

ИНТЕРПРЕТАТОР ДЛЯ ГЕНЕРАЦИИ ТИПОВЫХ СХЕМ АВТОМАТИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ НА БАЗЕ ПРОГРАММИРУЕМОГО SVG



А. Р. Гета, студент группы 3-го курса
E-mail: artyom.geta@mail.ru
ФГБОУ ВО «Калининградский государственный
технический университет»

А. Л. Онучин, ст. преподаватель
E-mail: aleksandr.onuchin@klgtu.ru
ФГБОУ ВО «Калининградский государственный
технический университет»

Н. А. Долгий, канд. техн. наук, доцент
E-mail: dolgi@klgtu.ru
ФГБОУ ВО «Калининградский государственный
технический университет»

В данной статье предложен способ генерации типовых схем автоматизации с использованием языка разметки масштабируемой векторной графики SVG. На примере разработки проекта автоматизации приточной вентиляции технологического объекта показаны этапы создания типовых шаблонов с использованием интерпретатора программируемого SVG. Приведены преимущества использования интерпретатора программируемого SVG, использование которого сократит время, необходимое для выполнения различных этапов проектирования и снизит трудозатраты проектировщика.

Ключевые слова: шаблон, интерпретатор программируемого SVG, типовая схема, приточная вентиляция, параметр, генератор.

ВВЕДЕНИЕ

Последние десятилетия наблюдается рост номенклатуры технических средств автоматизации как на отечественном, так и на зарубежном рынках [1]. При разработке проекта нового технического решения в области автоматизации технологических процессов и производств необходимо разработать функциональную схему автоматизации и техническую документацию к ней. Применяемые в настоящее время для проектирования схем автоматизации программные пакеты САПР имеют ряд недостатков, связанных с использованием определенных параметров и правил, заданных программным обеспечением, что снижает гибкость и уникальность проекта и часто делает этот процесс не эффективным с точки зрения проектировщика [2]. С целью минимизации затрат труда проектировщиков, упрощения процедуры проектирования схем автоматизации предложен способ генерации типовых схем автоматизации с помощью разработанного программного обеспечения, реализованного с использованием языка разметки масштабируемой векторной графики SVG [3, 4].

ОБЪЕКТ ИССЛЕДОВАНИЯ

Основными элементами проектируемой функциональной схемы автоматизации технологического процесса являются технологические аппараты и оборудование, технические средства автоматизации и управления. В качестве объекта исследования рассмотрим проектирование схемы автоматизации приточной вентиляции технологического объекта управле-

ния, в состав которой входят задвижка с электроприводом, регулирующая расход воздуха, и два вентилятора. Система приточной вентиляции является типовой схемой автоматизации, но несмотря на комплектную поставку, от проектировщика часто требуется дополнительно внести в проект автоматизации кабельную продукцию и различные внешние устройства. Такие дополнения влекут за собой необходимость обоснования принятых решений при проектировании схем автоматизации. При этом процесс разработки проектной документации можно упростить, используя специализированный интерпретатор, позволяющий генерировать типовые схемы автоматизации для конкретного технологического процесса.

ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

Цель работы заключается в разработке специальной программы, которая позволит проектировщику упростить выполнение функциональных схем автоматизации, используя язык разметки масштабируемой векторной графики SVG. Она представляет собой интерпретатор программируемого SVG, который поддерживает передачу параметров (переменных), работу с переменными и функции, требуемые для автоматизации проектирования схем систем управления технологическими процессами и объектами.

Задачей работы является разработка интерпретатора программируемого SVG, который позволит генерировать типовые схемы систем автоматизации, что в итоге сократит время, необходимое для выполнения различных этапов проектирования, и снизит трудозатраты проектировщика.

МЕТОДЫ ИССЛЕДОВАНИЯ

Разработанный интерпретатор использует масштабируемую векторную графику SVG для генерации типовых схем систем автоматизации и составления технической документации к ним. SVG формат является открытым стандартом, который использует информацию в виде математических описаний, что позволяет масштабировать изображения схем без потери качества. SVG-файлы обычно имеют меньший размер по сравнению с растровыми изображениями. Кроме того, формат SVG поддерживает использование градиентов и фильтров, что позволяет создавать более сложные графические элементы. Внешний вид кода SVG напоминает код HTML-документа, снабженный описанием стилевых классов и скриптами на языке JavaScript. Изображения формата SVG по языку представления совместимы как с форматом HTML, так и с форматом CSS. Это означает, что редактирование объектов формата SVG доступно через код [5].

