



УДК 004.94:692.25

Э. Г. ЮМАТОВА, д-р пед. наук, доц., проф. кафедры инженерной графики и информационного моделирования; Д. М. ЛАРИОНОВ, магистрант кафедры инженерной графики и информационного моделирования

ЗАДАЧИ АВТОМАТИЗАЦИИ В СИСТЕМАХ ИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ В СТРОИТЕЛЬСТВЕ

ФГБОУ ВО «Нижегородский государственный архитектурно-строительный университет».

Россия, 603952, г. Н. Новгород, ул. Ильинская, д. 65.

Тел.: (831) 430-54-00; эл. почта: yumatova.evelina@gmail.com

Ключевые слова: цифровые модели, автоматизация, дополнительные свойства и функции, программная реализация, системы информационного моделирования.

В статье рассматриваются проблемы автоматизации передачи данных с использованием сборок, динамически типизированных языков программирования и визуальных сред в ТИМ-системах разного уровня. Предложен алгоритм и на его основе программная реализация автономной передачи дополнительных свойств строительных изделий в Renga. Предложен алгоритм и плагин для автоматизации оформления проектной документации, выполнения расчетов с применением Revit API.

Введение

Программные системы автоматизации отдельных функций в строительстве на стадиях подготовки проектной и исполнения рабочей документации могут реализовать такие задачи:

- оптимизация управления проектами (составление календарных планов, финансирование, выдача разрешения на проект, отчетность, визуальная документация, идентификация сотрудника и др.);

- программная интеграция, что предполагает применение различного ПО на стадии проектирования при разработке цифрового двойника, начиная с *Progect*, затем работая в нескольких ТИМ-системах одновременно и обмениваясь информацией в *ifc*-формате на сервере (Нанокад, Автокад, *Renga*, *ArchiCAD*, *Revit*, *Scad*, Лира);

- аналитика информационной модели с применением экспертных систем для выявления наличия геометрических коллизий, сроков исполнения технического задания и передачи *ifc*-модели органам Госэкспертизы (*PilotBim*, *InterBridge*, *NavisWorks*) в соответствии с [1] и СП 333.1325800.2020 «Информационное моделирование в строительстве. Правила формирования информационной модели объектов на различных стадиях жизненного цикла»;

- оптимизация выполнения строительно-монтажных работ с применением средств робототехники, например, для вязки арматуры в каркасно-монолитных перекрытиях или перемещения небольшого объема кирпичей в малоэтажном строительстве;

- программно-техническая интеграция технологий ИИ, ТИМ и средств 3D-визуального контроля на всех рабочих стадиях реализации проекта для анализа исполнения проектной документации, контроля за техникой безопасности



на площадке и сроков строительства (Видеоинтеллект, ИМПульс, ОКО, Техзор и др.) [2, 3];

– автоматизация существующих процессов и запуск новых.

Отметим, что в настоящее время за рубежом в строительстве применяется ПО на основе ИИ, комплексно решающее перечисленные задачи, такие как *OpenSpace*, *Foresit*, *Procure*, *Alise Technologies*, *Doxel*, *Buildots*. В России список разработанного и лицензированного ПО с применением ИИ для решения отдельных видов задач размещен на сайте Минстроя РФ (34 программы) [4]. Очевидно, что центральным звеном таких систем в строительстве является информационная модель объекта строительства или реконструкции, разработанная средствами ТИМ-технологий.

Проблема исследования и программные способы ее решения

При решении каждой из перечисленных задач могут возникать проблемы, требующие расширения программных возможностей ТИМ-инструментария, например, за счет автоматизации процессов (выполнение расчетов, оптимизация построения узлов модели, маркировка элементов, разработка ориентированных под разделы шаблонов, создание фильтров и на их основе ведомостей, спецификаций и др. [5, 6]). У каждого строительного элемента в системе задаются следующие данные: тип, геометрические параметры, расчетные характеристики и свойства [7, 8]. Данные могут изменяться в зависимости от технического задания. В результате одна из проблем для автоматизации возникает при работе нескольких участников над проектом в различных средах, связанная с передачей наборов данных элементов ТИМ-моделей через сервер или автономно. Такая автоматизация важна как для проектировщиков, так и для ТИМ-координаторов, и может быть реализована с помощью написания плагинов и скриптов.

