

Вестник Евразийской науки / The Eurasian Scientific Journal <https://esj.today>

2026, Том 18, № 4 / 2026, Vol. 18, Iss. 4 <https://esj.today/issue-4-2026.html>

URL статьи: <https://esj.today/PDF/12SAVN426.pdf>

DOI: 10.15862/12SAVN426 (<https://doi.org/10.15862/12SAVN426>)

2.1.14. Управление жизненным циклом объектов строительства (технические науки)

Ссылка для цитирования этой статьи:

Вареник, К. А. Автоматизация управления данными информационных моделей объектов капитального строительства на стадии проектирования инженерных сетей с использованием среды визуального программирования Dynamo / К. А. Вареник, Е. М. Ковалева, С. Кошавкин, А. С. Чамеев, В. И. Кокин // Вестник евразийской науки. — 2026. — Т. 18. — № 4. — URL: <https://esj.today/PDF/12SAVN426.pdf>. DOI: 10.15862/12SAVN426.

For citation:

Varenik K.A., Kovaleva E.M., Koshavkin S., Chameev A.S., Kokin V.I. Automation of data management in information models of capital construction objects at the engineering networks design stage using the Dynamo visual programming environment. *The Eurasian Scientific Journal*. 2026;18(4): 12SAVN426. Available at: <https://esj.today/PDF/12SAVN426.pdf>. DOI: 10.15862/12SAVN426. (In Russ., abstract in Eng.).

УДК 721.021.23

Вареник Кирилл Александрович

ФГБОУ ВО «Новгородский государственный университет имени Ярослава Мудрого», Великий Новгород, Россия
Доцент

Кандидат технических наук, доцент

E-mail: vkirillv89@mail.ru

ORCID: <https://orcid.org/0009-0004-0604-0465>

РИНЦ: https://elibrary.ru/author_profile.asp?id=721765

Ковалева Екатерина Михайловна

ФГБОУ ВО «Новгородский государственный университет имени Ярослава Мудрого», Великий Новгород, Россия
E-mail: c.kowaliowa2014@yandex.ru

Кошавкин Станислав

ООО «Ленмонтаж», Санкт-Петербург, Россия

Ведущий инженер-конструктор

E-mail: stanislavkoshavkin@gmail.com

Чамеев Александр Сергеевич

ФГБОУ ВО «Новгородский государственный университет имени Ярослава Мудрого», Великий Новгород, Россия
Инженер

E-mail: s244313@std.novsu.ru

РИНЦ: https://elibrary.ru/author_profile.asp?id=1239941

Кокин Владимир Ильич

ФГБОУ ВО «Новгородский государственный университет имени Ярослава Мудрого», Великий Новгород, Россия
Инженер

E-mail: s253019@std.novsu.ru

**Автоматизация управления данными информационных
моделей объектов капитального строительства на стадии
проектирования инженерных сетей с использованием
среды визуального программирования Dynamo**

Аннотация. В научной статье рассматриваются уникальные авторские методы и структурные алгоритмы автоматизации управления данными цифровых информационных моделей объектов капитального строительства на этапе проектирования инженерных сетей. Актуальность исследования обусловлена необходимостью минимизации проектных ошибок и оптимизации рутинных процессов при разработке систем водоснабжения и канализации. Авторами детально описана внутренняя структура двух разработанных функциональных скриптов в среде визуального программирования Dynamo, содержащих блоки исходного кода на языке Python. Первый алгоритм полностью автоматизирует гидравлический расчет и подбор диаметров трубопроводов канализации согласно требованиям свода правил СП 30.13330.2020. Второй алгоритм обеспечивает сквозное автоматическое выведение наименований вложенных компонентов инженерных систем из родительских семейств напрямую в результирующую спецификацию оборудования. В работе приводятся подробные результаты верификации обоих алгоритмов на реальной информационной модели здания. В ходе тестирования исследуется проблема возникновения пространственных геометрических коллизий в узлах соединительных деталей фитингов, вызванная автоматическим изменением параметров труб. На основе проведенного исследования авторами дается комплексный сравнительный анализ эффективности внедрения средств визуального программирования в практику проектирования инженерных систем. Оценивается роль автоматизации в контексте оптимизации информационных процессов, повышения точности сметных расчетов и снижения рисков человеческого фактора на ранних стадиях жизненного цикла объектов капитального строительства. Доказано, что использование авторских скриптов обеспечивает стандартизацию проектных процедур и гарантирует высокую надежность данных.

Ключевые слова: управление жизненным циклом; информационное моделирование; инженерные сети; проектирование; гидравлический расчет; визуальное программирование; скрипт

Введение

Концепция управления жизненным циклом объектов капитального строительства предполагает непрерывность, актуальность и точную передачу информационных потоков между всеми фазами проектирования [1; 2]. Этап проектирования инженерных сетей и систем является критически важным звеном жизненного цикла, так как именно здесь закладываются ключевые параметры надежности, стоимости, энергоэффективности и эксплуатационной пригодности будущего объекта. Ошибки и неточности в цифровых информационных моделях коммуникаций, допущенные на данной стадии, неизбежно трансформируются в финансовые и временные потери на этапах строительства и последующей эксплуатации здания [3].

Разработка разделов водоснабжения и канализации связана с необходимостью обработки колоссальных массивов расчетных и атрибутивных данных. Традиционные «ручные» методы верификации, проведения гидравлических расчетов и формирования спецификаций оборудования характеризуются высокой трудоемкостью и подверженностью человеческому фактору. Это снижает общую эффективность управления данными модели, затрудняет её валидацию и оперативную актуализацию при изменении технического задания.

Позапное внедрение технологий информационного моделирования в строительной сфере требует поиска новых подходов к оптимизации рутинных процессов [4]. Использование технологий визуального программирования выступает эффективным системотехническим подходом к автоматизации принятия решений на всех стадиях жизненного цикла объекта [5], а также к оптимизации информационных потоков [6; 7]. Целью исследования является разработка авторами, тестирование и оценка эффективности специализированных алгоритмов управления

данными моделей информационного моделирования на стадии проектирования инженерных сетей для обеспечения высокого качества проектных решений в рамках единого жизненного цикла объекта капитального строительства.

Материалы и методы

В рамках исследования авторами были разработаны два независимых алгоритма (скрипта) в среде визуального программирования Дупамо, интегрированной в современный программный комплекс Autodesk Revit [8; 9]. Данные инструменты содержат встроенные скрипты на языке Python и предназначены для автоматизации обработки и валидации данных цифровой модели инженерных сетей здания.

Алгоритм автоматического расчета канализации по СП 30.13330.2020

Процесс автоматизации подбора диаметров трубопроводов базируется на нормативных требованиях и гидравлических формулах свода правил СП 30.13330.2020 «Внутренний водопровод и канализация зданий». Математическая модель, заложенная в разработанный авторами алгоритм, выполняет вычисление теоретического диаметра и скорости потока на основе геометрических параметров участка, уклона, коэффициента шероховатости, а также максимального секундного расхода стоков. Величина расхода определяется с учетом вероятности действия санитарно-технических приборов, рассчитываемой на основании общего числа жителей и установленного количества приборов. Для апробации методики была использована цифровая информационная модель канализационной сети (рис. 1).

