данных BIM модели доступен любому пользователю, что позволяет обеспечить совместную работу по редактированию модели, а также передачу данного файла взамен файла формата IFC. Таким образом, информация о модели, в том числе та, что не способна корректно передаться через IFC формат, передается через созданный плагином файл и может быть считана другой программой или пользователем.

Подобный способ решения данной проблемы представлен в статье [30], где также разрабатывается плагин на языке C#. Алгоритм плагина создает файл для экспорта информации из модели Revit, систематизируя необходимые данные в определённой последовательности и записывая их в текстовый файл. В этот файл попадает информация о геометрических характеристиках элементов, а в частности их аналитических моделей, а также данные о жесткостных показателях с учетом материалов и геометрии сечения элементов. Все это необходимо для последующего расчета в программе структурного анализа, базирующейся на методе конечных элементов, которая также разработана авторами статьи.

3.1.2 Вывод о работе с обширными базами данных / Conclusion about working with extensive databases

Эти два примера объединяет то, что в обоих случаях происходит формирование файла базы данных для экспортирования значительных объемов информации из модели Revit, который выполняет те же функции, что и файл формата IFC, но при этом не теряет необходимой информации. Подобные задачи по работе с обширными базами данных информационной модели для формирования полноценного файла экспорта, требуют большого количество разноплановых инструментов, которые могут предоставить только методы кодирования на программных языках. Структурировать сложный файл, способный заменить файл формата IFC, при помощи алгоритма визуального программирования слишком сложно, о чем также свидетельствует факт, что публикаций по решению данной проблемы с помощью Dynamo не было найдено.

3.2 Работа с базами данных узкого назначения / Work with narrow-purpose databases

Стоит учесть, что формирование столь обширной базы данных, которая была бы способна заменить файл формата IFC, требуется редко, так как в целом такие объемы информации обычно нужны при передаче модели между программными комплексами. Зачастую пользователям необходимо сформировать для вывода массив данных касательно только определенных элементов модели, или же внести некоторую информацию в модель для хранения, поэтому алгоритмы по экспорту и импорту данных и формировании небольших баз данных являются более популярными и востребованными, а также более простыми для выполнения.

3.2.1 Автоматизация путем разработки плагина / Automating with developed plugin

В статье [31] говорится о новом подходе, использованном авторами по интерактивному взаимодействию пользователя с моделью на этапе эксплуатации объекта. Плагин, загружаемый в Revit создает новые клавиши на панели инструментов, несущие функционал, связанный с работой с базой данных оборудования, которое установлено в реальном здании с привязкой к смоделированному оборудованию в пространстве модели. Информация в базе данных дополняется данными, связанными с сервисным обслуживанием, вносимыми со временем пользователем. Также предусматривается вывод данной информации в формате, удобном пользователю, что позволяет предоставить владельцу здания максимум инструментов по использованию информационной модели во время периода эксплуатации здания. Данная концепция является целью, стоящей перед BIM технологией на ближайшие годы (использование BIM модели при эксплуатации объекта), поэтому данный проект получил значительную поддержку ряда компаний Англии и Финляндии, о чем также говорится в статье.

Решением по выдаче информации при помощи плагина также пользовались авторы статьи [32]. Статья посвящена разработке строительных лесов из стальных труб при работе закодированного алгоритма, написанного с использованием языка программирования C#. Revit API автоматизирует моделирование лесов, подбирая их наиболее рациональные характеристики и сечения, что позволяет пользователю в кратчайшие сроки выдать проект. Формирование и извлечение технической информации о строительных лесах происходит также при помощи плагина в автоматизированном режиме.

В статье [33] описывается работа плагинов по демонстрации на внутренних поверхностях помещений значений тепловой энергии и освещенности. Пользователь создает модель здания при помощи специальных семейств, загружаемых из библиотек Modelica Buildings, разработанной Национальной лабораторией Лоуренса Беркли для теплового моделирования и Radiance, разработанной DAYSIM для моделирования дневного освещения. Далее при помощи разработанных авторами плагинов Revit2Modelica и Revit2Radiance происходит экспорт ключевых элементов созданной модели в специальные программы для анализа распределения тепловой энергии (LBNL Modelica Buildings) и света внутри помещений (DAYSIM). При этом, так как используются специально созданные для решения этой задачи семейства, идентификация в программах анализа происходит корректно, и после работы в расчётных программах, плагины производят импорт результатов расчета обратно в Revit. В итоге поля распределения тепловой энергии и освещенности, полученные по данным расчета, могут быть визуализированы на внутренних поверхностях помещений, что значительно упрощает восприятие этих характеристик, а также позволяет объединить базы данных архитектурной модели и энергетического анализа для дальнейшего использования.

3.2.2 Автоматизация путем визуального программирования в Dynamo / Automation through visual programming in Dynamo

Как было сказано ранее, создание замены обширному файлу формата IFC при помощи Dynamo является слишком сложным. Однако многими пользователями представлен пример реализации алгоритма по экспорту и импорту определенных данных в таблицы программы Excel. К примеру, в статье [34] описывается алгоритм в Dynamo, который обеспечивает обмен информацией касательно состава ограждающих конструкций различных комнат замоделированного здания, расположения окон, местоположения и габаритов затененных областей и т.п. Эта информация используется в дальнейшем пользователем для оценки ряда показателей среды внутри помещений, касающихся теплового, визуального, атмосферного и акустического комфорта в соответствии с установленными стандартами. В итоге алгоритм

сокращает время, необходимое для ручного ввода данных характеристик, что было подтверждено тестами, описанными в статье.

Авторы подобной работы [35], также использующей Dynamo как способ извлечения определенной информации из модели и передачи в Excel, ставили цель - подбор наиболее рационального состава ограждающей конструкции с экономической точки зрения. В данном случае пользователь заранее заносит данные характеристик материалов и конструкций в отдельные графы созданных параметров элементов модели. Затем, после построения модели здания, алгоритм Dynamo производит сбор этих данных: размеры стены, окон, показатели поглощения солнечного света, ориентация стены по сторонам света, стоимость строительства и т.д., формирует базу данных по стене и экспортирует ее в файл формата Excel для дальнейшей оценки теплотехнических свойств в программе MatLab. В результате пользователь может вычислить, насколько экономически выгоден выбранный вариант устройства стен для заданного климатического района и ориентации стены по сторонам света, и подобрать наиболее рациональный из возможных.

Аналогичный пример организации работы по обмену данными о показателях среды в помещениях через среду Dynamo, описывается в статье [36]. Пользователь заранее создает отдельный параметр в экземплярах элементов, который выполняет роль хранилища информации. При работе алгоритма этому параметру присваивается определенная информация, извлекаемая из файла Excel, при этом алгоритм Dynamo идентифицирует какому именно элементу присвоить ту или иную информацию и производит присвоение. В итоге база данных информационной модели здания дополняется данными о характеристиках окружающей среды.

Схожей методикой пользуются авторы статьи [37]. Проблема, которую решают авторы, заключается в том, что при импорте и расшифровке файла формата IFC, ПК Revit не корректно идентифицирует определенные параметры элементов, сохраняя их как отдельные параметры, хотя должен присваивать их значения уже существующим параметрам. В связи с тем, что информация загружается в проект из IFC, но ее представление некорректно, могут возникнуть проблемы, например, с выводом спецификаций. Решение данной проблемы заключается в копировании значения из неверно идентифицированного параметра в тот параметр элемента модели, который используется в Revit. При этом, так как подобное копирование может занимать длительное время, авторы создали алгоритм в Dynamo, который производит копирование значений параметров у необходимых элементов автоматически (Рис.3).

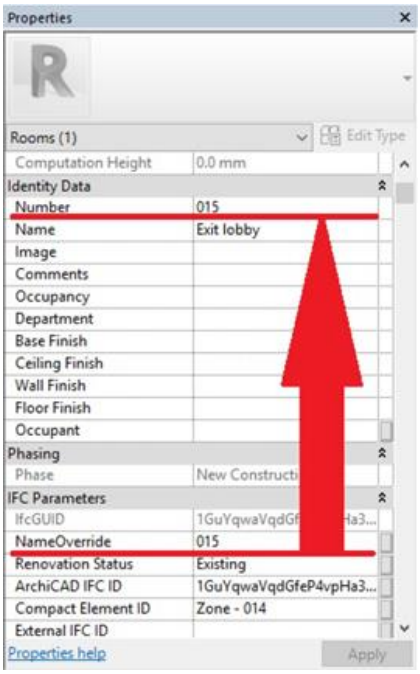

Рисунок 3. Принцип работы алгоритма по автоматическому копированию значения параметра в Dynamo [37]

В статье [38] описывается создание базы данных в файлах формата Excel по каменному строительству с разработкой алгоритма в модуле Dynamo для работы с ней. Вопрос

моделирования в среде Revit стен из каменной кладки является недостаточно проработанным, так как при моделировании ведется создание модели стены целиком, то есть характеристикам штучных материалов, из которых она состоит, не уделяется особое внимание. При этом в зависимости от производителя и вида кладки, характеристики камней могут сильно меняться, что не предусматривается в информационной модели. В данной работе была поставлена цель систематизировать каменные мелкоразмерные элементы, из которых возможно построение кладки в виде единой базы данных MUD (Masonry Unit Database). Алгоритм, созданный в модуле Dynamo, позволяет подключаться к базе данных MUD и связывать модели каменных стен в Revit с определенным видом камней, модели которых создавались на основе данных из таблиц Excel, заполненных заранее (Рис.4). При этом для построения моделей каменных материалов был написан код в узле «Python Script» для работы с атрибутами элементов.