Разработка и загрузка дополнительных приложений может осуществляться двумя возрастающими по сложности реализации способами: 1) написание скриптов на динамически типизированных языках (*Python*, *VBScript* и др.); 2) написание плагина на языках программирования *C++* или *C#*. В первом случае плагин запускается из автономного приложения, во втором – из ТИМ-программы. Примерами реализованных плагинов-шаблонов второго типа являются: обозреватель модели (*Renga*); Топоплан и *TBS Plus* для обработки результатов сканирования местности лазерными устройствами и ГИС-данных (Нанокэд); *RoofMaker* и *TrussMaker* для автоматизации разработки разделов КД и КМ (*ArchiCAD*); *Navisworks* для разработки ПД по разделам ПОР, ППР, *MagiCad* (*Revit*).

Наиболее простым и доступным способом автоматизации различных процессов для пользователя в ПО *Renga* является написание скриптов с применением языка *Python*. Рассмотрим алгоритм написания автономного приложения (т. е. запускаемого из командной строки), позволяющего расширять наборы свойств таких элементов строительных конструкций, как Стена, Балка, Колонна, Перекрытие, Крыша, Дверь, Окно.

Перечисленные объекты в *Renga* имеют свойства, каждое из которых задается следующими характеристиками:

– имя свойства, идентификатор свойства;

– тип свойства (действительное или целое число, строка, длина, площадь, объем, масса, длина, угол, булевый, логический, перечисление). Для свойства Перечисление*Enum* задается список значений. Для свойства Строка*String*



возможно задание формулы (*SetExpression*), определяющей математическую зависимость свойств. В случае применения операции копирования проекта не применяемые свойства будут накапливаться.

Для автоматизации решения таких задач скрипт реализует следующий алгоритм: 1) из командной строки в папке установки *Python* запустить *pip install ruwin32*; 2) загрузить скрипт и шаблон *JSON*-файла; 3) отредактировать *JSON*-файл для создания набора свойств; 4) файл-проект и *JSON*-файл со свойствами запустить из командной строки (рис. 1).

```
create_properties.py -- "C:\Projects\Multifunctional_building" --  
properties "C:\Projects\Properties_front"
```

Рис. 1. Строка запуска файла-проекта и *JSON*-файла

Фрагмент *JSON*-файла создания свойств строительных объектов, как «Тип слоя» (тип данных – перечисление) и «Площадь наружной отделки» для Колонны, Балки (тип данных – выражение), приведен на рис. 2.

```
{  
  "Тип слоя":  
    "ID": "14dbb99e-9024-4c92-933f-efa9e71d3a3a",  
    "property type": "Enum",  
    "list": ["Наружная отделка", "Внутренняя отделка"],  
    "object type": [{"0abcb18f-0aaf-4509-bf89-5c5fad9d5d8b"},  
                    {"0f0adba0-5c06-46c0-9c8a-b9d69ef1251f"},  
                    {"4329112a-6b65-48d9-9da8-abf1f8f36327"}, {"f5bd8bd8-39c1-47f8-8499-f673c580dfbe"},  
                    {"d9ee2442-e807-42fb-8fe5-9dcfe543035d"}, {"63478188-7c88-4a6d-b891-9725f04a5bc7"},  
                    {"bac4470f-d560-4f57-a49e-faa5f6e5a279"},  
                    {"1cfba99c-01e7-4078-ae1a-3e2ff0673599"}.{"80c34eed-5232-4cd7-8925-177ee02934d8"}],  
  }  
  { "Площадь наружной отделки":  
    "ID": "543e4fa0-bfb5-45a2-acdf-273a26c3a338",  
    "property type": "Area",  
    "SetExpressionS": ["Стиль колонны.Периметр отделки * Номинальная длина/1000000",  
                      "Стиль балки.Периметр отделки * Номинальная длина/1000000"],  
    "object type": [{"be49a354-19b7-435a-8957-9ef8782630c2"},  
                    {"cf2b8b04-f595-4432-98f4-8234c95adbdd"},  
                    {"be49a354-19b7-435a-8957-9ef8782630c2"}, {"63478188-7c88-4a6d-b891-9725f04a5bc7"}],  
  },  
}
```




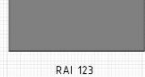

Рис. 2. *JSON*-файл. Включение дополнительных свойств строительных элементов

Конечным результатом работы скрипта станет разработка новых фильтров свойств строительных объектов для получения возможности добавления новых функций – расчета или таблиц.