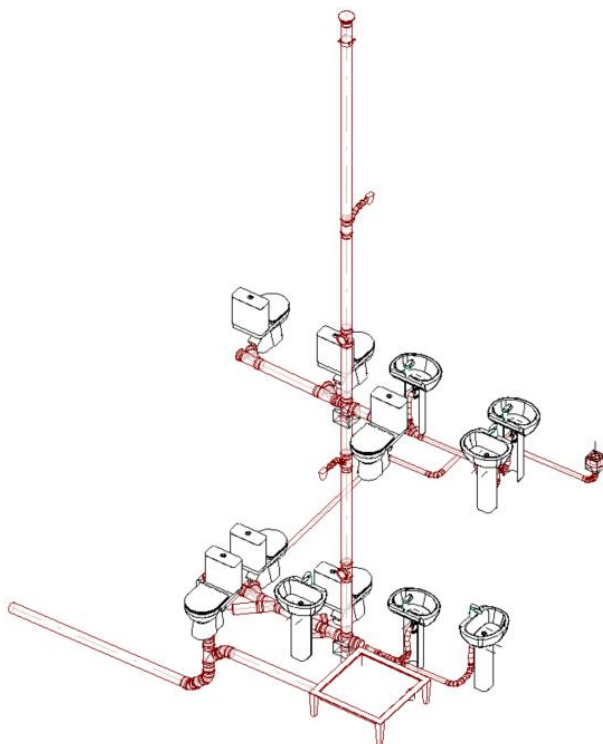
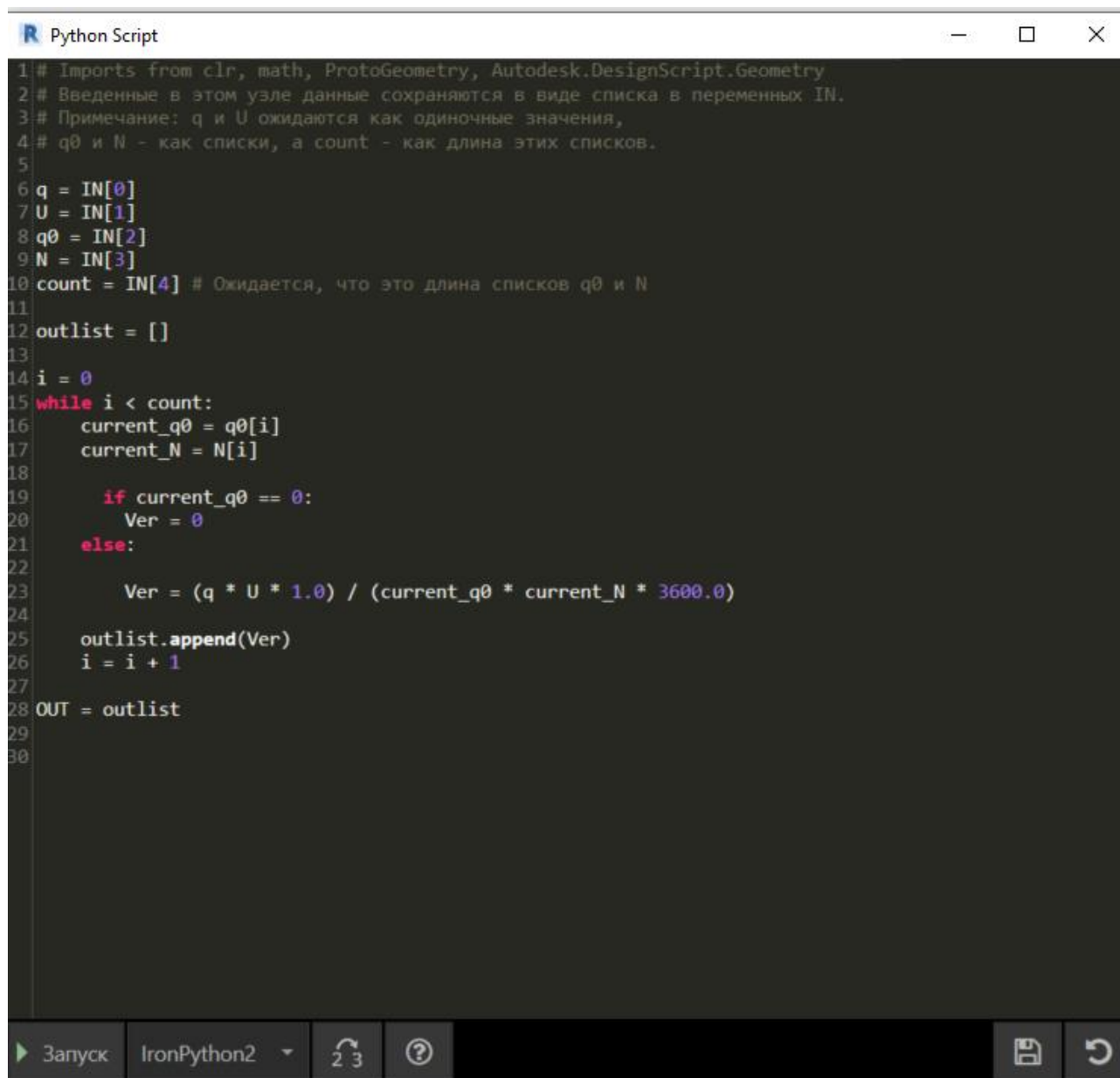


Рисунок 1. Цифровая информационная модель. 3D вид (рисунок авторов)

Для запуска алгоритма в параметрах модели Revit предварительно задаются нормативные секундные расходы для различных типов санитарно-технических приборов (унитазов, раковин, душевых поддонов). Также в модель добавляются два дополнительных параметра трубопроводов: число приборов на участке и скорость воды.

Структура информационного алгоритма в Динамо:

1. Экспорт параметров: выбирается категория «Трубы», из элементов которой считываются текущие атрибуты: «Число приборов», «Длина», «Расход приборов» и «Уклон».
2. Расчет вероятности действия приборов: реализован через разработанный авторами программный скрипт на языке Python, который принимает в качестве переменных расход в час наибольшего водопотребления, число жителей, расход приборов и количество трубопроводов (рис. 2).



```
1 # Imports from clr, math, ProtoGeometry, Autodesk.DesignScript.Geometry
2 # Введенные в этом узле данные сохраняются в виде списка в переменных IN.
3 # Примечание: q и U ожидаются как одиночные значения,
4 # q0 и N - как списки, а count - как длина этих списков.
5
6 q = IN[0]
7 U = IN[1]
8 q0 = IN[2]
9 N = IN[3]
10 count = IN[4] # Ожидается, что это длина списков q0 и N
11
12 outlist = []
13
14 i = 0
15 while i < count:
16     current_q0 = q0[i]
17     current_N = N[i]
18
19     if current_q0 == 0:
20         Ver = 0
21     else:
22         Ver = (q * U * 1.0) / (current_q0 * current_N * 3600.0)
23
24     outlist.append(Ver)
25     i = i + 1
26
27 OUT = outlist
28
29
30
```

Рисунок 2. Скрипт Python «Вероятность действия приборов» (рисунок авторов)

3. Линейная интерполяция: нормативные коэффициенты из приложения к своду правил перенесены во внешнюю таблицу Excel. Поскольку расчетное произведение количества приборов на вероятность их действия часто принимает промежуточные значения, авторами интегрирован встроенный Python-скрипт для автоматической аппроксимации и интерполяции данных (рис. 3).

```
Python Script
1 # Загрузить стандартную библиотеку Python и библиотеку DesignScript
2 # Эти импорты необходимы для работы с геометрией Dynaмо.
3 # Для данной конкретной математической операции они не используются,
4 # но сохраняются, если скрипт является частью более крупного проекта.
5 import clr
6 clr.AddReference('ProtoGeometry')
7 from Autodesk.DesignScript.Geometry import *
8
9
10 lookup_ranges = IN[0][0]
11 lookup_coeffs = IN[0][1] # Список соответствующих коэффициентов K (например, [1.0, 1.2, 1.5, 1.8])
12 np_values_to_interpolate = IN[3] # Список значений NP, для которых нужно найти коэффициент (например, [5, 15, 25])
13
14 num_np_to_process = IN[1] # Длина списка np_values_to_interpolate
15 num_coeffs = IN[2] # Длина списков lookup_ranges и lookup_coeffs
16
17 # Разместите код под этой строкой
18 outlist = []
19
20 i = 0
21 while i < num_np_to_process:
22     current_np_value = np_values_to_interpolate[i]
23     j = 0
24
25     # Внутренний цикл идет до предпоследнего элемента, чтобы всегда иметь j и j+1
26     while j < num_coeffs - 1:
27         NP1 = lookup_ranges[j]
28         NP2 = lookup_ranges[j+1]
29         K1 = lookup_coeffs[j]
30         K2 = lookup_coeffs[j+1]
31
32         # Проверяем, попадает ли текущее значение NP в данный диапазон [NP1, NP2)
33         if (current_np_value >= NP1) and (current_np_value < NP2):
34
35             denominator = NP2 - NP1
36             if denominator == 0:
37
38                 itog = K1
39             else:
40
41                 interpolated_fraction = (current_np_value - NP1) * 1.0 / (denominator * 1.0)
42                 itog = K1 + (K2 - K1) * interpolated_fraction
43
44             outlist.append(itog)
45             break
46         j += 1
47
48     i += 1
49
50 # Назначьте вывод переменной OUT.
51 OUT = outlist
```

Рисунок 3. Скрипт Python «Аппроксимация значений» (рисунок авторов)

Цепочка нодов для привязки файла Excel представлена на рисунке 4.

