Вестник Дагестанского государственного технического университета. Технические науки. Том 49, №4, 2022 Herald of Daghestan State Technical University. Technical Sciences. Vol.49, No.4, 2022 http://vestnik.dgtu.ru/ ISSN (Print) 2073-6185 ISSN (On-line) 2542-095X

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS УДК 004.4'236 DOI: 10.21822/2073-6185-2022-49-4-126-133 Оригинальная статья /Original Paper

Алгоритм проектирования несущих конструкций многоэтажного здания с использованием среды визуального программирования DYNAMO STUDIO Т.А. Юрошева, А.В. Калиниченко, В.Г. Макиев

Северо – Кавказский горно-металлургический институт (Государственный технологический университет), 362021, г. Владикавказ, ул. Николаева, 44, Россия

Резюме. Цель. Статья посвящена актуальной проблеме минимизации времени работы проектировщика с применением Dynamo Studio в качестве мощного инструмента автоматизации задач проектирования. Метод. Приведено описание некоторых возможностей среды визуального программирования Dynamo Studio при разработке проекта здания в программном комплексе Revit. Результат. Представлены и описаны методики работы сценариев Dynamo, позволяющие существенно ускорить процессы моделирования и подготовки проектной документации. Принципы информационного моделирования позволяют увеличить скорость выполняемой проектной работы, не сказываясь пагубно на ее качестве. Это позволяет избежать широкого ряда ошибок при проектировании, что в дальнейшем не приведет к соответствующим проблемам на этапе строительства объекта. Представлены результаты применения разработанных алгоритмов для автоматизированного построения трехмерной модели многоэтажного здания. Вывод. Использование среды визуального программирования позволяет создавать информационную модель, которая обновляет свою конфигурацию при изменении входящих в нее элементов.

Ключевые слова: информационное моделирование, автоматизация, визуальное программирование, программный код, оператор

Для цитирования: Т.А. Юрошева, А.В. Калиниченко, В.Г. Макиев. Алгоритм проектирования несущих конструкций многоэтажного здания с использованием среды визуального программирования DYNAMO STUDIO. Вестник Дагестанского государственного технического университета. Технические науки. 2022; 49(4):126-133. DOI:10.21822/2073-6185-2022-49-4-126-133

Algorithm for designing load-bearing structures of a multi-storey building using the visual programming environment DYNAMO STUDIO

T.A. Yurosheva, A.V. Kalinichenko, V.G. Makiev

North Caucasus Mining and Metallurgical Institute (State Technological University) 44 Nikolaeva Str., Vladikavkaz 362021, Russia

Abstract. Objective. The article is devoted to the actual problem of minimizing the designer's work time using Dynamo Studio as a powerful tool for automating design tasks. **Method**. A description of some of the possibilities of the Dynamo Studio visual programming environment when developing a building project in the Revit software package is given. **Result.** Dynamo scripts work methods are presented and described, which allow to significantly speed up the processes of modeling and preparation of project documentation. The principles of information modeling make it possible to increase the speed of the project work being carried out without adversely affecting its quality. This makes it possible to avoid a wide range of design errors, which in the future will not lead to corresponding problems during the construction phase of the facility. The results of the application of the developed algorithms for the automated construction of a three-dimensional model of a multi-storey building are presented. **Conclusion.** Using the visual programming environment allows you to create an information model that updates its configuration when the elements included in it change.

Keywords: information modeling, automation, visual programming, program code, operator For citation: T.A. Yurosheva, A.V. Kalinichenko, V.G. Makiev. Algorithm for designing load-bearing structures of a multi-storey building using the visual programming environment DYNAMO STUDIO. Herald of the Daghestan State Technical University. Technical Science. 2022; 49(4):126-133. DOI:10.21822/2073-6185-2022-49-4-126-133

Введение. Процесс проектирования можно представить, как пошаговый набор действий, выполняемых по определенному алгоритму. В этом же ключе работает система DYNAMO. Она представляет алгоритмы в наглядном формате, доступном для понимания пользователями без специальных знаний в области программирования.

Визуальное программирование – современная альтернатива текстовым форматам программирования (с использованием языков программирования). Вместо ввода текста со сложными и строгими правилами, достаточно соединять предварительно разработанные готовые узлы вместе, последовательно добиваясь нужной информации. Каждый узел уже содержит в себе куски кода, которые пришлось бы создавать в процессе классического программирования – получается некий конструктор алгоритма.