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

Задачей автоматизации процесса проектирования является разработка интерпретатора, который позволит генерировать типовые схемы систем автоматизации с минимальными затратами человеческого труда. Для решения указанной задачи можно написать простой скрипт, который в зависимости от значений параметров реализует различные схемы, но это было неудовлетворительно: для каждой новой схемы был бы необходим отдельный скрипт с подключением всех библиотек; более того, такой код был бы трудно читаемым. В связи с этим возникает необходимость разработать универсальный простой инструмент, создание схем для которого было бы удобно специалисту. За основу проекта был взят формат масштабируемой векторной графики SVG. Таким образом, необходимо разработать интерпретатор для программируемого SVG, который поддерживает передачу параметров, работу с переменными и функции для автоматизации проектирования схем систем управления технологическими процессами и объектами. На рисунке 1 приведена блок-схема генератора, поясняющая этапы проектирования схемы автоматизации приточной вентиляции.

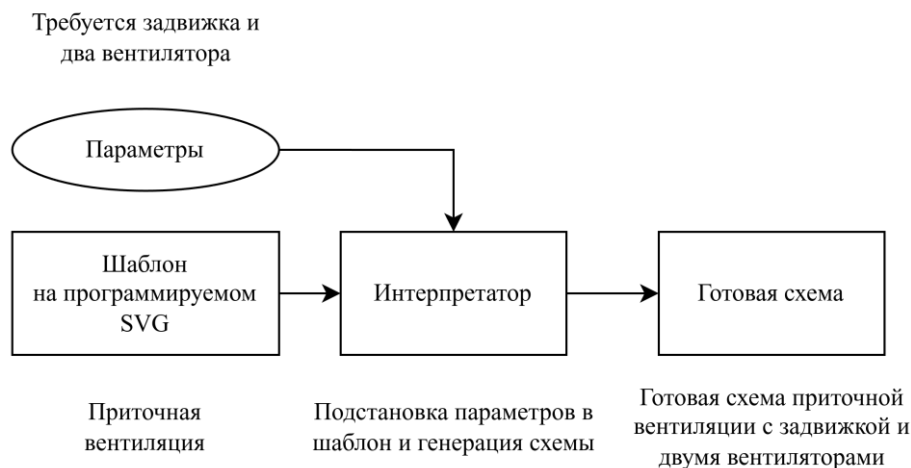


Рисунок 1 – Блок-схема генератора

Для начала работы интерпретатору, предварительно выбрав шаблон исполнения (готовая схема), необходимо передать параметры: наличие задвижки, датчиков, количество вентиляторов и т. д. Список параметров шаблон формируется следующим образом (рисунок 2):

```

<params>
  <v id="valve" name="Задвижка" type="b" value="0" />
  <v id="TEouter" name="Датчик температуры наружный" type="b"
    value="0" />
</params>
  
```

Рисунок 2 – Задача списка параметров шаблона

После того как пользователь ввел параметры, их значения необходимо обработать. Обработка выполняется при помощи специальных блоков if, elif, else (рисунок 3).

```

<if ex="valve=1">
  <g transform="translate({XL+1.5},{YL})">
    <circle cx="0" cy="0" r="1.5" fill="black" />
    <line x1="-2.5" y2="-2.5" x2="2.5" y1="2.5" stroke-width="0.6" />
  </g>
  <polygon points="0,1.5 -0.5,2.5 0.5,2.5" fill="black" />
  <v name="XL" value="XL + 30" />
</if>
  
```

Рисунок 3 – Выражение условия if

Все, что находится внутри тэга if (линии, фигуры, текст, программируемые тэги), будет выведено (или обработано) в конечную схему при истинности логического выражения в атрибуте ex. Также внутри тэга if можно поместить тэги elif и else (иначе если и иначе соответственно), которые будут обработаны в случае невыполнения логического выражения if в соответствии с рисунком **Ошибка! Источник ссылки не найден..**

```

<if ex="fan>1">
  ...
  <elif ex="fan=1">
    ...
  </elif>
  <else>
    ...
  </else>
</if>
  