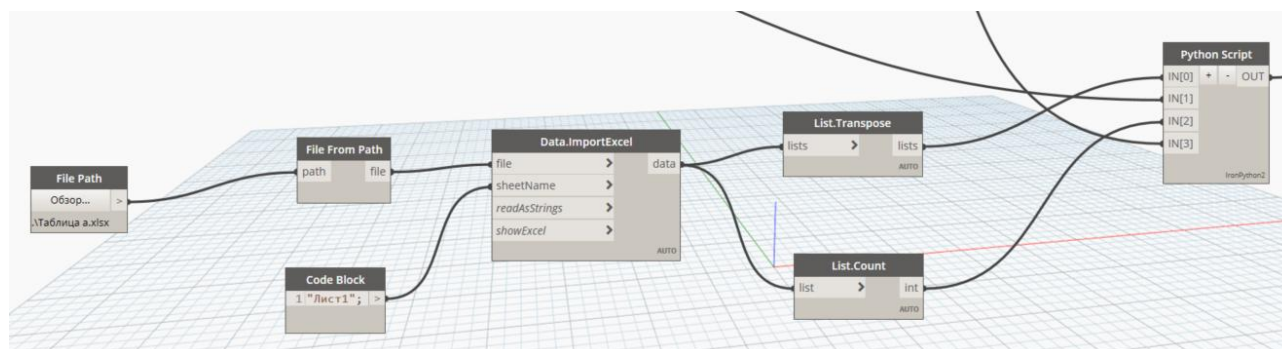


Рисунок 4. Цепочка нодов для привязки файла Excel (рисунок авторов)

4. Округление и запись: на основе гидравлических формул разработанный авторами Python-скрипт рассчитывает внутренний диаметр (рис. 5). Полученные значения автоматически округляются до ближайших реальных диаметров из сортамента труб, импортированного из файла Excel.



```
1# Загрузить стандартную библиотеку Python и библиотеку DesignScript
2import clr
3import math
4clr.AddReference('ProtoGeometry')
5from Autodesk.DesignScript.Geometry import *
6Alpha = IN[0] # Ожидается одиночное число (float/int)
7n = IN[1] # Ожидается одиночное число (float/int)
8qmax = IN[2] # Ожидается список чисел
9I = IN[3] # Ожидается список чисел, потенциально содержащий нечисловые значения
10countTrub = IN[4] # Ожидается целое число, длина списков qmax и I
11try:
12    numeric_Alpha = float(Alpha)
13    sinAlpha = math.sin(numeric_Alpha)
14except (ValueError, TypeError):
15    sinAlpha = 0.0
16    numeric_Alpha = 0.0 # Также установим numeric_Alpha, чтобы избежать ошибок далее
17outlist = [] # Более идиоматично использовать [] для создания пустого списка
18
19i = 0
20while i < countTrub:
21    current_qmax = 0.0 # Значение по умолчанию
22    current_I = None # Значение по умолчанию, чтобы понять, было ли оно преобразовано
23    itog = 0.0 # Значение по умолчанию для itog
24
25    try:
26        current_I_numeric = float(I[i])
27        current_qmax_numeric = float(qmax[i]) # Также проверяем qmax на всякий случай
28
29        if current_I_numeric > 0:
30
31            denominator_part_Alpha = numeric_Alpha - sinAlpha
32
33            if denominator_part_Alpha == 0:
34                itog = 0.0 # Или другое подходящее значение, если знаменатель равен 0
35            else:
36
37                chisl = math.pow(2, 26) * math.pow(numeric_Alpha, 4) * math.pow(n, 6) * math.pow(current_qmax_numeric, 6)
38
39                # current_I_numeric уже проверен на > 0
40                znam = math.pow(current_I_numeric, 3) * math.pow(denominator_part_Alpha, 10)
41
42                if znam == 0:
43                    itog = 0.0
44                else:
45                    a = chisl / znam
46                    itog = math.pow(a, 0.0625)
47                    itog = itog * 1000.0 # Явно умножаем на float для float-результата
48
49    except (ValueError, TypeError):
50
51        itog = 0.0 # Убедимся, что itog установлено в 0.0 в случае ошибки
52    except IndexError:
53
54        itog = 0.0
55
56    outlist.append(itog)
57    i += 1 # Более идиоматичный способ инкрементировать счетчик
58
59# Назначьте вывод переменной OUT.
60OUT = outlist
```

Рисунок 5. Скрипт Python «Расчет диаметров» (рисунок авторов)

Цепочка узлов для расчета угла представлена на рисунке 6.

Скрипт Python для округления диаметров до значений сортамента трубопроводов представлен на рисунке 7.

Финальный блок на базе кода Python производит вычисление скоростей движения стоков и автоматически перезаписывает итоговые диаметры и скорости обратно в информационные параметры элементов модели Revit (рис. 8).