Пример фрагмента Ведомости отделки элементов фасада после автоматической загрузки свойств объекта Стена показан на рис. 3.



Параметры		Свойства	
Свойство	Значение		
URL	http:		
Категория объекта	K2		
Код классификатора	91.060.10		
Наименование классификато...	Несущие		
Обозначение	ФАС-01		
Описание кода классификато...	ТН_Фасад Вент (Тем		
Поз.отделки	2		
Тип слоя	Наружная отделка		
Фильтр систем TN	Фасад		
Эталон цвета	RAI 211		

Ведомость отделки фасадов			
Наименование элементов фасада	Наименование материала	Наименование и номер эталона цвета, образец колера	Площадь отделки, м2
Стена 400,00 мм	ФАС-01_ТН-ФАСАД Вент	 RAI 234	861,48
Стена 400,00 мм	ФАС-01_ТН-ФАСАД Вент (тёмный)	 RAI 211	488,70
Стена 80,00 мм	Облицовка цоколя	 RAI 215	42,47
Стена 320,00 мм	ФНД-02-05_ТН-ФУНДАМЕНТ Стандарт Оптима_цоколь	 RAI 123	252,08
Стена 200,00 мм	ФНД-02-05_ТН-ФУНДАМЕНТ Стандарт Оптима_цоколь	 RAI 123	2,80

ННГАСУ. ИИТ-2025-AP

Рис. 3. Ведомость отделки фасада многофункционального здания. Фрагмент

В отличие от *Renga*, добавление функций в ПО *Revit* осуществляют средствами наглядного программирования *Dynamo* (*VPL*-среда), а именно с помощью графических схем (ноды – узлы – функции, связи – процессы – перемещение данных). Фрагмент схемы автоматизации редактирования шаблонов, видов, экспликации по СПДС представлен в табл. 1.

Таблица 1

Схема присоединения узлов скрипта оформления ПД. Фрагмент

Задачи – функции	Узлы – перемещение данных
1. Редактировать шаблон листа	Get viewports from Template Sheets – Filter legends – Viewports center
2. Выровнять положение видов	Get viewports from Sheets – filter legends – Get center point from Viewports – Viewports center
3. Выровнять положение экспликации	Get viewports from Sheets – Viewports center – Get center point from ViewSchedule
4. Установить ширину столбца экспликации	Get selection – Set Schedule on Sheet – ViewSchedule. SetcolumnWidth

Данные скрипты на основе ПО *Python* реализуют задачи редактирования и параметризации шаблонов, рабочих наборов, формообразующих и связанных файлов [9]. К недостаткам функционирования скриптов, написанных на типизированных языках, следует отнести: версионность, линейность и отсутствие разветвленных алгоритмов, ограничения при интеграции различного ПО в ТИМ-систему и работе с большими объемами данных, трудности в реализации согласованности с *Revit API* или *Renga API*. При разработке плагинов для добавления добавочных функций в ТИМ-систему на языках *C++*, *C#* перечисленные недостатки могут быть устранены. Такой плагин должен содержать 2 файла: код компиляции (*.NET*-сборка); *XML*-файл описания плагина и инструкцию по загрузке (*.rndesc*). Для *Renga XML*-файл размещается в папке: `[RengaInstallation]\Plugins\Sample` с расширением *.rndesc*. Фрагмент плагина



автоматизации составления коммерческого предложения в части спецификации по вычислению планируемых объемов и сроков работ, затрат на основные и расходные материалы показан на рис. 4.

```
import clr
clr.AddReference('RevitAPI')
clr.AddReference('RevitServices')

from RevitServices.Persistence import DocumentManager
from Autodesk.Revit.DB import *
import sys

sys.path.append(r'C:\Program Files (x86)\Microsoft Office\root\Office16')
clr.AddReference('Microsoft Office.Interop.Excel')
import Microsoft Office.Interop.Excel as Excel

doc = DocumentManager.Instance.CurrentDBDocument

def simple_export(elements, parameters, file_path):
    data = [[ID] + parameters]
    for element in elements:
        row = [element.ID.IntegerValue]
        for param_name in parameters:
            param = element.LookupParameter(param_name)
            if param:
                row.append(param.AsValueString() or **)
            else:
                row.append("N\A")
        data.append(row)
```

Рис. 4. Автоматизация разработки экспликации с выгрузкой из *Revit* в *Excel*. Фрагмент

NET-сборка реализует интеграцию с *Revit.API*, а также с *Excel* для выгрузки параметров модели строительного объекта из *Revit*. При этом недостаток работы данного плагина – замедление работы основного ПО.