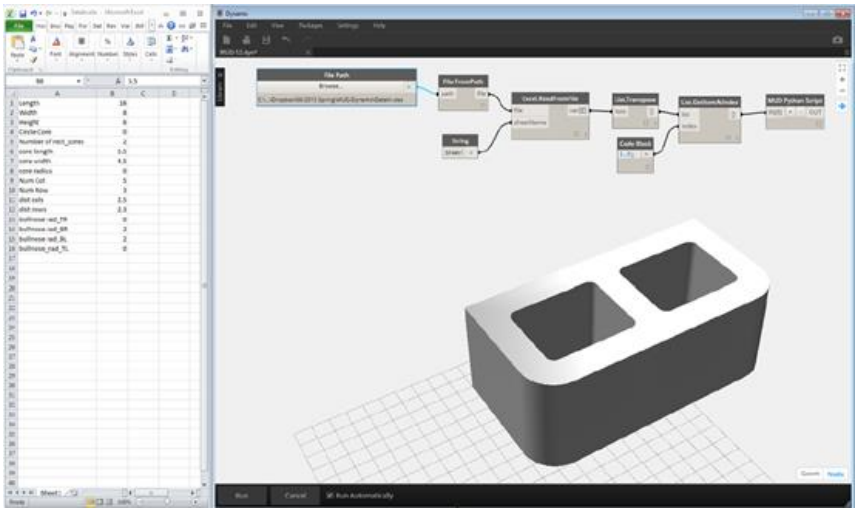

Рисунок 4. Алгоритма по моделированию каменных материалов с применением узла «Python Script» в Dynamo [38]

Пример решения задачи по передаче информации об обслуживании и эксплуатации здания можно найти в работе [39]. В статье описывается алгоритм на Dynamo, при помощи которого производится копирование из внешних таблиц и занесение в модель информации о проведенном обслуживании и ремонте элементов. Таким образом, формируется единая информационная модель здания, получающая информацию не только во время проектирования, но и во время эксплуатации здания. При этом стоит отметить, что представленный в статье алгоритм также способен выводить запрашиваемую пользователем накопленную информацию об обслуживании в виде таблиц.

Статья [40] направлена на создание алгоритма в Dynamo по автоматизированному формированию календарных графиков для зданий с железобетонным каркасом. Алгоритм сначала извлекает необходимые данные из информационной модели здания, включая размеры элементов, их количество, информацию об их положении в пространстве, принадлежность к этажам, материалы и прочее, и формирует сводную базу данных обо всех элементах, подлежащих последующему учету в календарном плане (Рис.5). Затем рассчитывается длительность бетонирования отдельных элементов, и применяются заложенные в алгоритм правила по составлению последовательности бетонирования элементов конструкции, основанные на учете сначала этажа, к которому принадлежит элемент, а затем типа этого элемента (плита, колонна, стена). В итоге при экспорте данных в MS Project, формируется календарный график, который может показать, какие сроки требуются для выполнения элементов железобетонного каркаса.

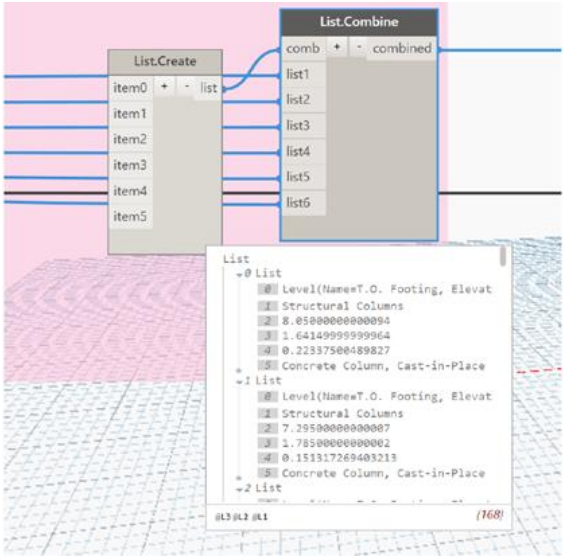

Рисунок 5. Фрагмент алгоритма, демонстрирующий поэлементно собранную информацию [40]

В статье [41] авторы также поднимают вопрос бетонирования железобетонного каркаса, основываясь на информации из модели Revit, но здесь целью является подбор опалубок для бетонирования элементов. Алгоритм Dynamo извлекает данные о размерах, количестве и положении в пространстве выбранных пользователем железобетонных элементов. Затем, применяя заложенные в алгоритм правила подбора опалубки и базу данных модульных опалубок с информацией о доступных размерах и типах, производит подбор для каждого элемента. В итоге пользователь получает таблицы, в требуемом ему виде с информацией о том, какие типы опалубок требуются для выполнения бетонирования конкретных железобетонных элементов (Рис.6).

Column ID	Column Base Level	Family Name	Column Width [m]	Column Depth [m]	Column Height [m]	Concrete Volume [m3]	Formwork System: L
T-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U(-648)-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U(-648)-3	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U-6	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U-9	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U(-648)-11	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U(-648)-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
T-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
S-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
R-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
Q-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70

Рисунок 6. Пример вывода спецификации опалубочных элементов в результате работы алгоритма Dynamo по подбору опалубки [41]

3.2.3 Вывод о работе с базами данных узкого назначения / Conclusion about working with narrow-purpose databases

Приведенные выше примеры показывают разноплановость подобных задач по формированию и работе с базами данных, связанных с определенными элементами модели. При этом с учетом количества найденных примеров алгоритмизации, выполненных при помощи кодирования и визуального программирования, автоматизация при помощи Dynamo является более популярным методом по решению подобных задач. Связано это с более низким уровнем сложности задач, а соответственно и меньшей потребностью в инструментах и структуризации данных.

3.3 Работа с информационной моделью / Work with an information model

Помимо обработки баз данных, работа с которыми неразрывно связана с формированием информационной модели, среди пользователей распространены задачи, связаны с работой по изменению информационной модели здания, более быстрому построению сложной геометрии, а также оценки соответствия характеристик модели требованием норм.

3.3.1 Автоматизация путем разработки плагина / Automating with developed plugin

В статье [42] авторы используют Revit для создания модели деревянной башни Инсянь. При этом моделирование особого элемента деревянного каркаса под названием «DouGong», который является традиционным компонентом подобной архитектуры и выполняет роль своеобразной капители, встретило определенные трудности, связанные со сложностью исполнения и

параметризации модели. В результате авторы для решения данной проблемы разработали плагин на платформе.NET, который позволяет создать модель DouGong путем задания геометрии каждого из элементов конструкции в отдельности в отдельном окне (Рис.7), открываемом при запуске плагина. Данный алгоритм генерации подобной сложной структуры имеет ограниченный функционал, но значительно ускоряет работу по моделированию DouGong, что также описывается в статье.

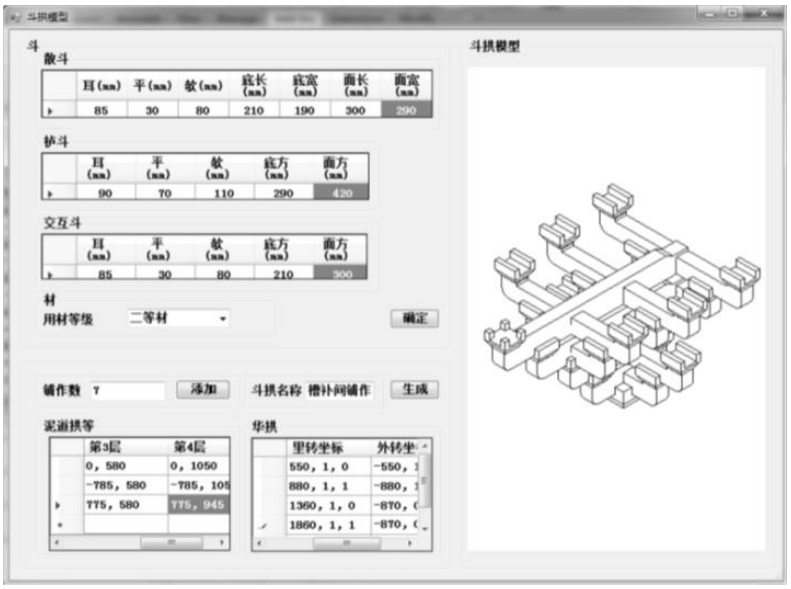

Рисунок 7. Окно плагина по работе с моделированием сложной структуры [42]

В статье [43] применяется плагин, написанный на языке программирования C#, и авторы пытаются автоматизировать создание сборных железобетонных элементов. В виду того, что сборные элементы обладают структурной независимостью и не изменчивостью относительно других объектов, контактирующих с ними, значит, их армирование является постоянным. В работе предлагается плагин, предоставляющий возможность ускоренного наполнения арматурными стержнями сборных балок, плит и колонн. При этом плагин предоставляет собственный интерфейс, который открывается в отдельном окне, при запуске плагина (Рис.8). В итоге пользователь может быстро решить задачу моделирования многочисленных сборных элементов, требующихся в проекте с проработанной структурой материалов и арматурных стержней в теле железобетона.