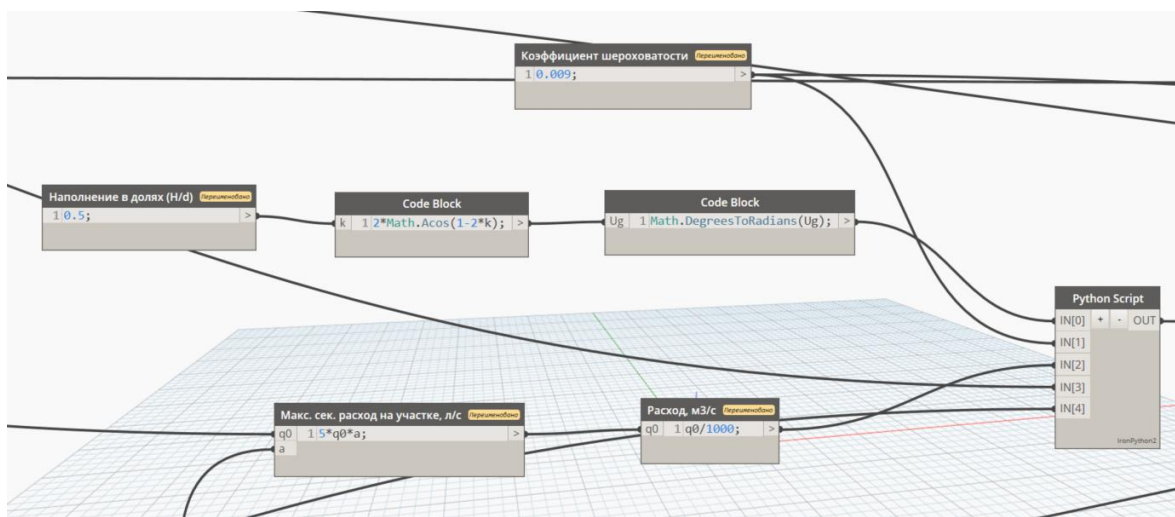


Рисунок 6. Цепочка узлов для расчета угла (рисунок авторов)

```
Python Script
1 # Загрузить стандартную библиотеку Python и библиотеку DesignScript
2 import clr
3 clr.AddReference('ProtoGeometry')
4 from Autodesk.DesignScript.Geometry import *
5
6 DiamRasch_raw = IN[0] if isinstance(IN[0], list) else []
7 DiamRasch_processed = []
8 for item in _DiamRasch_raw:
9     try:
10         DiamRasch_processed.append(float(item))
11     except (ValueError, TypeError):
12         DiamRasch_processed.append(0.0) # Если элемент не число, используем 0.0
13
14 DiamNomin_raw = IN[1] if isinstance(IN[1], list) else []
15 DiamNomin_processed = []
16 for item in _DiamNomin_raw:
17     try:
18         DiamNomin_processed.append(float(item))
19     except (ValueError, TypeError):
20         DiamNomin_processed.append(0.0) # Если элемент не число, используем 0.0
21
22 try:
23     final_count_rasch = len(DiamRasch_processed)
24 except (ValueError, TypeError):
25     final_count_rasch = 0
26
27 try:
28     final_count_nomin = len(DiamNomin_processed)
29 except (ValueError, TypeError):
30     final_count_nomin = 0
31
32 outlist = [] # Более идиоматично использовать [] для создания пустого списка
33 i = 0
34 while i < final_count_rasch:
35     current_rasch_diam = DiamRasch_processed[i] # Этот элемент уже гарантированно float
36     D = 0.0 # Инициализируем D как float для каждой итерации.
37     if current_rasch_diam == 0.0: # Сравниваем с float
38         D = 0.0
39     # 2. Обработка случаев с пустым или слишком коротким списком номинальных диаметров
40     elif final_count_nomin == 0:
41         D = 0.0 # Если нет номинальных диаметров, то и выбрать нечего
42     elif final_count_nomin == 1:
43         D = DiamNomin_processed[0]
44     else:
45         if current_rasch_diam <= DiamNomin_processed[0]:
46             D = DiamNomin_processed[0]
47         elif current_rasch_diam > DiamNomin_processed[final_count_nomin - 1]:
48             D = DiamNomin_processed[final_count_nomin - 1]
49         else:
50             j = 0
51             while j < final_count_nomin - 1:
52                 nomin_lower = DiamNomin_processed[j] # Гарантированно float
53                 nomin_upper = DiamNomin_processed[j+1] # Гарантированно float
54                 if current_rasch_diam > nomin_lower and current_rasch_diam <= nomin_upper:
55                     D = nomin_upper
56                     break # Важно: если найдено, выходим из внутреннего цикла
57                 j += 1 # Pythonic way to increment j
58
59     outlist.append(D)
60     i += 1 # Pythonic way to increment i
61
62 # Назначьте вывод переменной OUT.
63 OUT = outlist
```

Рисунок 7. Скрипт Python «Округления диаметров до значений сортамента трубопроводов» (рисунок авторов)

```
Python Script
1 # Загрузить стандартную библиотеку Python и библиотеку DesignScript
2 import clr
3 import math
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6 D_in = IN[0] # Список диаметров, может содержать нечисловые значения
7 n_in = IN[1] # Коэффициент n, может быть нечисловым
8 I_in = IN[2] # Список уклонов/градиентов, может содержать нечисловые значения
9 countTrub_in = IN[3] # Количество элементов, может быть нечисловым
10 outlist = [] # Более идиоматично использовать [] для создания пустого списка
11 n_val = 0.0 # Значение по умолчанию
12 try:
13     n_val = float(n_in)
14 except (ValueError, TypeError):
15     pass
16
17 countTrub = 0 # Значение по умолчанию
18 try:
19     countTrub = int(countTrub_in)
20 except (ValueError, TypeError):
21     pass # Если countTrub не является числом, цикл не будет выполнен
22
23 i = 0
24 while i < countTrub:
25     V = 0.0 # Инициализируем V для каждой итерации, значение по умолчанию в случае ошибки
26     current_D = 0.0
27     current_I = 0.0
28     try:
29         current_D = float(D_in[i])
30         current_I = float(I_in[i])
31     except (ValueError, TypeError, IndexError):
32         outlist.append(V) # Добавляем 0.0 в outlist
33         i += 1
34         continue # Переходим к следующей итерации цикла
35     if current_D == 0:
36         V = 0.0
37     else:
38         R = current_D / 4.0
39         sqrt_n_safe = 0.0
40         if n_val >= 0:
41             sqrt_n_safe = math.sqrt(n_val)
42
43         sqrt_R_safe = 0.0
44         if R >= 0:
45             sqrt_R_safe = math.sqrt(R)
46         y_term_n_minus_0_1 = sqrt_n_safe - 0.1
47         y = sqrt_n_safe - 0.13 - 0.75 * sqrt_R_safe * y_term_n_minus_0_1
48         C = 0.0
49         if n_val != 0: # Предотвращаем деление на ноль
50             try:
51                 C = (1.0 / n_val) * math.pow(R, y)
52             except Exception: # Ловим общие ошибки math.pow (напр., если R отрицательное и y не целое, но здесь R всегда >=0)
53                 C = 0.0
54         product_R_I = R * current_I
55         if product_R_I >= 0:
56             V = C * math.sqrt(product_R_I)
57         else:
58             V = 0.0 # Если аргумент для корня отрицательный, V = 0.0
59     outlist.append(V) # 8. Исправлен отступ
60     i += 1 # Более идиоматичный способ инкрементировать счетчик
61 OUT = outlist
```

Рисунок 8. Скрипт Python «Вычисление скоростей» (рисунок авторов)

Цепочка узлов для задания диаметров и скорости в модели представлена на рисунке 9.

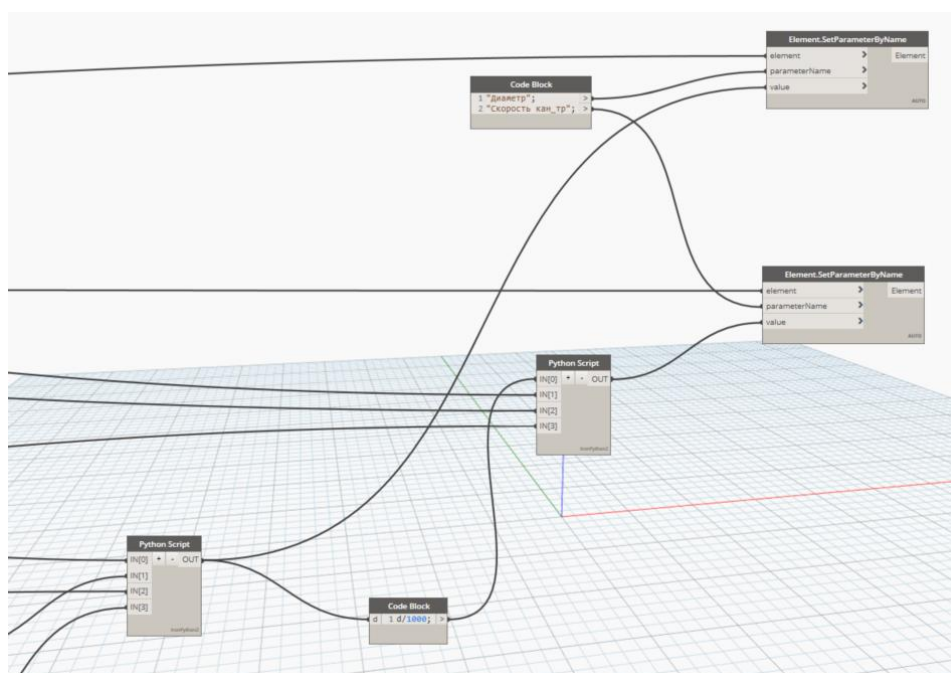


Рисунок 9. Цепочка узлов для задания диаметров и скорости в модели (рисунок авторов)

Запуск разработанного авторами скрипта производится с помощью соответствующей команды в интерфейсе Dynamo.

Тестирование алгоритма при базовых проектных значениях нагрузок показало, что расчетный диаметр основного стояка полностью совпал со значениями, полученными ранее в ходе ручного проектирования (рис. 10).