Dynamo - среда визуального программирования с открытым программным кодом. Главным назначением среды является создание новых программных функций, позволяющих анализировать большие массивы данных и оптимизировать рутинные процессы. Процесс визуального программирования среды Dynamo можно представить, как последовательный набор действий, выполняемых по заданному алгоритму. Данный алгоритм выполняется с помощью предварительно разработанных текстовых фрагментов программного кода – узлов (англ. node), последовательность которых составляет сценарий (англ. script) [1]. В данной работе описаны созданные алгоритмы Dynamo, позволяющие автоматизировать моделирование несущих конструкций многоэтажного здания в программном комплексе Revit.

Постановка задачи. На данный момент проектирование вышло на такой уровень, при котором обеспечивается возможность не только построения трехмерной модели и заданий элементам всех необходимых свойств, но также и автоматизированного создания отдельных элементов сооружения. Речь идет о том, что современные программы строительного проектирования предусматривают возможность создания дополнений или макросов с использованием стандартных языков программирования, владение которыми, тем не менее, требует наличия у проектировщика дополнительных компетенций.

Данное обстоятельство определило целесообразность проведения исследования, целью которого является разработка алгоритма создания различных элементов строительных конструкций, либо монотонно повторяющихся операций с использованием средств автоматизации в составе программных сред строительного проектирования.

Для достижения указанной цели в рамках исследования были поставлены и решены следующие задачи:

- 1. Оценить возможность применения среды визуального программирования при разработке информационной модели проектируемой строительной конструкции замкнутого типа.
- 2. Создание обобщенного алгоритма проектирования элементов строительной конструкции замкнутого типа; осуществление монотонных процессов, с использованием средств автоматизации в составе программных сред строительного проектирования.
- 3. Реализация разработанного алгоритма на практическом примере.

Методы исследования. Прежде всего, следует отметить, что решение современных задач в области автоматизированного строительного проектирования предполагает последовательную реализацию двух видов процессов [2]:

Алгоритмизация – заключается в разработке алгоритма как набора инструкций, последовательное выполнение которых на основе данных, задаваемых пользователем (имплементация), в соответствующей программной среде обеспечит построение модели строительного объекта или его отдельных элементов в информационном аспекте – в виде массивов данных и различных взаимосвязей между ними. В наиболее общем случае обеспечивается визуальными редакторами программирования, зачастую являющимися надстройками для программных сред трехмерного моделирования.

Визуализация – предполагает графическую интерпретацию (как правило, в трехмерном пространстве) информационной модели строительного объекта или его отдельных элементов как результата выполнения алгоритма. В наиболее общем случае обеспечивается современными программными средами трехмерного моделирования.

Алгоритм автоматизированного создания уровней проектируемой модели. Результатом работы алгоритма, схема которого показана на рис. 1. являются уровни модели, построенные по предопределенным параметрам этажности. Они также добавлены в план этажей, для дальнейшего удобного пользования.



Рис.1. Алгоритм автоматизированного создания уровней Fig.1. Algorithm for automated creation of levels

Данное решение позволяет свести к минимуму трудозатраты по созданию и правильному переименованию уровней:

- 1. Пользовательский интерфейс создается посредством подключения пакета Data-Shapes и использования его нодов для расширения функций скриптов Dynamo.
- 2. Входные параметры включают в себя: количество этажей в проектируемом здании; высота этажей.
- 3. Перебираем пользовательское количество этажей для дальнейшей вставки в проектируемую модель.
- 4. Добавление этажа осуществляется функцией «Level.ByElevationAndNane». Она позволяет создавать уровень, учитывая высоту и имя.
- 5. Добавление уровня земли, крыши и подвала осуществляет та же функция.
- 6. Перебираем созданные пользователем уровни.
- 7. Размещаем созданные уровни в план этажа проекта. Осуществляется это посредством использования узла «Views.FloorPlanView.ByLevel».
- 8. Перебираем все уровни, существующие в проекте.
- 9. Проверяем, принадлежит ли уровень автоматически созданному программным обеспе-

чением в проекте. Если да, то этот уровень удаляется. В противном случае берётся следующий уровень.