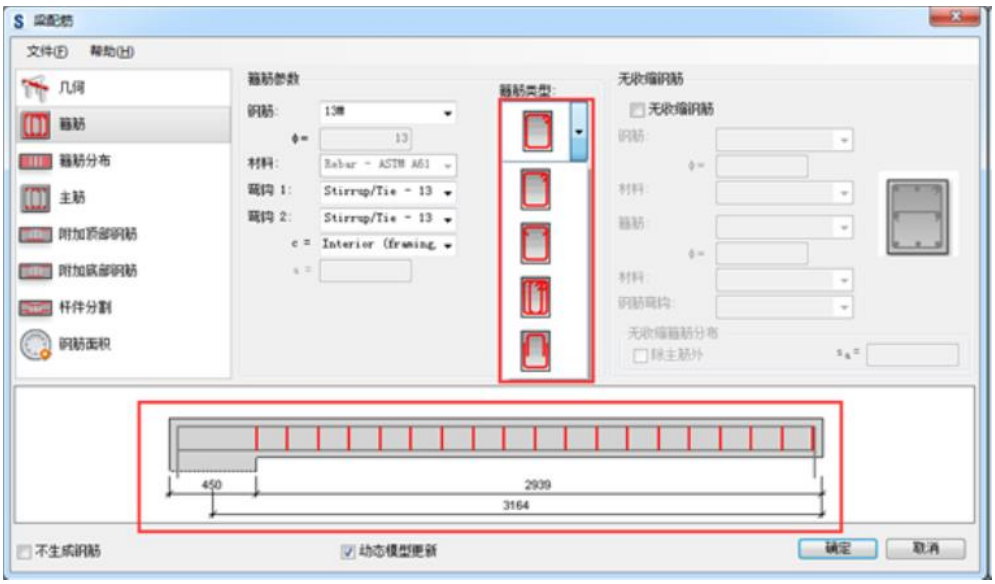

Рисунок 8. Окно плагина по работе с армированием сборного железобетонного элемента [43]

Еще один пример решения задач армирования железобетонных элементов представлен в статье [44]. Плагин, написанный также на языке программирования C# извлекает информацию из 2D-чертежей армированного сечения конструкции и идентифицирует в сечении арматурные стержни. Затем происходит анализ выбранного пользователем элемента в модели и, если сечение элемента и сечение, представленное в 2D-чертеже, совпадают и возможно провести армирование по предоставленным данным, то плагин открывает окно редактора, где пользователь может дополнительно скорректировать характеристики стержней перед моделированием. Далее при подтверждении пользователем правильности введенных характеристик, алгоритм производит моделирование арматурных стержней, что заметно ускоряет работу по армированию конструкций по сравнению со стандартными инструментами Revit.

В данный момент повышается внимание к использованию ресурса солнечного света и возможности создавать затенение соседних участков и зданий в определенные часы дня, так как, например, если соседнее здание использует фасад с фотоэлектрическими панелями, то строительство здания по соседству с ним и перекрывание солнечных лучей может существенно снизить эффективность применения данной технологии. В связи с этим авторами статьи [45] был предложен алгоритм, реализованный в виде подгружаемого плагина, который моделирует максимальные габариты проектируемого здания, чтобы оно не перекрывало доступ к солнечному свету соседним строениям в определенные часы. Изначально пользователь создает модель участка между зданиями и упрощенные модели самих зданий, демонстрирующие их габариты в виде параметрических элементов, с занесением информации о высоте. Эти элементы в дальнейшем будут определять границы фасадов, по которым будет вестись расчет траекторий лучей света. Затем в рабочем окне плагина (Рис.9) задается информация о сроках возможного затенения в течение дня и максимально допустимая высота затенения соседнего здания. Последний параметр актуален, например, в том случае, если на первых этажах здания, которое будет затеняться, расположены магазины, допускающие возможность создания тени на прилавках для предотвращения порчи продукции при солнечном воздействии. Далее алгоритм формирует две группы плоскостей. Первая группа связана с собственными размерами затеняющего здания. Вторая состоит из плоскостей, которые проходят таким образом, чтобы здание не создавало тень выше требуемой отметки. В результате вырезания части внутреннего пространства образованного первой группой плоскостей, вторая группа создает максимально допустимые размеры затеняющего здания в виде оболочки. В дальнейшем оболочку можно сравнить с габаритами проектируемого здания и оценить будет ли происходить затенение фасада соседних строений или нет.

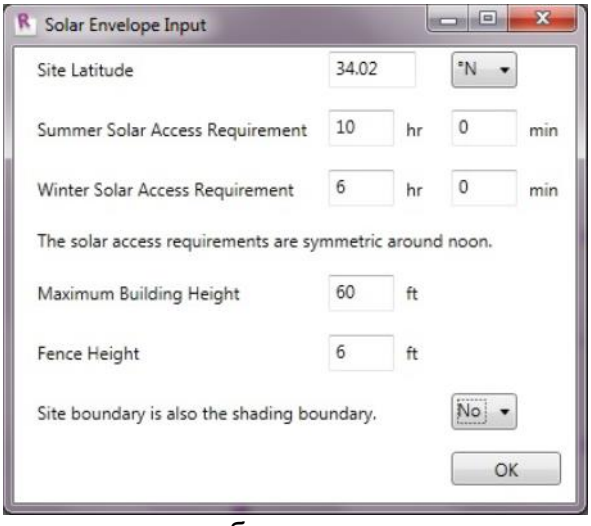

Рисунок 9. Окно плагина по работе с затенением соседних зданий [45]

Проверка модели соответствию определенным требованиям также является довольно распространенной задачей автоматизации BIM программ[46]. Так в статье [47] авторы описывают разработанный плагин, добавляющий в Revit инструменты по анализу помещений модели здания вивария – места, предназначенного для содержания лабораторных животных. К содержанию таких животных предусматриваются определенные требования, такие как минимальные размеры помещений/вольеров, материал стен, наличие светопропускающих элементов, оборудования для

жизнеобеспечения и т.п. В итоге плагин отбирает помещения, помеченные пользователем как предназначенные для того или иного вида животного, и проверяет их характеристики, наличие требуемого оборудования и выводит в отдельном окне отчет о соответствии характеристик заявленным. В дальнейшем отчет можно сохранить и предоставить службе по контролю для доказательства соответствия стандартам проектирования.

Также решение по проверке модели требованиям стандартов представлено в работе [48]. Разработанный плагин анализирует соответствие путей эвакуации людей из здания требованиям пожарным требованиям безопасности. При этом форматируется несколько альтернативных путей, и алгоритм рассчитывает степень опасности каждого из них с учетом длины пути, встречаемых препятствий и характеристик огнестойкости материалов окружающей среды. В итоге пользователь получает отчет с рекомендациями и оценкой здания с точки зрения соответствия нормам пожарной безопасности.

3.3.2 Автоматизация путем визуального программирования в Dynamo / Automation through visual programming in Dynamo

Применение Dynamo также позволяет решать многочисленные задачи по работе с моделью здания. К примеру, в статье [49] описывается алгоритм Dynamo, вычисляющий насколько безопасным является здание для пролетающих птиц. В программу заложены методы вычисления показателя безопасности, которые соотносятся с данными из LEED (The Leadership in Energy & Environmental Design – «Руководство в энергетическом и экологическом проектировании»). Суть методики расчета такого показателя через BIM заключается в том, что пользователь при моделировании использует специальные семейства, в которых создан дополнительный параметр, зависящий от характеристик материала (прозрачности, вероятности создания зеркального эффекта, бликов и т.п.). При запуске алгоритм считывает все элементы, извлекая значение созданного параметра, идентифицирует, на какой высоте располагается элемент (согласно LEED конструкции выше 3 этажа имеют большую опасность для пролетающих птиц) и рассчитывает по формулам значение опасности для птиц данной комбинации материалов для конкретного здания. В итоге алгоритм сравнивает полученные показатели с нормативными из LEED и делает заключение, обеспечивается ли соответствие требованиям или нет.