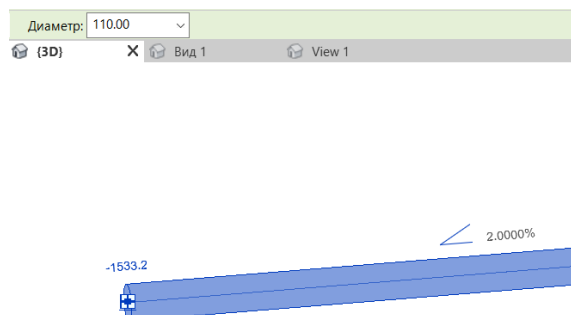


Рисунок 10. Результат расчета (рисунок авторов)

При этом на выходах из приборов с минимальной нагрузкой алгоритм автоматически занизил диаметры труб. С точки зрения строительных норм и государственных стандартов по канализации, данные участки (например, для подключения сантехнических приборов) должны строго сохранять нормативный диаметр. Автоматическое перестроение смежных фитингов (рис. 11) также спровоцировало пространственные коллизии в узлах соединительных деталей сетей: тройников, отводов (рис. 12).

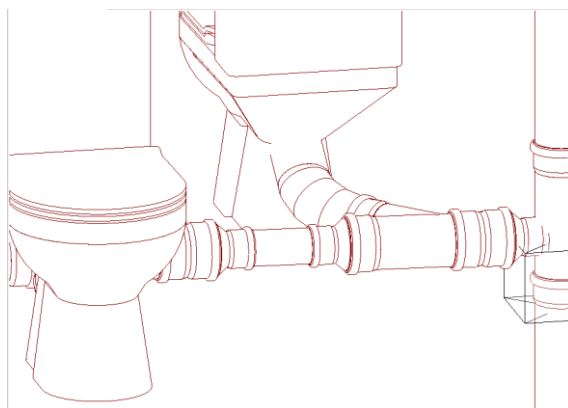


Рисунок 11. Участок трубопровода с малым количеством приборов (рисунок авторов)

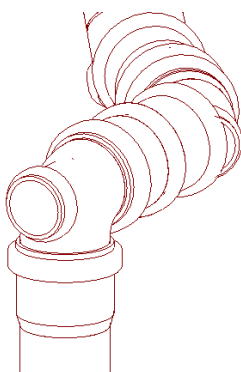


Рисунок 12. Коллизии (рисунок авторов)

При изменении параметров, связанных с увеличением расчетной численности пользователей и корректировкой нормативного наполнения, алгоритм мгновенно осуществил пересчет системы. Итоговый диаметр стояка на выходе закономерно увеличился в соответствии с возросшей нагрузкой, что доказало высокую гибкость автоматизированного метода управления моделью при изменении технического задания.

Алгоритм управления данными вложенных семейств инженерных систем

Разработанный авторами скрипт предназначен для интеграции данных и обеспечения сквозной совместимости и интероперабельности информации на стадии формирования спецификаций комплектов оборудования [10]. Он решает проблему «скрытых» вложенных компонентов элементов водоснабжения и канализации (смесителей, сифонов, пьедесталов раковин). Рассматриваемая цифровая модель системы водоснабжения и канализации представлена на рисунке 13.

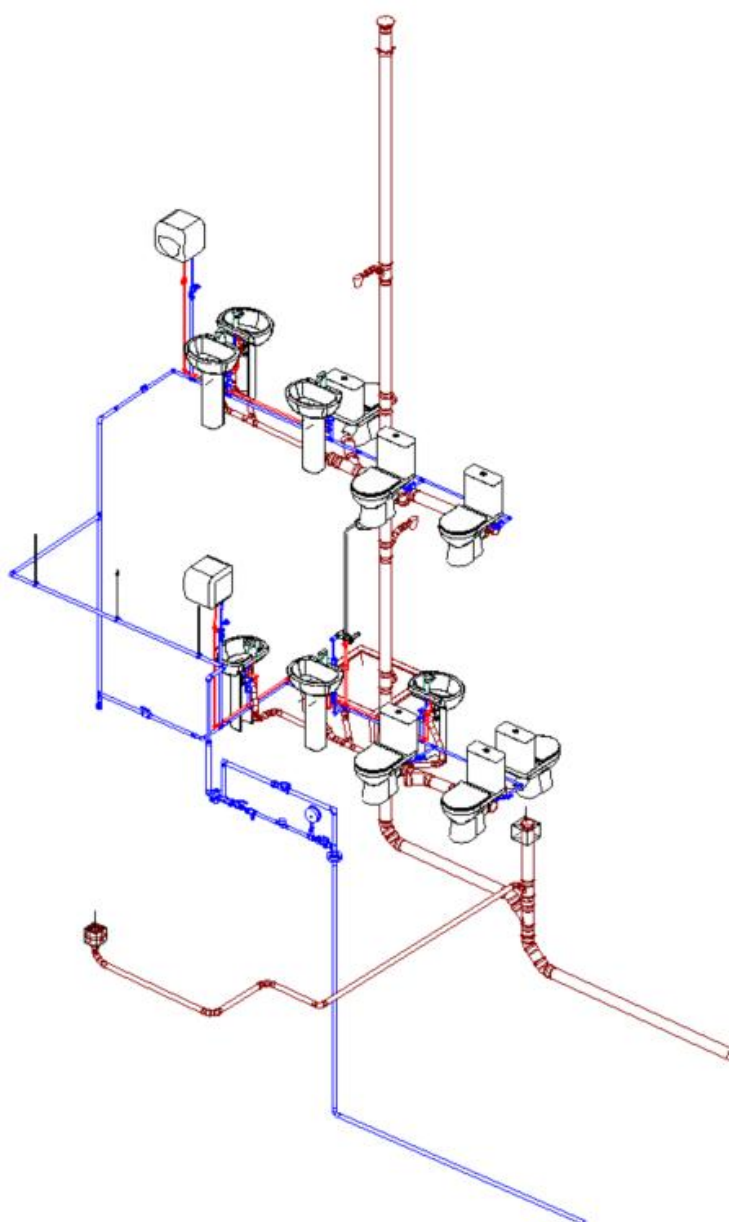


Рисунок 13. Рассматриваемая модель системы водоснабжения и канализации в Revit (рисунок авторов)

В стандартной среде проектирования Revit при создании спецификации по умолчанию отображаются данные только родительского семейства (рис. 14). При этом вложенные элементы (например, сифон или смеситель) остаются «невидимыми» для автоматического подсчета.

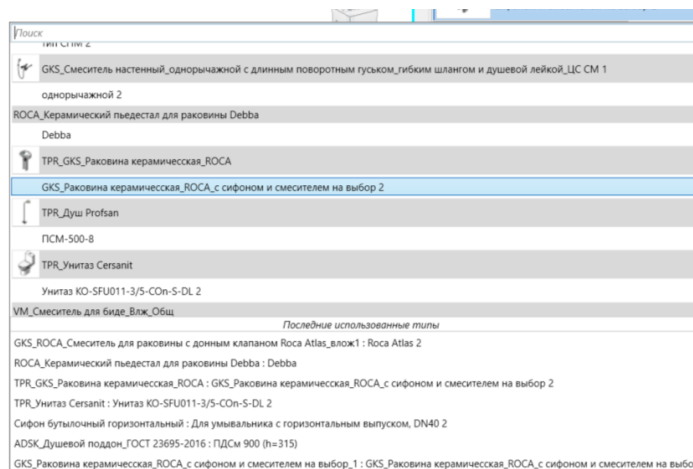
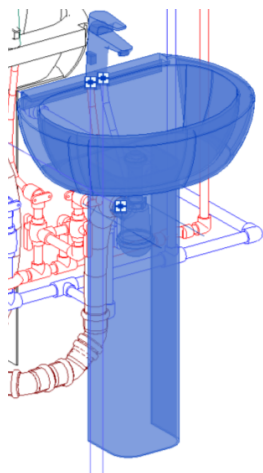


Рисунок 14. Наименование семейства в модели Revit (рисунок авторов)

Последовательность выполнения алгоритма:

1. Выборка объектов по категории «Сантехнические приборы».
2. Локализация зоны обработки: в свойствах элементов предварительно заполняется параметр «Комментарии» с указанием этажа и признака оборудования (рис. 15).