10. Удаляется уровень функцией Delete().

Алгоритм копирования типового этажа на выбранные уровни. В первую очередь для сокращения времени проектирования был применен сценарий по копированию имеющихся в модели элементов на остальные уровни жилых этажей с указанием, на какие именно этажи должно быть произведено копирование. Алгоритм этого сценария продемонстрирован на рис. 2.



Рис.2. Алгоритм копирования элементов Fig.2. Algorithm for copying elements

Главными преимуществами данного подхода по сравнению с применением собственных инструментов системы автоматизированного проектирования являются скорость выполнения процесса и его точность:

- 1. Пользовательский интерфейс создаётся на основе пакета нодов Data-Shapes, позволяющий выбрать этаж для копирования, и уровни, на которые происходит копирование.
- 2. Исходные данные определяются особенностями объекта проектирования и решаемой задачи. Они включают в себя: этаж, конструкции которого используются для копирования; уровни, предназначенные для копирования.
- 3. Перебираем уровни, выбранные для копирования.
- 4. Задание стен на выбранном уровне осуществляется путем получения всех элементов, расположенных в модели на указанной отметке («All Element at level»). И дальнейшей фильтрации («Element.IsOfCategory», «List.FilterByBoolMask»), путем поиска соответствий с выбранной категорией («Categories» – стены).
- 5. Загружаем пользовательский пакет нодов «Clockwork»; применем узел «Element.CopyToLevel». Он осуществляет копирование выбранных элементов на нужные уровни.
- 6. Задание перекрытия осуществляется аналогично стенам.
- 7. Процесс копирования перекрытий также производится аналогично стенам.
- 8. Задание колонн осуществляется аналогично стенам.
- 9. Процесс копирования колонн также производится аналогично стенам.

Скрипт автоматического создания уровней проектируемой модели. Сценарий создания уровней, изображенный на рис. 3. представляет собой последовательность узлов, выполняющих соответствующие функции в указанном порядке:

1. Создание пользовательского интерфейса ввода данных о параметрах проектируемых уровней.

2. Написание встроенного сценария Python создания уровней и добавления их в план этажей, как показано на рис. 3.



Рис.3. Скрипт создания уровней Fig.3. Script for creating levels

Так как в базовый функционал Dynamo не входят инструменты, создающие окно представлений, воспользуемся возможностью загрузить пакет нодов Data-Shapes. Основной целью, которого является увеличение спектра функциональных возможностей проигрывателя Dynamo [3]. Значения количества и высоты этажей задается с помощью узла "Slider", который создаёт ползунки ввода в форме. Нод "List Create" позволяет вводить входные данные в виде списка. Формированием интерфейса занимается узел "MultipleinputForm++". Он способен интерпретировать всю входную информацию, создавать форму пользовательского интерфейса, а затем возвращать полученные данные.

Также мы можем задать описание формы ("Description_optional"), текст кнопки ("ButtenText_optional") и переключатель ("toggle"). Для переключения UI.MultipleInputForm++ должно быть установлено значение "true" [2]. Этот ввод имеет решающее значение, поскольку он позволяет лучше контролировать порядок, в котором будет функционировать скрипт, показанный на рис. 4.



Рис.4. Сценарий формирования интерфейса Fig.4. Interface formation scenario

Результат выполнения сценария формирования интерфейса представлен на рис. 5.



Рис.5. Результат выполнения сценария формирования интерфейса Fig.5. The result of the execution of the interface formation script

Пользовательские входные данные обрабатываются в «Python Script», который позволяет расширить возможности Dynamo и заменить большое количество узлов компактными строками кода. Программный модуль, показанный на рис. 6, осуществляет построение уровней по заданным значениям и добавление их в план этажей.



Рис.6. Скрипт, создающий пользовательские уровни и добавляет их в план этажей Fig.6. Script that creates custom levels and adds them to the floor plan

Для создания пользовательских сценариев и обеспечения доступа к API Revit подключим следующие библиотеки:

- 1. "RevitNodes" позволяет в Python Script использовать стандартные узлы Dynamo.
- 2. "RevitServices" предназначен для связи Dynamo и Revit, импортируя DocumentManger (выдаёт проект, с которым мы работаем) и Transacrion.Manager (даёт возможность внести изменения в проект).
- 3. "RevitAPI" содержит множество методов Revit.