Задача по автоматизированному построению модели по заданным характеристикам и координатам ставилась перед авторами в статьях [50][51]. В данном случае по задумке пользователь определяет начальные координаты элементов в программе по обработке облаков точек, полученных путем сканирования. Затем эти координаты заносятся в алгоритм Dynamo, и он размещает экземпляры необходимого семейства в заданных местоположениях. Также пользователю доступно проведение редактирования геометрии и характеристик элементов до моделирования их алгоритмом. Все это позволяет ускорить процесс создания геометрии по облакам точек, автоматизировав часть процесса, о чем также говорится в работах.

Попытки расширения возможностей Revit по моделированию дорог представлены в статье [52]. Алгоритм Dynamo обеспечивает построение осевой линии пути по заданным контрольным точкам и затем создает трехмерную модель дорожного покрытия. Характеристики покрытия могут быть разными, в зависимости от выбора пользователя, при этом предусматривается слоистость структуры дороги и возможность задания материалов в отдельности каждому слою на определенном участке. Кроме того при помощи программирования в узлах «Python Script» в алгоритме создан расчётный модуль, который по характеристикам дорожного покрытия и задаваемой пользователем информации о нагрузках на дорогу, прогнозирует какие деформации получит покрытие со временем (Рис.10). Расчет ведется с учетом усталостного растрескивания слоя асфальтовой смеси, вертикальной деформации сжатия верхней поверхности дорожного полотна и возможности растрескивания при низкой температуре.

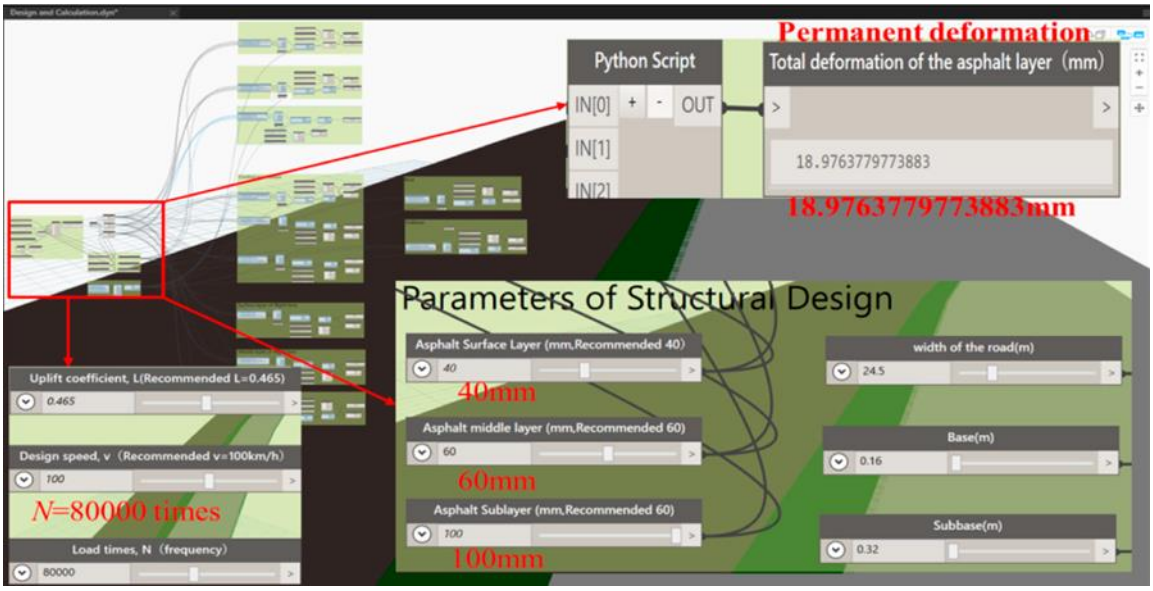

Рисунок 10. Фрагмент алгоритма Dynamo по расчету деформаций покрытия при задании нагрузки на дорогу [52]

Работа [53] по созданию инструментов для ускорения проектирования терминала международного аэропорта Бангалора, описывает несколько проблем, которые удалось решить при помощи алгоритмов на визуальном языке программирования. Dynamo был использован для улучшения качества и контроля процессов проектирования сложной геометрии крыши для аэропорта. Моделирование поверхностей крыши производилось по заданным координатам и математическим зависимостям для последующего сравнения со спроектированной документацией. Это позволило унифицировать ряд параметрических элементов в связи с приближением модели к более простым формам. Также был разработан алгоритм для взаимодействия с командой архитекторов, находящий зависимости между названием помещения в модели Revit, прописанным архитектором на этапе AP, и величиной полезной нагрузки, которая создается в этом помещении согласно нормам. В итоге инженерам не было необходимости задавать в отдельности для каждого помещения значение нагрузки, которое в дальнейшем использовалось для структурного анализа, Dynamo выполнял задание величин автоматически.

В статье [54] предлагается метод оценки экологичности строительства здания с железобетонными элементами путем расчета CUI (Concrete Usage Index - Индекс использования бетона). Данный показатель используется при сертификации проекта здания, предусматривая несколько уровней соответствия экологическим требованиям. Ранее CUI рассчитывался только вручную или при помощи таблиц Excel. Авторы статьи предложили способ, основанный на алгоритме Dynamo, когда автоматически происходит отбор из модели несущих элементов, выполненных из железобетона, суммирование объемов этих элементов и расчет CUI, путем деления просуммированного объема на площадь этажа здания. Данная методика позволяет экономить время на ручном занесении информации в таблицы Excel для последующего расчета, и при этом, обладает достаточной точностью, что показали сравнения с результатами, полученными в Excel, что также отражено в статье.

Примером по анализу наружных поверхностей здания является алгоритм, описанный в статье [65]. В данном случае целью является оценка поверхностей с точки зрения интенсивности воздействия на них солнечной энергии в течение года. Это необходимо для отбора наиболее благоприятных мест для установки солнечных панелей, вырабатывающих электроэнергию, с учетом ориентации поверхностей фасадов здания относительно сторон света и наклона относительно горизонтальной плоскости. Модель формируется с учетом окружающей застройки, которая создана как отдельное семейство по результатам сканирования городской застройки с беспилотного летательного аппарата. Общий файл сканирования был предоставлен городом Монреаль в виде файла системы CityGML, и обрезан, чтобы нести информацию только о ближайшей застройке, способной повлиять на освещение фасадов исследуемого здания.

Алгоритм Dynamo отбирает только наружные поверхностей здания и группирует их в зависимости от принадлежности к элементам по уровням и функции (крыша, несущие и навесные стены разных этажей). Затем алгоритмом производится создание сети точек с шагом 1 метр, по

отобранным поверхностям. Запускается инструмент Revit Solar Analyst из загружаемого модуля Autodesk Insight для моделирования солнечного излучения на поверхностях здания по географическому положению и ориентации здания относительно сторон света. Значение солнечного воздействия рассчитывается для каждой отдельной точки созданной по поверхностям сети. Далее алгоритм экспортирует полученные величины в файл формата Excel с указанием координат точек. Пользователь производит импорт данных в MatLab для визуализации (Рис.11) и анализа с целью наиболее удачных мест для установки разных типов солнечных панелей.

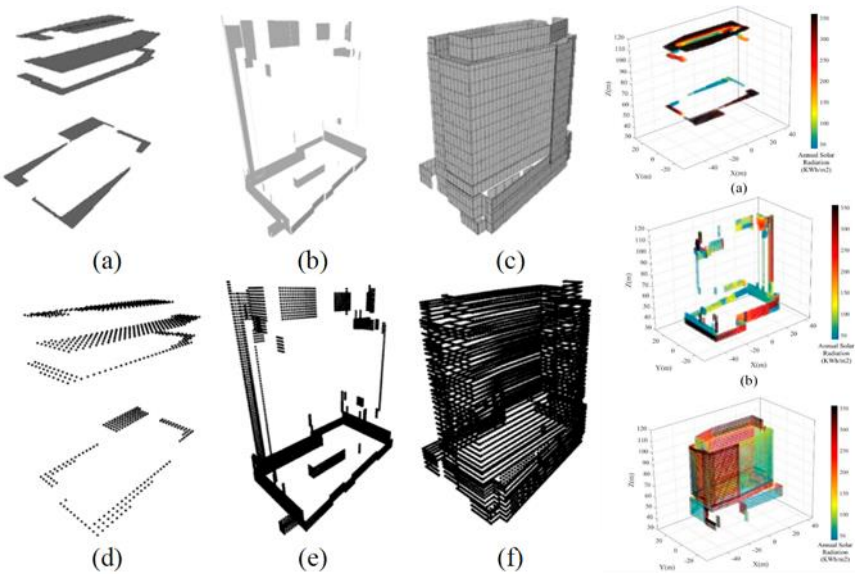

Рисунок 11. Извлечение внешних поверхностей (a) крыш, (b) несущих стен, (c) навесных стен и создание сети точек по поверхностям (d) крыш, (e) несущих стен и (f) навесных стен для последующего анализа (слева); визуализированные результаты моделирования солнечного излучения для (a) крыш, (b) несущих стен и (c) навесных стен в MATLAB (справа) [65]

Взгляд на BIM как технологию, дающую информацию о возможности деконструкции здания используется не часто, но при этом показатели эффективности строительства также зависят от того, можно ли будет демонтировать конструктивные элементы здания и использовать снова для другого объекта. Так в статье [55] предлагается методика количественной оценки возможности деконструкции стальных конструкций здания через показатель DAS (Deconstructability Assessment Scoring). По задумке пользователь в процессе моделирования размещает в соединительных узлах созданные авторами статьи особые семейства, которые несут информацию о том, насколько эффективно можно разобрать определенное соединение металлических элементов. Данная информация содержится в специально созданных параметрах этих семейств. Алгоритм Dynamo отбирает эти семейства из модели и извлекает данные параметров. Затем в результате вычислений по заложенным формулам расчета показателя DAS, алгоритм оценивает насколько эффективно можно будет в будущем провести демонтаж элементов с возможностью последующего использования, и предоставляет отчет о проведенном анализе.