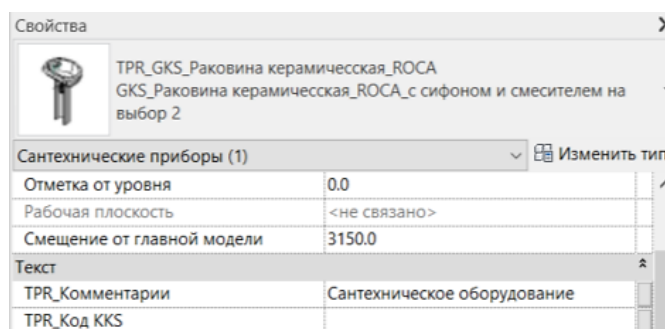


Рисунок 15. Заполнение параметра «Комментарии» (рисунок авторов)

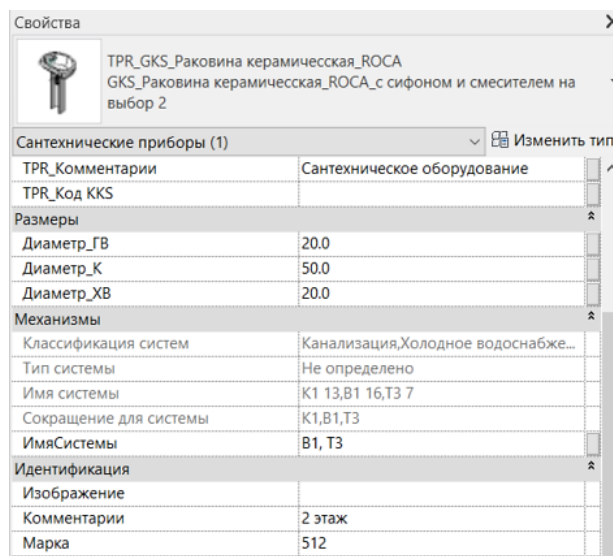


Рисунок 16. Заполненные параметры (рисунок авторов)

Заполненные параметры в свойствах элемента представлены на рисунке 16.

Пример изменяемых нодов представлен на рисунке 17.

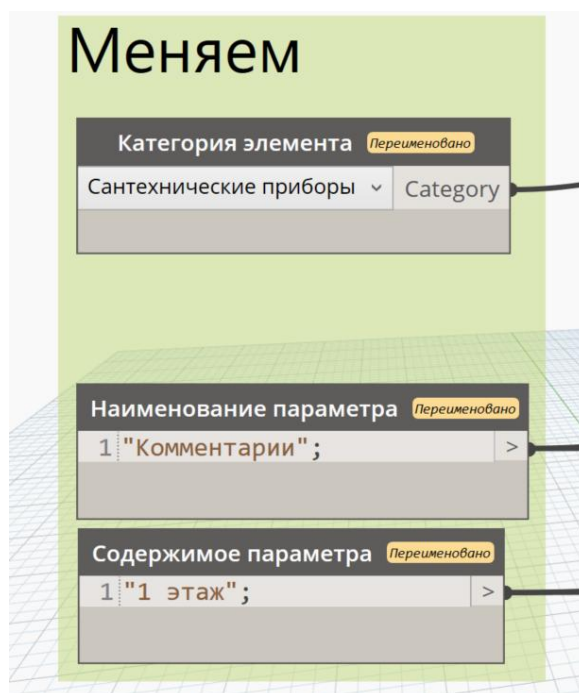


Рисунок 17. Изменяемые ноды (рисунок авторов)

Список сантехнических приборов в модели представлен на рисунке 18.

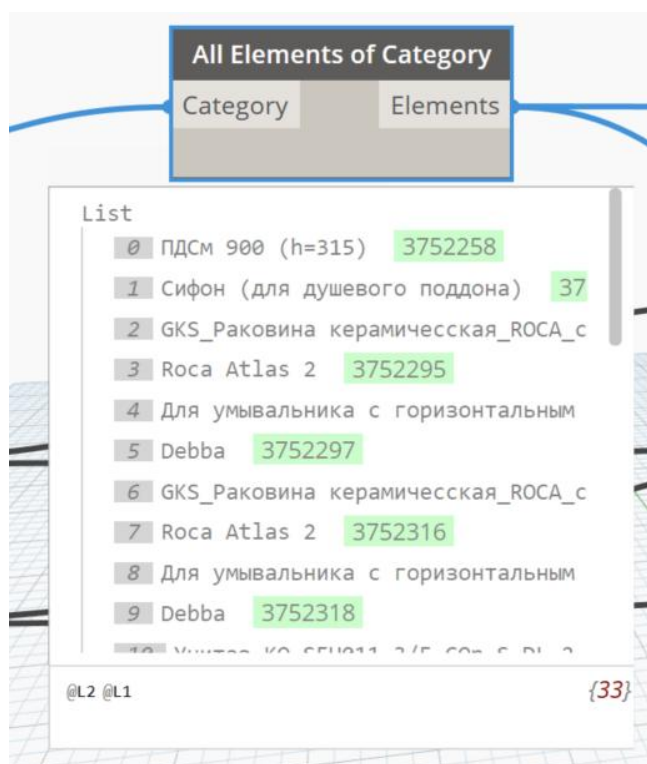


Рисунок 18. Список сантехнических приборов в модели (рисунок авторов)

3. Логическое сопоставление и фильтрация: на основе логических значений скрипт отсекает лишние объекты, формируя поток целевых родительских семейств для конкретной зоны здания (рис. 19).

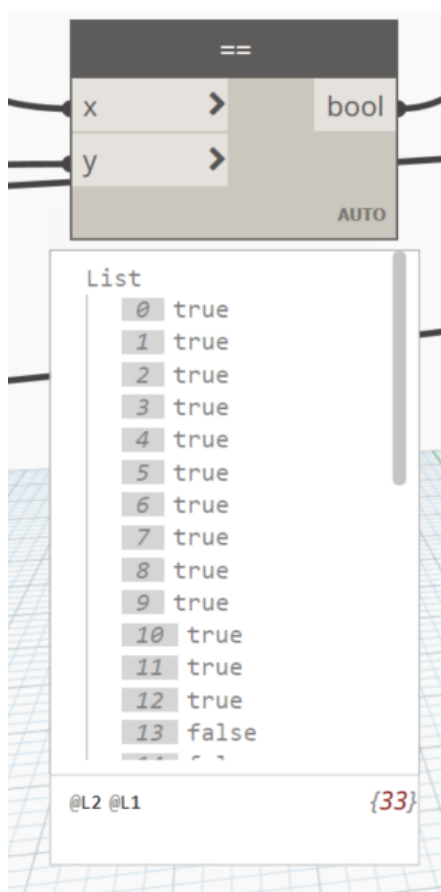


Рисунок 19. Список, составленный нодом (рисунок авторов)

Промежуточный результат скрипта представлен на рисунке 20.