Заключение

Рассмотрены особенности *IT*-разработки на языках высокого уровня и типизированных языках автоматизации работы с применением классификатора процессов взаимодействия *API*: 1) ТИМ-координатора по добавлению рабочих наборов, разработке файла составления параметров и объектов, оптимальному размещению оборудования, фильтрации данных при коллизиях, размещению проекта по разделам на сервере, обеспечению прав доступа, подготовке шаблона с добавлением параметров; 2) инженера-проектировщика по оформлению и нормоконтролю проектной документации, разработке параметрических формообразующих семейств. Разработка скриптов и плагинов осуществлялась для разного уровня ТИМ-систем (*Renga*, *Revit*), выделены их особенности и недостатки, приведены примеры реализации.



БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Требования к информационным моделям объектов капитального строительства. Часть 1. Общие требования к цифровым моделям зданий для прохождения экспертизы при использовании технологии информационного моделирования. – URL: https://www.mos.ru/upload/documents/files/1115/01_ObshietrebovaniyakCMzdanii_40.pdf. – Текст : электронный.
2. Примеры применения искусственного интеллекта в строительной отрасли. – Текст : электронный // Все о стройке : независимая площадка деволпмента России и стран СНГ : [сайт]. – URL: <https://всеостройке.рф/primery-primeneniya-iskusstvennogo-intellekta-v-stroitelnoj-otrasli/>.
3. Юматов, В. А. Искусственный интеллект в судебной экспертологии: проблемы и перспективы / В. А. Юматов, Э. Г. Юматова // Вестник Нижегородского университета им. Н. И. Лобачевского. – Нижний Новгород, 2025. – № 4. – С. 137–146.
4. Программное обеспечение на основе ИИ. – Текст : электронный // Минстрой России : [сайт]. – URL: <https://minstroyrf.gov.ru/iskusstvennyy-intellekt-v-stroitelnoy-otrasli/>.
5. Ларионов, Д. М. Модификация шаблонов в информационной среде Revit / Д. М. Ларионов, Э. Г. Юматова. – Текст : электронный // Международная конференция Нижегородского фестиваля науки, Нижний Новгород, 24–25 октября 2024 г. : сборник докладов. – Нижний Новгород : ННГАСУ, 2025. – С. 129–131. – URL: https://nngasu.ru/science/SMU/sbornik/NFN-2024_Sbornik.pdf.
6. Применение BIM-системы Renga для создания информационной модели цеха для использования при техническом перевооружении. – Текст : электронный // САПР и графика. – 2020. – № 6 (284). – С. 32–35. – URL: <https://sapr.ru/article/26068>.
7. Голованов, Н. Н. Геометрическое моделирование / Н. Н. Голованов. – Москва : ДМК Пресс. – 2024. – 408 с. – ISBN: 978-5-93700-304-1.
8. Юматова, Э. Г. Задачи структуризации данных в системах информационного моделирования в строительстве / Э. Г. Юматова, Н. Д. Чучмар, Д. М. Ларионов // Приволжский научный журнал / Нижегородский государственный архитектурно-строительный университет. – Нижний Новгород, 2025. – № 1 (73). – С. 351–356.
9. Создание скрипта в Дунано. – Текст : электронный // Образовательная платформа по BIM, CAD, CG : [сайт]. – URL: <https://autocad-specialist.ru>.

YUMATOVA Evelina Gennadievna, doctor of pedagogical sciences, associate professor, professor of the chair of engineering graphics and information modeling; LARIONOV Dmitry Mikhailovich, master degree student of the chair of engineering graphics and information modeling

STANDARDIZATION OF QUALITY CONTROL OF DESIGN WORKS IN CONSTRUCTION USING INFORMATION MODELING TECHNOLOGY

Nizhny Novgorod State University of Architecture and Civil Engineering.