3.3.3 Вывод о работе с информационной моделью / Conclusion about working with an information model

Как видно из приведенных примеров, имеется значительно количество задач по работе с моделью, которые требуется автоматизировать, а с учетом того, что большинство представленных статей было опубликовано не раньше 2018 года, следовательно, развитие подобных направлений автоматизации только начинается. При этом стоит отметить, что частота применения модуля Dynamo при автоматизации примерно равна, частоте создания плагинов путем кодирования. Это означает, что определенные задачи по взаимодействию с моделью имеют более высокий уровень сложности, чем работа с формированием и выводом баз данных, и соответственно требуют более обширного инструментария, который рациональнее реализовать при помощи кодирования.

3.4 Взаимодействие с контроллерами / Interaction with controllers

3.4.1 Автоматизация путем разработки плагина / Automating with developed plugin

Отдельным кластером можно выделить работы, направленные на организацию импорта в BIM-модель данных с реальных датчиков, установленных на фасадах или во внутренних помещениях зданий. Эта информация может быть использована для подбора оптимального угла наклона солнечных панелей, генерирующих электричество, визуализации климатических данных (температуры, влажности, интенсивности освещенности солнечным светом и т.д.) на основе модели здания, а также проведения других мероприятий по контролю эффективности распределения и применения энергии [56][57][58][59].

Так в статье [60] описывается плагин, разработанный на языке программирования C#, который соотносит угол нормали к поверхности каждой панели, замоделированной как параметрический элемент; высоту и положение Солнца относительно горизонтальной плоскости в любой момент времени. Данный алгоритм позволяет рассчитать наиболее эффективное положение панели при ее статическом закреплении с учетом движения Солнца в течение дня. Также возможен расчет наилучшего угла поворота панели для конкретного момента времени, что может быть использовано при динамическом устройстве закрепления панелей, когда отдельное устройство с помощью датчиков слежения за солнцем определяет его положение и запускает расчет для корректного поворота фотоэлектрической фасадной панели на нужный угол для наилучшего ее освещения.

Другим примером работы с фасадными системами является статья [61], где описывается ряд исследований по взаимодействию фотоэлектрических датчиков и серводвигателей через плату Arduino с моделью фасада в Revit. Авторами проводился эксперимент, с моделью имитирующей динамический фасадный элемент, к которому были присоединены датчики интенсивности освещения и электродвигатели, способные изменять положение панели. В случае, если бы панель крепилась к фасаду здания, то подобными изменениями положения она могла бы менять степень освещенности внутренних помещений здания. Исследования были направлены на то, чтобы протестировать различные способы взаимодействия панели с моделью BIM. Для этого применялись такие ПК, как Rhino с модулем автоматизации Grasshopper и Revit с интерфейсом прикладного программирования Revit API. В одном из этих экспериментов целью являлось обеспечение передачи данных с фоторезистора в модель Revit, вычисление необходимого угла наклона панели по заложенному алгоритму (Рис.12) и обратная передача информации о том, на какой угол необходимо повернуть панель контроллеру Arduino (Рис.13). Данные процессы были обеспечены путем работы созданного плагина, основанного на взаимодействии с динамической библиотекой DLL, содержащей данные, к которым могут многократно обращаться разные программы в одно и то же время. По итогу испытания по взаимодействию окончились успешно и физическая панель, также как и модель панели в Revit корректно меняла свое положение в зависимости от интенсивности света попадающего на фоторезистор.

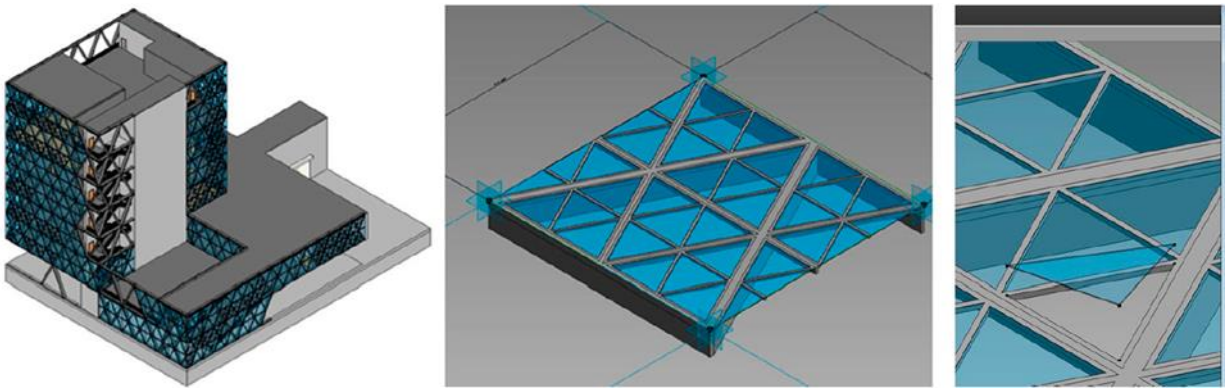

Рисунок 12. Модель здания в Revit, модель параметрического элемента фасада и демонстрация изменения положения панели в Dynamo [61]

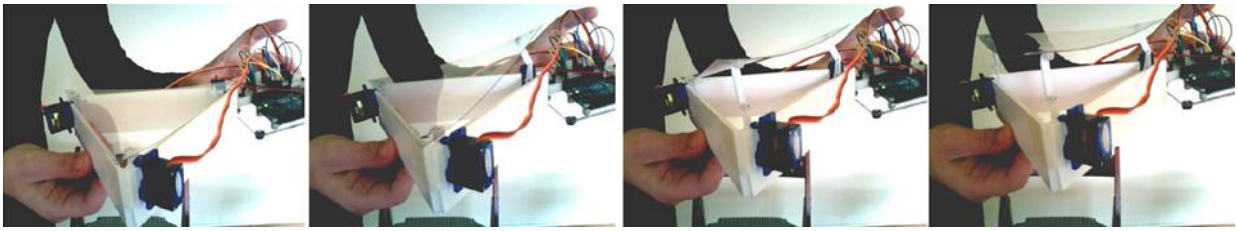

Рисунок 13. Физическая модель фасадного элемента в различных положениях [61]

3.4.2 Автоматизация путем визуального программирования в Dynamo / Automation through visual programming in Dynamo

Как упоминалось ранее, в статье [61] было описано несколько экспериментов по исследованию возможностей взаимодействия с моделью Revit для имитации контроля положения динамической фасадной панели. Кроме решения, основанного на плагине Revit API, в одном из экспериментов было продемонстрированно применение разработанного алгоритма Dynamo, позволяющего получать информацию об интенсивности солнечного света с фоторезисторов. Алгоритм постоянно обращается к файлу, созданному при работе контроллера Arduino, и извлекает данные о воздействии солнечного света на фотоэлектрический элемент цепи. Далее эти данные идентифицируются в зависимости от положения датчика на фасаде. Производится расчёт величины угла, на который необходимо повернуть панель, чтобы достичь оптимального солнечного воздействия на внутренние помещения, и затем происходит изменение положение панелей в модели фасада Revit. Как сказано в статье, в данном эксперименте не удалось создать алгоритм Dynamo, способный посылать команды обратно контроллеру Arduino для того, чтобы он менял положение физической панели. Связано это с нехваткой инструментария, предоставляемого Dynamo в 2014 году, поэтому более полный алгоритм работы с фасадной панелью был выполнен на основе плагина, что было описано ранее.

Позднее подобная задача была решена авторами статей [62][63] через Dynamo, где процесс контроля работы фасадной системы был основан на работе с BIM моделью фасада в Revit. Были разработаны новый тип фасадных панелей, имеющих более проработанный дизайн, и оптимизированную схему работы, за счет возможности обработки одним электродвигателем большей площади фасадных панелей. Алгоритм Dynamo также как и в предыдущем примере извлекал значения, снятые фоторезисторами, и вносил изменения в смоделированную фасадную систему (Рис.14), выполненную в виде кинетических фасадных блоков (Рис.15). Затем данные о степени требуемого раскрытия фасадных блоков передавались на контроллер Arduino, который выполнял изменения физической модели блоков.