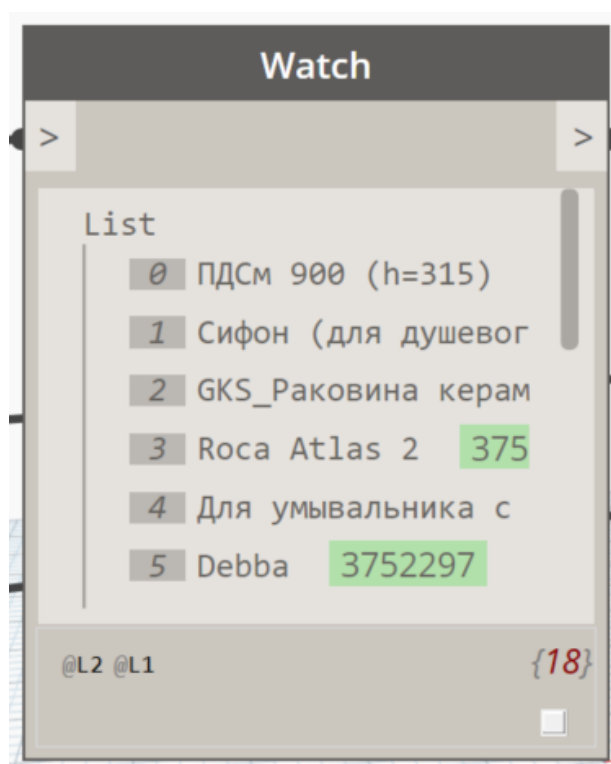


Рисунок 20. Промежуточный результат скрипта (рисунок авторов)

4. Извлечение структуры данных: специализированный нод осуществляет глубокий поиск внутри родительских элементов, извлекает спецификационные наименования всех вложенных узлов, нормализует данные и транслирует их в модель на основе общих принципов классификации строительной информации [11]. Это гарантирует актуальность информации для смежных участников жизненного цикла объекта (отделов смет, закупок и логистики). Финальный список элементов, а также готовый скрипт Dynamo представлен на рисунках 21, 22.

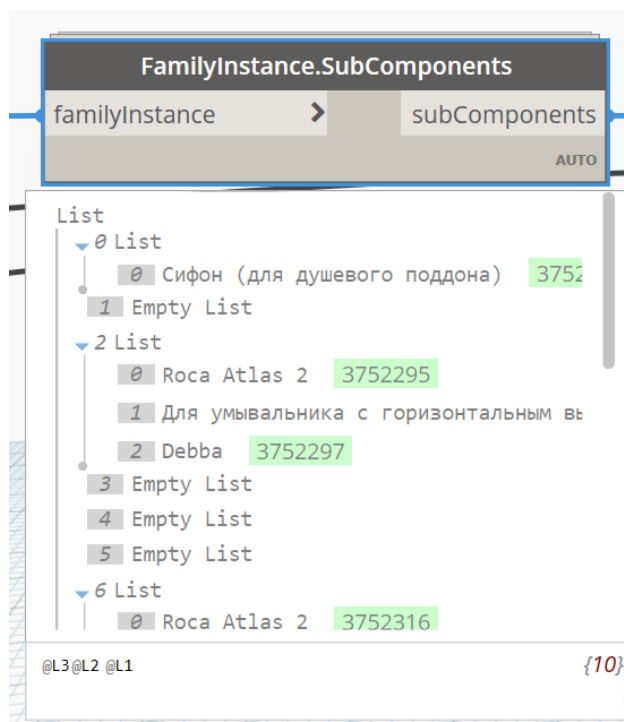


Рисунок 21. Финальный список элементов (рисунок авторов)

Готовый скрипт Dynamo представлен на рисунке 22.

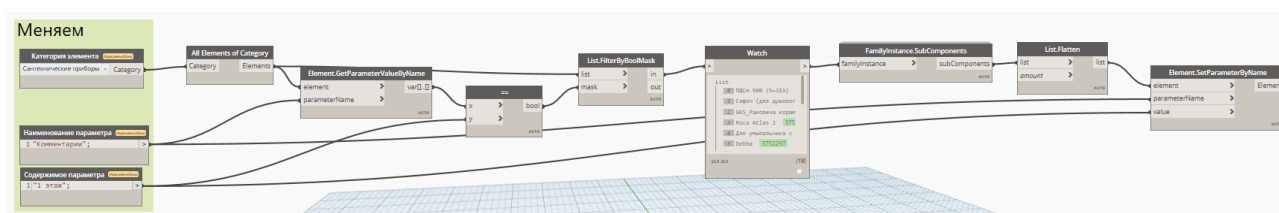


Рисунок 22. Готовый скрипт Dynamo (рисунок авторов)

В рамках апробации разработанного авторами алгоритма была выполнена проверка его работоспособности на выборке сантехнических приборов, размещенных в пространстве информационной модели.

Программный комплекс успешно идентифицировал целевую группу родительских компонентов и в автоматическом режиме извлек связанные с ними скрытые элементы инженерного оснащения (такие как смесители, сифонные узлы и пьедесталы). Вся собранная информация была корректно транслирована в результирующую ведомость оборудования (рис. 23).

Благодаря автоматизации процесса удалось полностью ликвидировать риск потери атрибутивных данных и сформировать сквозную спецификацию, где комплектующие отображаются наряду с основным оборудованием, что исключает возникновение ошибок из-за человеческого фактора.

Поз.	Наименование и техническая характеристика	Тип, марка, обозначение документа, опросного листа	Поставщик	Ед. измерения	Кол.	Масса 1ед., кг
Сантехническое оборудование						
85	Душевая система со смесителем		ПрофСан	шт.	1	0
86	Душевой поддон некий стальной эмалированный ПДСн 900			шт.	1	
87	Керамический пьедестал для раковины ROCA Debba	3379900Y	ROCA	шт.	6	10
88	Раковина Roca Debba	32799500Y	Roca	шт.	6	
89	Сифон для душевого поддона			шт.	1	
90	Сифон для раковины AlcaPlast A41Flex	A41+A710	AlcaPlast	шт.	6	
91	Смеситель для раковины с боковым клапаном Roca Atlas	5A3090COM	ROCA	шт.	6	
92	Унитаз STREET FUSION NEW CLEAN ON 011 3/5	S-K0-SFU011-3/5-C0n-S-DL-w	Cersanit	шт.	6	38.38

Рисунок 23. Результат работы скрипта Dynamo (фото авторов)

Интеграция разработанных авторами алгоритмов в практику проектирования позволяет оптимизировать информационные процессы жизненного цикла объектов капитального строительства, что можно наблюдать в таблицах 1 и 2.

Таблица 1

Сравнение эффективности автоматизированного расчета и ручного метода

Критерий	Метод Dynamo	Метод «Вручную»	Роль в управление жизненным циклом
Расчет по нормам (СП)	Преимущество	Недостаток	Автоматическое выполнение, существенная экономия времени
Контроль коллизий фитингов	Недостаток	Преимущество	Требует ручной доработки фитингов после изменения диаметров
Вариативность и гибкость	Преимущество	Недостаток	Возможность оперативного пересчета модели при изменении исходных данных
Масштабируемость кода	Преимущество	Недостаток	Однократное создание алгоритма обеспечивает его тиражирование на новые проекты

Разработано авторами

Таблица 2

Сравнение методов управления данными вложенных семейств

Критерий	Метод Dynamo	Метод «Вручную»	Роль в управление жизненным циклом
Скорость обработки	Преимущество	Недостаток	1,5 ч (авторасчет по скрипту)
Человеческий фактор	Преимущество	Недостаток	0,2 ч (автоматический запуск нодов)
Глубина извлечения данных	Преимущество	Недостаток	0,1 ч (двусторонний экспорт/импорт данных)
Актуальность информации	Преимущество	Недостаток	0,05 ч (автогенерация заданий на отверстия)

Разработано авторами

Анализ аналогичных решений автоматизации инженерных систем и параметрических методов проектирования, доступных в открытых источниках, подтверждает общую тенденцию эффективности визуального программирования [12; 13]. К ним относятся специализированные плагины и скрипты автоматической расстановки зданий на отверстия при пересечении труб со строительными конструкциями, автоматического подсчета длины изоляции фитингов (запись в параметр ADSK_Количество), нумерации стояков (параметр ADSK_Номер стояка), а также расстановки хомутов [14].

Особое значение приобретает автоматизация процессов обнаружения пространственных пересечений коммуникаций. Как показывает опыт трансляции и координации сложных моделей, своевременное выявление межраздельных наложений является обязательным условием обеспечения интероперабельности проектных данных [15]. Внедрение подобных алгоритмов на этапе проектирования инженерных сетей критически важно для управления жизненным циклом объекта капитального строительства. Однако это накладывает жесткие требования к

качеству корпоративных шаблонов, версии интерпретатора Python, квалификации специалистов компании, а также подразумевает надлежащие процедуры контроля [16].