```

Рисунок 4 – Выражение условий «иначе если» elif и «иначе» else

Наравне с параметрами в шаблоне можно создать внутренние переменные, которые не будут выведены пользователю, однако обладают тем же самым функционалом. На рисунке **Ошибка! Источник ссылки не найден.** происходит переопределение существующей переменной PipeWidth и присвоение ей значения на 30 больше предыдущего.

```
<v id="PipeWidth" value="=PipeWidth + 30" />
```

Рисунок 5 – Переопределение переменной

Создание переменной происходит таким же образом (без ссылки на нее в атрибуте value). В примере выше также показано, как можно использовать переменные в атрибутах. Формула должна начинаться со знака «=», чтобы сообщить интерпретатору о том, что это формула и ее необходимо вычислить. На рисунке **Ошибка! Источник ссылки не найден.** показан случай, когда не весь атрибут должен быть формулой, а лишь его часть. Эту часть (части) нужно взять в фигурные скобки.

```
<g transform="translate({PipeWidth+1.5},{PipeY})">...</g>
```

Рисунок 6 – Атрибут с вычисляемыми частями

Для упрощения изображений нескольких копий объектов на схеме была добавлена возможность создания цикла for (рисунок 7):

```
<for var="i" range="=van">
...
</for>
```

Рисунок 7 – Цикл for

Все, что находится внутри цикла, будет обработано и выведено в конечную схему некоторое количество раз. Тело цикла определяется значением (математическим выражением) в атрибуте range. Для использования номера текущей итерации внутри тела цикла существует переменная-счетчик с названием, взятым из атрибута var.

При помощи тэга device в схеме автоматизации можно отобразить новое устройство (прибор) в соответствии ГОСТ 21.208-2013 (рисунок **Ошибка! Источник ссылки не найден.**). Все атрибуты, начинающиеся с «a-» нужны для автонумерации и генерации.

```
<device x="0" y="=fan*(-17.5)-35"
t="PDS" a-group="PDS"
a-spec="Дифференциальное реле
%%давления"
a-from="ШУ1"
a-cable="МКЭШнг (А) -LSLTx 4x0,5" />
```

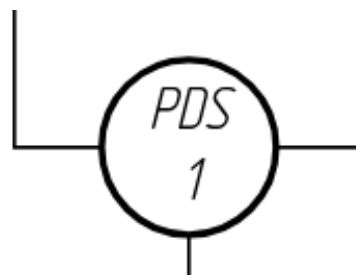


Рисунок 8 – Устройство (прибор)

Атрибут a-group позволит сгруппировать несколько устройств в одну позицию в угловой спецификации; a-spec – подпись в ней же и кабельном журнале; a-from – графа «откуда» в кабельном журнале; a-cable – графа информации о кабеле.

Если существует необходимость вставить просто номер без обозначения прибора, можно использовать модифицированный тэг text. В нем использовать атрибут a-type, чтобы определить группу, внутри которой необходимо нумеровать позиции, при этом существуют стандартные группы device и signal (рисунок 9).

```

<text x="4" y="12" align="c"
  a-type="device"
  a-group="vents"
  a-сpec="Привод вентилятора%"
  a-from="Шкаф управления"
  a-cable="МКЭШнг (А) -LSLTx
    4x0,5">
  ?
</text>

```

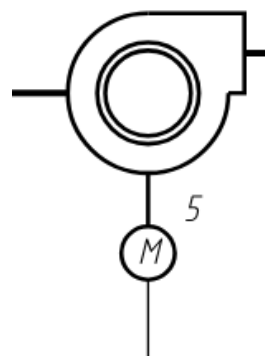


Рисунок 9 – Номер без устройства

Все номера будут проставлены в приоритете по горизонтали, затем по вертикали, не зависимо от того, в каком порядке они записаны в шаблоне. Угловую спецификацию можно вставить в чертеж тэгом corner-spec (рисунок 10).