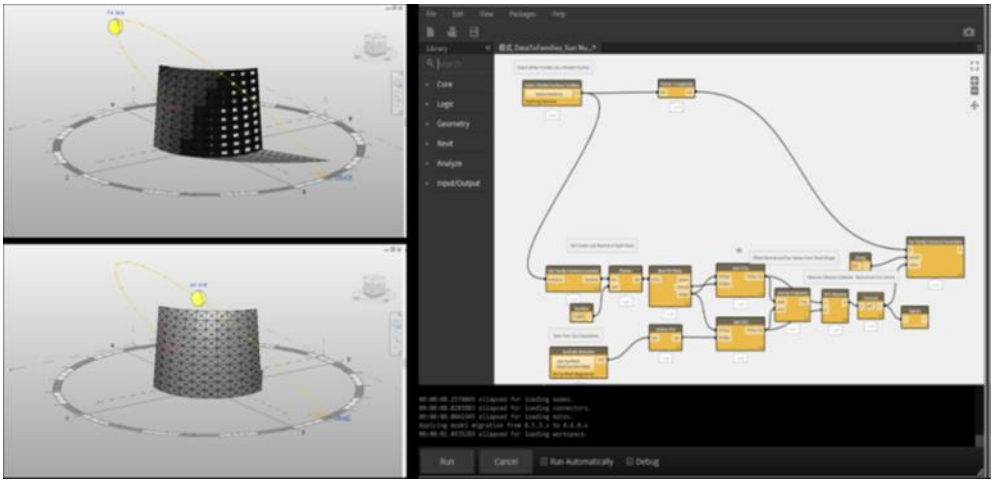

Рисунок 14. Модель кинетической фасадной системы в Revit (слева); управляющий алгоритм Dynamo для контроля раскрытия фасадных элементов (справа) [62]

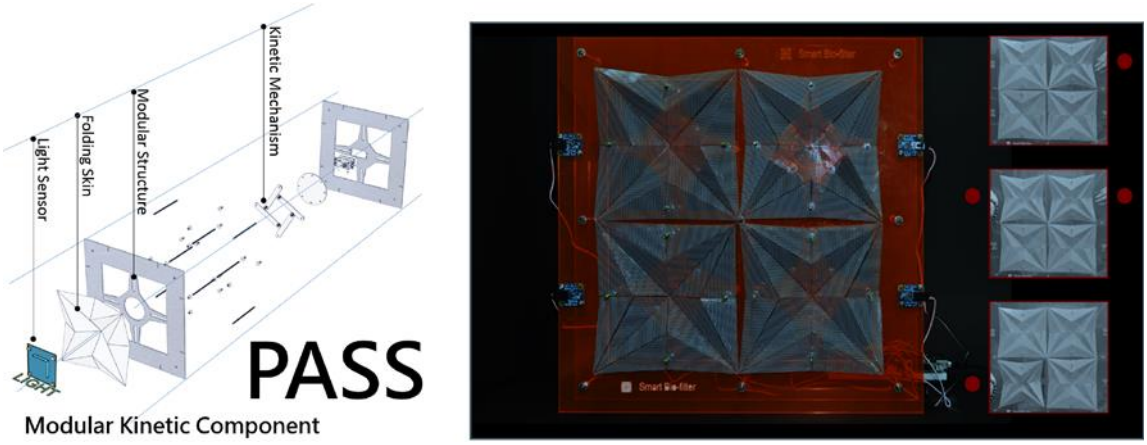

Рисунок 15. Схема кинетического фасадного блока [62]

Работа с изменением положения панелей под управлением BIM-модели также описывается в статье [64]. Авторами была создана физическая установка в виде основы с тремя вращающимися квадратными панелями, и соответствующая ей модель в Revit. В качестве эксперимента, поворачивались две крайние панели. Этот поворот фиксировался датчиками Arduino и значение угла поворота сохранялось в файле формата CSV. Далее алгоритм Dynamo считывал информацию из данного файла, идентифицировал значения углов поворота каждой панели и изменял угол поворота смоделированных панелей элементов, путем присваивания импортированного из CSV значения параметру поворота экземпляров элементов. Далее алгоритм производил расчет значения угла, на который необходимо провернуть среднюю панель. Сложность заключалась в том, что по задумке, средняя панель должна была занять промежуточное значение поворота между левой и правой панелью (Рис.16). В итоге авторы внесли в алгоритм математические зависимости, которые вычисляли необходимый угол и затем Dynamo посылал команду контролеру Arduino для изменения положения средней панели. Данное исследование не столь приближено к реальным задачам строительства, но, способно в очередной раз продемонстрировать возможности Dynamo для задач контроля и изменения положения реальных тел, что может быть использовано в других проектах, связанных, например, с фасадными системами.

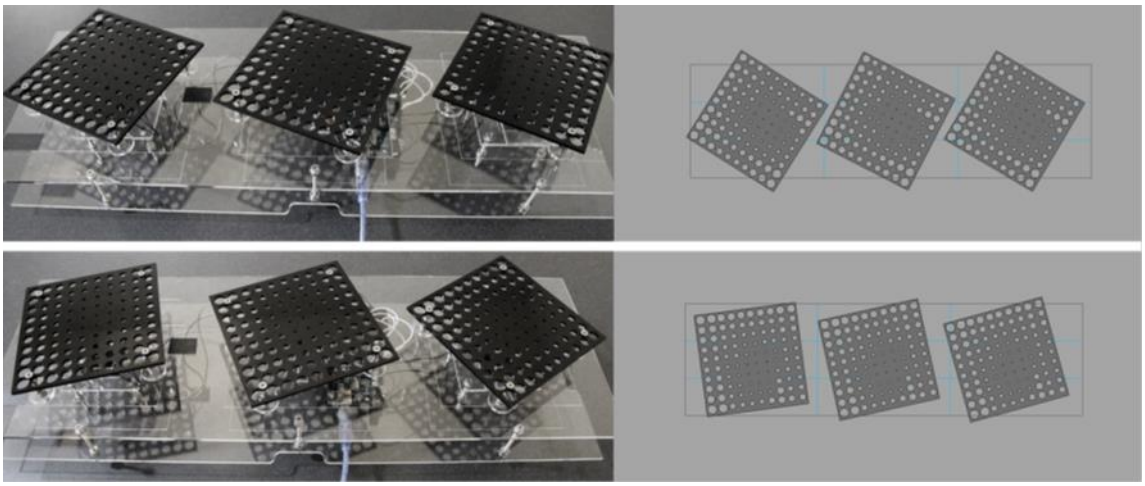

Рисунок 16. Результат работы алгоритма Dynamo по повороту панели в рассчитанное положение [64]

В статье [66] представлен алгоритм по визуализации в модели Revit данных о влажности, полученных ранее с датчиков на реальном фасаде здания. Информация загружается из таблиц и связывается по номерам датчиков, которые в модели представлены специальными параметрическими элементами на фасадах. Затем с помощью стандартных узлов Dynamo представляется пространственное распределение влаги относительно геометрии фасадов по результатам съемки датчиков. Данная методика визуальной оценки показателей может быть использована для более наглядного представления результатов, поиска и анализа наиболее

уязвимых участков фасада здания, а также планирования технического обслуживания конструкций.

Еще одним примером взаимосвязи датчиков в пространстве и информационной модели здания является алгоритм Dynamo описанный в статье [67]. Здесь целью ставилось исследование и визуализация значений PMV (Predicted Mean Vote - индекс прогнозируемого среднего теплового комфорта человека) по датчикам температуры и влажности в пространстве помещения. По задумке датчики системы Arduino размещают в реальном помещении на определенной высоте от уровня пола и они собирают информацию об окружающей среде. После сбора данные подгружаются в модель Revit при помощи алгоритма Dynamo, использующего для корректного взаимодействия с Arduino загружаемый модуль Firefly. Производится идентификация данных от разных датчиков и привязка их значений к координатам точек в модели помещения. Во внутреннем пространстве помещения создается сеть дополнительных точек с шагом, задаваемым пользователем. В базе данных Dynamo для каждой точки сети методом интерполяции вычисляются значения температуры и влажности. Далее алгоритм обрабатывает данные по каждой точке: рассчитывает значение индекса PMV и, в зависимости от его величины, подбирает цвет для каждой точки с учетом выбранных характеристик палитры. В конечном итоге программа выводит трехмерный график индекса PMV во внутреннем пространстве модели помещения, позволяя пользователю оценить распределение более и менее комфортных зон для пребывания человека в помещении (Рис.17). Стоит отметить, что сама процедура расчета индекса PMV для каждой точки была выполнена путем кодирования узле «Python Script». Это связано с тем, что формула расчета индекса является достаточно сложной для создания в виде последовательности сценариев, так как предусматривает порядка 11 независимых показателей, среди которых основополагающими являются температура и влажность, определяемые в данной работе.