Выводы

Проведенное авторами исследование доказывает, что автоматизация управления данными на стадии проектирования инженерных сетей является ключевым инструментом повышения эффективности управления жизненным циклом объектов строительства. На основе полученных результатов сформулированы следующие выводы.

Алгоритм гидравлического расчета наиболее эффективен в качестве аналитической поддержки принятия решений на масштабных проектах со сложной топологией сетей. Его рекомендуется применять для автоматического определения расчетных параметров магистралей и стояков, тогда как непосредственную замену диаметров на сложных концевых участках следует производить вручную во избежание нормативных ошибок и пространственных коллизий фитингов.

Внедрение алгоритма автоматического добавления параметров из родительских семейств во вложенные полностью исключает разрыв информационных связей между элементами инженерных систем и их комплектующими. Это гарантирует точность сметных расчетов и прозрачность данных для последующих стадий материально-технического снабжения и строительства.

Главными барьерами применения среды визуального программирования Dynamo на этапе проектирования коммуникаций остаются зависимость кода от версий программного обеспечения и высокий порог входа для специалистов. Тем не менее, использование подобных алгоритмов обеспечивает стандартизацию проектных процессов, минимизирует влияние человеческого фактора и гарантирует высокую точность и валидацию данных цифровых моделей инженерных систем объектов капитального строительства.

ЛИТЕРАТУРА

1. Гинзбург А. В. BIM-технологии на протяжении жизненного цикла строительного объекта // Информационные ресурсы России. — 2016. — № 5. — С. 28–31. — EDN WTQRRH.
2. Галишникова В. В., Бочаров М. Е. Информационные модели объектов строительства в течении их жизненного цикла // Недвижимость: экономика, управление. — 2024. — № 2. — С. 65–72. — EDN UMRISY.
3. Шинкарева Г. Н. Модель инжиниринговой схемы организации строительства для контрактов жизненного цикла // Вестник МГСУ. — 2018. — Т. 13. — № 10. — С. 1204–1210. — EDN VMCLUB.
4. Четверик Н. П. Поэтапное внедрение технологий информационного моделирования (BIM) в строительной сфере // Строительные материалы, оборудование, технологии XXI века. — 2014. — № 12(191). — С. 44–47. — EDN TFCASZ.
5. Бузало Н. А., Царитова Н. Г., Платонова И. Д., Лысенко В. Н. Параметрическое моделирование конструкций в среде визуального программирования // Вестник евразийской науки. — 2021. — Т. 13, № 4. — EDN PSJZHP.
6. Клековкин Е. А., Сунцов А. С. Применение визуального программирования для задач автоматизации в строительстве // Construction and Geotechnics. — 2023. — Т. 14. — № 2. — С. 128–143. — EDN SRAXWK.

7. Смакаев Р. М., Низина Т. А. Применение среды визуального программирования Dynamo при разработке проекта здания в Autodesk Revit // Основы экономики, управления и права. — 2020. — № 2(21). — DOI 10.51608/23058641_2020_2_48. — EDN VTDDXI.
8. Талапов В. В. Основы BIM: введение в информационное моделирование зданий. — М.: ДМК Пресс, 2011. — 392 с. — EDN YPVPZT.
9. Борисов М. П., Вавин А. А., Уткина В. Н. Современные автоматизированные системы Revit и Renga для информационного моделирования зданий // Огарев-Online. — 2020. — № 3(140). — EDN VHGAIQ.
10. Гинзбург А. В., Куликова Е. Н., Павлов А. С., Вайнштейн М. С. Обеспечение интероперабельности при проектировании с применением технологий информационного моделирования // Вестник евразийской науки. — 2019. — Т. 11. — № 6. — EDN XMCKSE.
11. Волков А. А., Челышков П. Д., Давыдов А. Е., Никитенко А. А. Разработка принципов алгоритмов применения классификатора строительной информации // Вестник НИЦ «Строительство». — 2025. — № 45(2). — С. 154–166. — DOI 10.37538/2224-9494-2025-2(45)-154-166. — EDN ACXFTJ.
12. Федчун Д. О. Сравнительный анализ методов генеративного, параметрического и информационного архитектурного проектирования // Научный журнал строительства и архитектуры. — 2018. — № 2(50). — С. 103–114. — EDN XQGYJF.
13. Георгиев Н. Г., Шумилов К. А., Семенов А. А. Визуальное программирование в задачах моделирования строительных конструкций // Инженерно — строительный вестник Прикаспия / ИГТУ имени М. Т. Калашникова (Ижевск). — 2021. — № 4(38) — С. 117–123. — DOI 10.52684/2312-3702-2021-38-4-117-123. — EDN ZPSGCJ.
14. Анисимова Н. В. Обзор основных плагинов средового моделирования и оптимизации геометрии в Dynamo и Grasshopper // BIM-моделирование в задачах строительства и архитектуры: материалы II Международной научно-практической конференции. — Санкт-Петербург, 2019. — С. 228–233. — DOI 10.23968/BIMAC.2019.042. — EDN KGJVHK.
15. Вареник К. А., Вареник А. С., Храмов Д. Д., Чамеев А. С., Николаев Д. И. Применение технологий информационного моделирования в изучении и трансляции объектов архитектурного наследия // Вестник евразийской науки. — 2024. — Т. 16. — № 2. — EDN TSAXRN.
16. Меркишкина К. В., Семенихина Ю. Е. Применение платформы визуального программирования Dynamo при проектировании систем отопления жилых зданий в Алтайском крае // Наука и молодежь: материалы XVIII всероссийской научно-технической конференции студентов, аспирантов и молодых ученых, Барнаул, 19–23 апреля 2021 года. Том ТОМ 1, часть 2. — Барнаул: Алтайский государственный технический университет имени И. И. Ползунова, 2021. — С. 97–100. — EDN TVDRJB.

Varenik Kirill Aleksandrovich

The Yaroslav-the-Wise Novgorod State University, Veliky Novgorod, Russia

E-mail: vkirillv89@mail.ru

ORCID: <https://orcid.org/0009-0004-0604-0465>

RSCI: https://elibrary.ru/author_profile.asp?id=721765

Kovaleva Ekaterina Mikhailovna

The Yaroslav-the-Wise Novgorod State University, Veliky Novgorod, Russia

E-mail: c.kowaliowa2014@yandex.ru

Koshavkin Stanislav

LLC «Lenmontage», Saint Petersburg, Russia

E-mail: stanislavkoshavkin@gmail.com

Chameev Aleksandr Sergeevich

The Yaroslav-the-Wise Novgorod State University, Veliky Novgorod, Russia

E-mail: s244313@std.novsu.ru

RSCI: https://elibrary.ru/author_profile.asp?id=1239941

Kokin Vladimir Ilich

The Yaroslav-the-Wise Novgorod State University, Veliky Novgorod, Russia

E-mail: s253019@std.novsu.ru

Automation of data management in information models of capital construction objects at the engineering networks design stage using the Dynamo visual programming environment

Abstract. The article discusses authorial methods and algorithms for automating data management within digital information models of capital construction objects at the engineering networks design stage. The study is due to the need to minimize design errors and optimize routine processes during the development of water supply and sewage networks. The authors describe two developed scripts in Dynamo, which incorporate source code in Python. The first algorithm automates hydraulic calculations and sizing of sewage pipelines in compliance with the SP 30.13330.2020 code of practice. The second script ensures end-to-end automatic extraction of the names of nested components within engineering systems from parent families into the equipment schedule. The paper provides the results of verifying these algorithms using a real building information model. During testing, the authors investigate spatial clashes occurring in pipeline fitting nodes due to automated geometry changes. Based on the conducted research, a comparative analysis of the efficiency of implementing visual programming tools is presented. The authors evaluate this approach in the context of optimizing information processes, improving budget estimation, and reducing human-factor risks during early project phases. The study proves that custom visual programming algorithms facilitate the standardization of design workflows, minimize errors, and ensure reliable validation of data.

Keywords: lifecycle management; information modeling; engineering networks; design; hydraulic calculation; visual programming; script