В качестве входных данных Python Script возьмём значения количества этажей (kolvo) и их высоту (vysota). Создадим список elev[], куда будем добавлять генерируемые уровни. Затем начнем перебор уровней. Их создание осуществляется командой "Level.ByElevayionAndName". Функцией elev.append() добавим этаж в список elev. Также необходимо добавить сгенерированные уровни в план этажей ("Views.FloorPlanView.ByLevel()"). Так как Revit по умолчанию создаёт 2 уровня, их необходимо удалить ("doc.Delete()"). Для копирования конструкций был использован скрипт, продемонстрированный на рис. 7. Он выполняется в следующем порядке:

- 1. Создание пользовательского интерфейса ввода данных о параметрах копирования.
- 2. Копирование и перенос стен на выбранные уровни.
- 3. Копирование и перенос перекрытий на выбранные уровни.
- 4. Копирование и перенос колонн на выбранные уровни.





Пользовательский интерфейс, скрипт которого представлен на рис. 8., создаётся нодами из пакета Data-Shapes. Этаж, элементы которого копируются, задается из выпадающего окна узлом "DropDown Data". "Keys" – элементы, которые будут показаны пользователю. Они получены путем взятия имен ("Level.Name") из всех существующих в проекте уровней ("Categoties(Уровни)"-"All Elements of Category"). "Values" – это элементы, которые будут возвращены в соответствии с ключами ("Categoties(Уровни)"-"All Elements of Category"). Biofopom этажей для копирования занимается узел "SelectModelElements Data" [3]. Он предусматривает применение фильтра категории к выбору. Это означает, что пользователь сможет выбирать только указанные категории. Это уменьшает ошибки выбора и гарантирует, что рабочий процесс работает так, как задумано. Поскольку копирование осуществляется на этажи, потому в качестве значения "CategoryFilter" возьмем "Categories (Уровни)". Входные данные должны вводиться в виде списка, поэтому создадим его нодом "List Create" и подадим в "MultipleinputForm++". Также мы создадим описание формы ("Description_optional") и переключатель("toggle"). Результат выполнения скрипта формирования интерфейса представлен на рис. 9.



Рис.8. Скрипт формирования интерфейса Fig.8. Interface formation script

🛃 Data-Shapes Multi Inp	ut UI ++ X
Копирование этажей	
Этажи для копирования	Select Уровни
Этаж копирования	~ ·
	Set Values

Рис.9. Результат выполнения скрипта формирования интерфейса Fig.9. The result of the execution of the interface formation script

Рис. 10. описывает процесс копирования элементов. Спецификой программы Dynamo является то, что узлов по прямому дублированию элементов нет, поэтому импортируем пакет нодов "Clockwork" [4]. Создадим список стен, расположенных на выбранном уровне. Для этого проверим, входят ли элементы на этаже в состав пользовательской категории ("Element.IsOfCategory"). Отфильтруем список и подадим элементы, принимающие значение – "true", на вход узла "Element.CopyToLevel". Данный нод копирует элементы и вставляет их на выбранные уровни.



Рис.10. Скрипт, осуществляющий копирование Fig.10. Script that performs copying

Главными преимуществами данного подхода по сравнению с применением собственных инструментов Revit являются скорость выполнения процесса и его точность. Построение элементов данным образом исключает человеческий фактор и ошибки соосности, привязки к уровням и осям, неточности геометрических размеров конструкций. Результат выполнения скрипта можно увидеть на рис. 11.



Рис.11. Результат выполнения скрипта копирования Fig.11. The result of the execution of the copy script

Представленные выше алгоритмы являются лишь частными случаями возможностей, открывающихся перед пользователем Dynamo. Воплощение данных возможностей позволяет раскрыть весь современный потенциал информационного моделирования. Стартовый экран программы содержит необходимые ссылки на форум для обсуждения и веб-сайт продукта [5], при переходе по которым можно найти множество полезных советов по созданию сценариев, ознакомиться с важными понятиями и просмотреть демонстрации использования методов работы с исчерпывающими примерами.