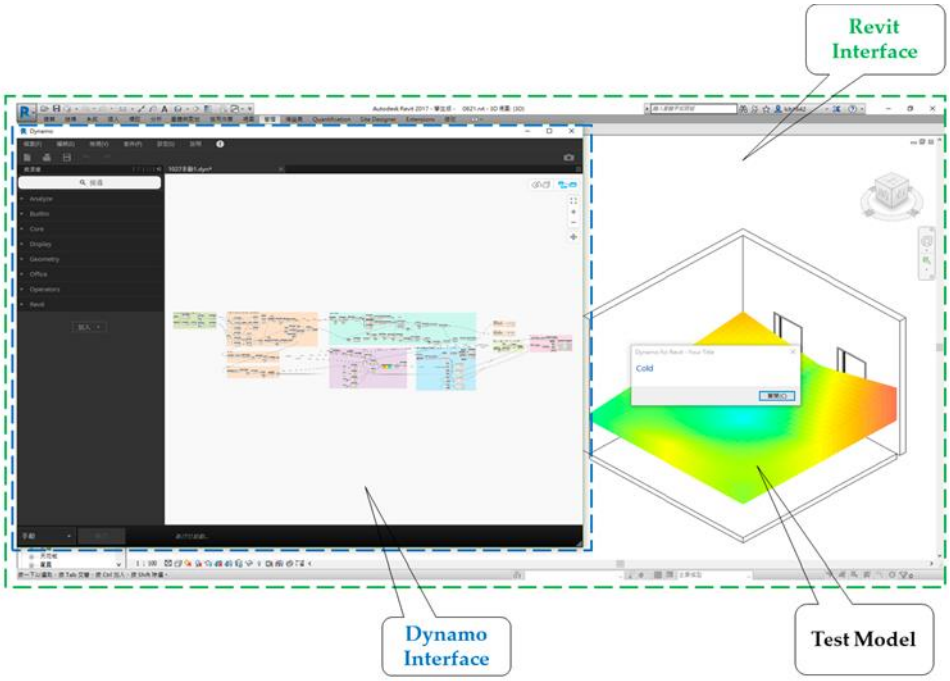

Рисунок 17. Результаты анализа индекс прогнозируемого среднего теплового комфорта человека в модели BIM [67]

3.4.3 Вывод о задачах по взаимодействию с контроллерами / Conclusion on the tasks of interaction with controllers

Все статьи, связанные с организацией взаимодействия BIM-программ с датчиками и контролерами методом создания плагинов на API имеют более ранние годы публикации (2011-2014гг.), чем статьи, описывающие работу алгоритмов на Dynamo по решению этих задач (2014-2018гг.). Это связано с развитием модулей взаимодействия Dynamo и системы Arduino, которые сумели предоставить достаточный пакет инструментов (модуль Firefly) пользователям для взаимодействия этих систем. При этом постепенно данная тема находит свое применение в реальных задачах, так как способна предоставить BIM-модели данные, считываемые в реальном

времени, которые можно использовать для управления системами, анализа энергетических показателей и создания баз данных для автоматизированного мониторинга состояния здания в течение длительного времени.

4 Заключение / Conclusions

Исходя из вышесказанного, можно сделать вывод, что оба метода автоматизации не могут заменить друг друга, так как предназначены для пользователей с разным уровнем навыков программирования, а также способны решать задачи различного уровня сложности. Помимо автоматизации, они отлично справляются с обработкой информации находящейся как внутри модели, так и извне, что может значительно расширить возможности ПК Revit.

Прогрессивным методом на данный момент является применение модуля Dynamo с использованием узлов «Python Script». Это связано с тем, что данный путь предоставляет все возможности визуального программирования той части алгоритма, которая не требует написания кода, а также дает возможность сразу проверять работу созданной программы по взаимодействию с моделью, но при этом обеспечивает всеми инструментами по работе с информацией, подразумевающимися при написании кода на языке Python.

References

1. Becerik-Gerber B., Kensek K. Building information modeling in architecture, engineering, and construction: Emerging research directions and trends. J. Prof. Issues Eng. Educ. Pract. 2010. 136, 3. Pp. 139–147.
2. Kensek K.M. Advancing BIM in academia: Explorations in curricular integration. Computational Design Methods and Technologies: Applications in CAD, CAM and CAE Education. 2012. Pp.101– 121.
3. Smith D.K., Tardiff M. Building Information Modeling: A Strategic Implementation Guide for Architects, Engineers, Constructors, and Real Estate Asset Managers. Building Information Modeling: A Strategic Implementation Guide for Architects, Engineers, Constructors, and Real Estate Asset Managers. 2009. 186 p.
4. Stacks R. et al. BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers, 3rd Edition. 3rd ed. Wiley, 2018. 688 p.
5. Ford S. et al. An information engineering approach to modelling building design. Autom. Constr. Elsevier, 1995. 4(1). Pp. 5–15.
6. Kensek K.M. Building information modeling. Building Information Modeling. 2014. 1285 p.
7. Kensek K. BIM guidelines inform facilities management databases: A Case Study over Time. Buildings. 2015. 5(3). Pp. 899–916.
8. Migilinskas D. et al. The benefits, obstacles and problems of practical bim implementation. Procedia Engineering. 2013. 57. Pp. 767–774.
9. Celani G., Eduardo Verzola Vaz C. CAD scripting and visual programming languages for implementing computational design concepts: A comparison from a pedagogical point of view. Int. J. Archit. Comput. 2012. 10(1) Pp. 121–137.
10. Dieckmann A. Utilising Dynamo “Beyond” Computational Design. Revit Technology Conference Europe 2014. Dublin, Ireland, 2014.
11. Zaragoza-Grife J.N., Solis-Carcaño R.G., Corona G.A. Hybrid Model for Developing BIM Software Extensions. J. Adv. Manag. Sci. EJournal Publishing, 2015. Pp. 227–232.
12. Ignatova E., Zotkin S., Zotkina I. The extraction and processing of BIM data. IOP Conference Series: Materials Science and Engineering. Institute of Physics Publishing, 2018. 365, 6.
13. Yarmohammadi S., Castro-Lacouture D. Automated performance measurement for 3D building modeling decisions. Autom. Constr. Elsevier B.V., 2018. 93. Pp. 91–111.
14. Ko A.J., Myers B.A., Aung H.H. Six learning barriers in end-user programming systems. Proceedings - 2004 IEEE Symposium on Visual Languages and Human Centric Computing. 2004. Pp. 199–206.
15. Khaja M., Seo J.D., McArthur J.J. Optimizing BIM Metadata Manipulation Using Parametric Tools. Procedia Engineering. Elsevier Ltd, 2016. 145. Pp. 259–266.
16. Wülfing A., Windisch R., Scherer R.J. A visual BIM query language. eWork and eBusiness in Architecture, Engineering and Construction - Proceedings of the 10th European Conference on Product and Process Modelling, ECPPM 2014. 2015. Pp. 157–164.
17. Kensek K. Visual programming for building information modeling: Energy and shading analysis case studies. J. Green Build. 2015. 10(4). Pp. 28–43.
18. Pertceva A., Khizhnyak N., Radaev A. Algorithm of designing complex shape construction using automation tools (by example of Autodesk Revit, Autodesk AutoCAD and Dynamo). Russ. J. Transp. Eng. Publishing Company World of Science LLC, 2018. 5(4).