65, Pjinskaya St., Nizhny Novgorod, 603952, Russia.

Tel.: (831) 430-54-00; e-mail: yumatova.evelina@gmail.com

Key words: digital model, automation, additional properties and functions, program development, BIM.



The article discusses the challenges of automating data transfer using assemblies, dynamically typed programming languages, and virtual environments in various levels of TIM systems. An algorithm is proposed and, based on it, an autonomous plugin for transferring additional properties of building products to Renga. An algorithm and software implementation are proposed for automating the design of project documentation and performing calculations using the Revit API.

REFERENCES

1. Trebovaniya k informatsionnym modelyam obektov kapitalnogo stroitelstva. Chast 1. Obshchie trebovaniya k tsifrovym modelyam zdaniy dlya prokhozheniya ekspertizy pri ispolzovanii tekhnologii informatsionnogo modelirovaniya [Requirements for information models of capital construction objects. Part 1. General requirements for digital building models for expert examination using information modeling technology]. URL: https://www.mos.ru/upload/documents/files/1115/01Obshietrebovaniyak-CMzdanii_40.pdf (accessed: 12.10.25).
2. Primery primeneniya iskusstvennogo intellekta v stroitelnoy otrasli [Examples of the application of artificial intelligence in the construction industry]. Vse o stroyke: nezavisimaya ploshchadka razvitiya Rossii i stran SNG [All about Construction: independent development platform of Russia and CIS countries]. URL: <https://vseostroyke.pf/primery-primeneniya-iskusstvennogo-intellekta-v-stroitelnoy-otrasli/>.
3. Yumatov V. A., Yumatova E. G. Iskusstvennyy intellekt v sudebnoy ekspertologii: problemy i perspektivy [Artificial intelligence in forensic expertology: problems and prospects]. Vestnik Nizhegorodskogo universiteta im. N. I. Lobachevskogo [Vestnik of Lobachevsky University of Nizhni Novgorod]. Nizhny Novgorod, 2025, № 4, P. 137–146.
4. Programmnoye obespechenie na osnove II [AI-based software]. Minstroy Rossii [Ministry of Construction of Russia]: [website]. URL: <https://minstroyrf.gov.ru/iskusstvennyy-intellekt-v-stroitelnoy-otrasli/>.
5. Larionov D. M., Yumatova E. G. Modifikatsiya shablonov v informatsionnoy srede Revit [Modification of templates in the Revit information environment]. Mezhdunarodnaya konferentsiya Nizhegorodskogo festivalya nauki [International Conference of the Nizhny Novgorod Science Festival]. Nizhny Novgorod, 24–25 oktyabrya 2024 g. : sbornik dokladov. Nizhny Novgorod, NNGASU, 2025, P. 129–131. URL: https://nngasu.ru/science/SMU/sbornik/NFN-2024_Sbornik.pdf.
6. Primenenie BIM-sistemy Renga dlya sozdaniya informatsionnoy modeli tsekha dlya ispolzovaniya pri tekhnicheskoy perevooruzhenii [Application of the Renga BIM system for creating an information model of a workshop for use in technical re-equipment]. SAPR i grafika [CAD and Graphics], 2020, № 6 (284), P. 32–35. URL: <https://sapr.ru/article/26068>.
7. Golovanov N. N. Geometricheskoye modelirovaniye [Geometric modeling]. Moscow, DMK Press, 2024, 408 p. ISBN: 978-5-93700-304-1.
8. Yumatova E. G., Chuchmar N. D., Larionov D. M. Zadachi strukturizatsii dannykh v sistemakh informatsionnogo modelirovaniya v stroitelstve [Problems of data structuring in information modeling systems in construction]. Privolzhskiy nauchnyy zhurnal [Privolzhsky Scientific Journal]. Nizhny Novgorod, Nizhegorodskiy gosudarstvennyy arkhitekturno-stroitelnyy universitet, 2025, № 1 (73), P. 351–356.
9. Sozdaniye skripta v Dynamo [Creating a script in Dynamo]. Obrazovatel'naya platforma po BIM, CAD, CG [Educational platform on BIM, CAD, CG]. URL: <https://autocad-specialist.ru> (accessed: 11.04.25).

© Э. Г. Юматова, Д. М. Ларионов, 2026

Получено: 02.12.2025 г.