Вывод. Представленные выше алгоритмы являются лишь частными случаями огромного количества возможностей, открывающихся перед пользователем Dynamo. Воплощение данных возможностей позволяет раскрыть весь современный потенциал информационного моделирования. Разработана 3D модель многоэтажного здания с применением скриптов визуального программирования, автоматизирующих выполнение рутинных процессов. Реализация всех вышеизложенных сценариев занимает минимальное количество времени, что подтверждает целесообразность применения Dynamo в качестве мощного инструмента автоматизации задач, которые стоят перед проектировщиками.

Эффективность применения таких сценариев существенно увеличивается при выполнении большого количества повторяющихся однотипных операций, разработке типовых объектов или построении сложной параметрической геометрии. Возможности Dynamo ограничены только возможностями самого проектировщика. Среда визуального программирования поддерживает выполнение внутренних узлов, содержащих фрагменты кода Python, что открывает неограниченные возможности ее применения. Полная и сформированная база сценариев позволяет уменьшить сроки проектирования.

Библиографический список:

- 1. Смакаев Р.М., Низина Т.А. Автоматизация задач проектирования с помощью среды визуального программирования DYNAMO STUDIO [Электронный ресурс] // Огарев-online. - 2020. - №3.
- 2. Калиниченко А.В. О расширении функциональности системы autocad на приме-ре автоматизации проектирования водоснабжения жилых зданий Наука и бизнес: пути развития. 2016. № 6 (60). С. 12-15.
- Поддорогина Е. А., Шумилов К. А., Мазинг А. А. Разработка строительных объектов в DYNAMO REVIT // BIMмоделирование в задачах строительства и архитектуры: материалы Всерос. науч.-практ. конф., 29-30 марта 2018 г. – СПб: СПбГАСУ, 2018. – С. 177–182.
- Поддорогина Е. А., Шумилов К. А., Мазинг А. А. Разработка строительных объектов в DYNAMO REVIT // ВІМмоделирование в задачах строительства и архитектуры: материалы Всерос. науч.-практ. конф., 29-30 марта 2018 г. – СПб: СПбГАСУ, 2018 – С. 177–182.
- 5. Официальный сайт поддержки продукта Dynamo Studio [Электронный ресурс]. Режим доступа: https://www.dynamoprimer.com/index.html

References:

- 1. Smakaev R.M., Nizina T.A. Automation of design tasks using DYNAMO STUDIO visual programming [Electronic resource] *Ogarev-online*. 2020; 3. (In Russ)
- 2. Kalinichenko A.V. On expanding the functionality of the autocad system using the automation of the design of water supply for residential buildings Science and business: development path. 2016; 6 (60): 12-15. (In Russ)
- Poddorogina E. A., Shumilov K. A., Mazing A. A. Development of construction objects in DYNAMO REVIT // BIMmodeling in construction tasks and architecture: materials of All-Russian Scientific and Practical Conference, March 29-30, 2018 St. Petersburg: SPbGA-SU, 2018; 177-182. (In Russ)
- Poddorogina E. A., Shumilov K. A., Mazing A. A. Development of construction objects in DYNAMO REVIT // BIMmodeling in construction tasks and architecture: materials of All-Russian Scientific and Practical Conference, March 29-30, 2018 – St. Petersburg: SPbGA-SU, 2018; 177-182. (In Russ)
- 5. https://www.dynamoprimer.com/index.html Official website of Dynamo Studio product support [El.res.]. Access mode: Сведения об авторах:
 - Юрошева Татьяна Александровна, кандидат технических наук, доцент; trini-83@yandex.ru Калиниченко Алла Викторовна, кандидат технических наук, доцент; kalinichenkoalla@mail.ru Макиев Владимир Георгиевич, магистр, volodya.maki@yandex.ru **Information about authors:**

Tatiana A. Yurosheva, Cand. Sci. (Eng), Assoc. Prof.; trini-83@yandex.ru

Alla V. Kalinichenko, Cand. Sci. (Eng), Assoc. Prof.; kalinichenkoalla@mail.ru

Vladimir G. Makiev, Master, volodya.maki@yandex.ru

Конфликт интересов/Conflict of interest.

Авторы заявляют об отсутствии конфликта интересов/The authors declare no conflict of interest.

Поступила в редакцию/ Received 15.09.2022.

Одобрена после рецензирования / Reviced 12.10.2022.

Принята в печать /Accepted for publication 12.10.2022.