19. Kilkelly M. What is Dynamo [Electronic resource]. Archsmater. 2018. URL: https://archsmarter.com/what-is-dynamo-revit/.
20. Moore J., Benjamin J. Everyday Dynamo: Automating Simple Solutions That Bridge Workflow Gaps Within Revit. Autodesk University. Autodesk, 2016. Pp. 1–74.
21. Kensek K.M. Teaching visual scripting in bim: A case study using a panel controlled by solar angles. J. Green Build. 2018. 13(1). Pp. 115–137.
22. Ignatova E., Zotkin S., Zotkina I. The extraction and processing of BIM data. IOP Conference Series: Materials Science and Engineering. Institute of Physics Publishing, 2018. 365, 6.
23. van der Zee A., De Vries B. Design by computation. Proc. 11th Int. Conf. Gener. Art 2008. 2008. Pp. 35–52.
24. Laakso M., Kiviniemi A. The IFC standard - A review of history, development, and standardization. Electron. J. Inf. Technol. Con tr. 2012. 17. Pp. 134–161.
25. Froese T. Future directions for IFC-based interoperability. Electron. J. Inf. Technol. Constr. 2003. 8. Pp. 231–246.
26. Wenighofer R. et al. BIM use case – Payment of tunnel excavation classes – Example Zentrum am Berg | BIM-Anwendungsfall (AwF) Abrechnung-Vortrieb am Beispiel des Zentrums am Berg. Geomech. und Tunnelbau. 2020. 13(2). Pp. 237–248.
27. Pazlar T., Turk Z. Interoperability in practice: Geometric data exchange using the IFC standard. Electron. J. Inf. Technol. Constr. 2008. 13. Pp. 362–380.
28. Steel J., Drogemuller R., Toth B. Model interoperability in building information modelling. Softw. Syst. Model. 2012. 11, 1. Pp. 99–109.
29. Zada A.J., Tizani W., Oti A.H. Building information modelling (BIM) - Versioning for collaborative design. Computing in Civil and Building Engineering - Proceedings of the 2014 International Conference on Computing in Civil and Building Engineering. 2014. Pp. 512–519.
30. Zotkin S.P., Ignatova E. V., Zotkina I.A. The Organization of Autodesk Revit Software Interaction with Applications for Structural Analysis. Procedia Engineering. Elsevier Ltd, 2016. 153. Pp. 915–919.
31. Pärn E.A., Edwards D.J. Conceptualising the FinDD API plug-in: A study of BIM-FM integration. Autom. Constr. Elsevier B.V., 2017. 80. Pp. 11–21.
32. Li Z. et al. Research on the Design of Scaffold Based on Application of Secondary Development in Revit. Hunan Daxue Xuebao/Journal Hunan Univ. Nat. Sci. Hunan University, 2018. 45(9). Pp. 65–73.
33. Yan W. et al. INTERFACING BIM WITH BUILDING THERMAL AND DAYLIGHTING MODELING Sandeep Kota, Jose Luis Bermudez Alcocer, and Manish Dixit. 13th Conf. Int. Build. Perform. Simul. Assoc. 2013. Porceedings of BS2013. Pp. 3521; 3528.
34. Piaskowski, AK, Petersons, R, Wyke, SCS, Petrova, EA & Svidt K. Automation of data transfer between a BIM model and an environmental quality assessment application. Cib Proc. 2019.
35. Lim Y.-W. et al. Computational BIM for Building Envelope Sustainability Optimization. MATEC Web Conf. EDP Sciences, 2019. 278. Pp. 04001.
36. Bueno C., Pereira L.M., Fabricio M.M. Life cycle assessment and environmental-based choices at the early design stages: an application using building information modelling. Archit. Eng. Des. Manag. 2018. 14(5). Pp. 332–346.
37. Санджиев Н.В. et al. Dynamo platform for automation Revit/ Sandzhiev N. Платформа Dynamo для автоматизации Revit©. 2018. 6. 75–82 p.
38. Sharif S., Gentry R. BIM for Masonry: Development of BIM Plugins for the Masonry Unit Database. Ecaade 2015 Real Time - Extending Reach Comput. Vol 1. 2015. 1. Pp. 567–576.
39. Sadeghi M. et al. Developing building information models (BIM) for building handover, operation and maintenance. J. Facil. Manag. 2019. 17(3). Pp. 301–316.
40. Wang Z., Rezazadeh Azar E. BIM-based draft schedule generation in reinforced concrete-framed buildings. Constr. Innov. 2019. 19(2). Pp. 280–294.
41. Romanovskyi R., Sanabria Mejia L., Rezazadeh Azar E. BIM-based decision support system for concrete formwork design. Proceedings of the 36th International Symposium on Automation and Robotics in Construction, ISARC 2019. 2019. Pp. 1129–1135.
42. Chen Q. et al. Research on the modeling of Yingxian wooden tower based on revit and revit API. Xi'an Jianzhu Keji Daxue Xuebao/Journal Xi'an Univ. Archit. Technol. Science Press, 2017. 49(3).
43. Bai Q. et al. Application of BIM in the creation of prefabricated structures local parameterized component database. Archit. Eng. 2019. 4(2). Pp. 13–21.
44. Yang M. et al. The module of rebar modeling for Chinese building standard detailing drawings by BIM-based methods. ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things. International Association for Automation and Robotics in Construction I.A.A.R.C), 2018.

45. Kensek K., Henkhaus A. Solar access zoning + building information modeling. 42nd ASES National Solar Conference 2013, SOLAR 2013, Including 42nd ASES Annual Conference and 38th National Passive Solar Conference. 2013. Pp. 250–257.
46. Zhang S. et al. Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules. Autom. Constr. 2013. 29. Pp. 183–195.
47. Pereira S.M.S.A., Amorim S.R.L. O desenvolvimento de ferramenta de verificação de requisitos de projeto para o Revit® através de API. XV Encontro Nacional de Tecnologia do Ambiente Construído. Marketing Aumentado, 2014. 1. Pp. 2954–2963.
48. Fan T.W. Applying Fire Simulation to BIM Modeling with API Programming for Evacuation Time Calculation. Lecture Notes in Civil Engineering. 2020. 59. Pp. 175–184.
49. Kensek K., Ding Y., Longcore T. Green building and biodiversity: Facilitating bird friendly design with building information models. J. Green Build. 2016. 11(2). Pp. 116–130.
50. Yang X., Koehl M., Grussenmeyer P. Mesh-to-bim: From segmented mesh elements to bim model with limited parameters. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. International Society for Photogrammetry and Remote Sensing, 2018. 42(2). Pp. 1213–1218.
51. Yang X. et al. HBIM modeling from the surface mesh and its extended capability of knowledge representation. ISPRS Int. J. Geo-Information. MDPI AG, 2019. 8, 7.
52. Tang F. et al. Integrating three-dimensional road design and pavement structure analysis based on BIM. Autom. Constr. Elsevier B.V., 2020. 113.
53. Jordan A. et al. Automated integration: A new frontier in BIM. 20th Congress of IABSE, New York City 2019: The Evolving Metropolis - Report. 2019. Pp. 1859–1866.
54. Seghier T.E., Ahmad M.H., Lim Y.-W. Automation of concrete usage index (CUI) assessment using computational BIM. Int. J. Built Environ. Sustain. Penerbit UTM Press, 2019. 6(1). Pp. 23–30.
55. Basta A., Serror M.H., Marzouk M. A BIM-based framework for quantitative assessment of steel structure deconstructability. Autom. Constr. 2020. 111.
56. Jin R. et al. Integrating BIM with building performance analysis in project life-cycle. Automation in Construction. Elsevier B.V., 2019. 106 p.
57. Sehrawat P., Kensek K. Urban energy modeling: GIS as an alternative to BIM. 2014 ASHRAE/IBPSA-USA Building Simulation Conference. 2014. Pp. 235–242.
58. Singh S., Kensek K. Early design analysis using optimization techniques in design/practice. 2014 ASHRAE/IBPSA-USA Building Simulation Conference. 2014. Pp. 284–291.
59. Zardo P., Ribeiro L.A., Mussi A.Q. Bim and parametric design applications for buildings' energy efficiency: An analysis of practical applications. Arquiteturarevista. Universidade do Vale do Rio dos Sinos, 2019. 15(2). Pp. 238–255.
60. Xuan X. Application of building information modeling in building integrated photovoltaics. Advanced Materials Research. 2011. 171–172. Pp. 399–402.
61. Kensek K.M. Integration of Environmental Sensors with BIM: Case studies using Arduino, Dynamo, and the Revit API. Inf. la Constr. 2014. 66, 536.
62. Shen Y.T.Y.T., Lu P.W.P.W. The development of kinetic façade units with BIM-based active control system for the adaptive building energy performance service. CAADRIA 2016, 21st International Conference on Computer-Aided Architectural Design Research in Asia - Living Systems and Micro-Utopias: Towards Continuous Designing. The Association for Computer- Aided Architectural Design Research in Asia (CAADRIA), 2016. Pp. 517–526.
63. Shen Y.T., Wu T.Y. Sync-BIM: The interactive BIM-based platform for controlling data-driven kinetic façade. Communications in Computer and Information Science. Springer Verlag, 2016. 618. Pp. 445–450.
64. Al-Qattan E., Yan W., Galanter P. ESTABLISHING PARAMETRIC RELATIONSHIPS FOR DESIGN OBJECTS THROUGH TANGIBLE INTERACTION. P. Janssen, P. Loh, A. Raonic, M. Schnabel (eds.), Protocols, Flows, and Glitches - Proceedings of the 22nd CAADRIA Conference, Xi'an Jiaotong-Liverpool University, Suzhou, China, 5-8 April 2017, Pp. 147-156. 2017.
65. Salimzadeh N., Vahdatikhaki F., Hammad A. BIM-based surface-specific solar simulation of buildings. ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things. 2018.
66. Pocobelli D.P. et al. Building information models for monitoring and simulation data in heritage buildings. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. International Society for Photogrammetry and Remote Sensing, 2018. 42(2). Pp. 909–916.
67. Chang K.M., Dzeng R.J., Wu Y.J. An automated IoT visualization BIM platform for decision support in facilities management. Appl. Sci. MDPI AG, 2018. 8(7).