Поз. обознач.	Наименование	Кол.	Примечание
1, 3	Дифференциальное реле давления	2	
2, 6	Датчик температуры	2	
4, 5	Привод вентилятора	2	
7, 8	Частотный преобразователь	2	

Рисунок 10 – Угловая спецификация

Кабельный журнал формируется из списка устройств. Его можно сгенерировать тэгом cables (рисунки 11 и 12). Также дополнительно в спецификацию можно вставить примечания к схеме.

Номер кабеля, жгута, трубы	Направление		Направление по чертежам расположения
	откуда	куда	
П11-1	Шкаф управления ШУ-П11	Дифференциальное реле давления П11-1	
П11-2	Шкаф управления ШУ-П12	Датчик температуры П11-2	
П11-3	Шкаф управления ШУ-П13	Дифференциальное реле давления П11-3	
П11-4	Шкаф управления ШУ-П14	Привод вентилятора П11-4	
П11-5	Шкаф управления ШУ-П15	Привод вентилятора П11-5	
П11-6	Шкаф управления ШУ-П16	Датчик температуры канальный П11-6	
П11-7	Шкаф управления ШУ-П17	Частотный преобразователь П11-7	
П11-8	Шкаф управления ШУ-П18	Частотный преобразователь П11-8	

Рисунок 11 – Кабельный журнал (левая часть)

Кабель, провод			Труба		Измерительная цепь	Чертеж установки
Марка, число жил, сечение	Длина, м		Марка диаметр	Длина, м		
	проектируемая	фактическая				
МКЭШнг(А)-LS Tx 4x0,5	8		ПВХ гибкая гофр. Ø 16 мм	4		
МКЭШнг(А)-LS Tx 4x0,5	8		ПВХ гибкая гофр. Ø 16 мм	4		
МКЭШнг(А)-LS Tx 4x0,5	8		ПВХ гибкая гофр. Ø 16 мм	4		
МКЭШнг(А)-LS Tx 4x0,5	8		ПВХ гибкая гофр. Ø 16 мм	4		
МКЭШнг(А)-LS Tx 4x0,5	8		ПВХ гибкая гофр. Ø 16 мм	4		
МКЭШнг(А)-LS Tx 4x0,5	8		ПВХ гибкая гофр. Ø 16 мм	4		
МКЭШнг(А)-LS Tx 4x0,5	8		ПВХ гибкая гофр. Ø 16 мм	4		
МКЭШнг(А)-LS Tx 4x0,5	8		ПВХ гибкая гофр. Ø 16 мм	4		

Рисунок 12 – Кабельный журнал (правая часть)

Символ «;» является в данном случае разделителем в списке. Также в текстах и атрибутах можно ставить «%%» в качестве разделителя строк.

Генерация штампа также автоматизирована и выполняется тэгом stamp (рисунок 13). Он заполняется параметрами, переданными пользователю по умолчанию.

Примечания						
1 Шкаф управления и датчики поставляются комплектно с вентиляционной установкой						
2 Условные обозначения приборов и средств автоматизации соответствуют ГОСТ 21.208-2013						
						01.03.2024-TX2
						Адрес
Изм.	Кол. уч.	Лист	№ док.	Подпись	Дата	
Разраб.		Иванов И.				Стадия
Проверил		Петров П.				Лист
Норм. контр.		Степанов С.				Листов
ГИП		Тимофеев Т.				Р
						Приточная система П1
						Схема автоматизации
						Волна Автоматизации

Рисунок 13 – Штамп с примечаниями

ЗАКЛЮЧЕНИЕ

В статье предложен интерпретатор для расширенного программируемого SVG, который поддерживает передачу параметров проекта, обработку переменных и функций, предназначенных для проектирования схем автоматизации технологических процессов и объектов. Разработанная методика построения типовой схемы автоматизации приточной вентиляции технологического объекта, и ее реализация опробованы и протестированы в рамках одной задачи. Предложенную методику можно использовать для построения web-сервиса генератора типовых схем автоматизации. В дальнейшем, используя комплексный подход к построению законченных проектных решений, генератор типовых схем автоматизации планируется интегрировать в систему построения цифровой модели и цифрового двойника объекта проектирования.

СПИСОК ЛИТЕРАТУРЫ

1. Филатова, Н. Н. Генерация решающих правил для проектирования функциональных схем систем автоматизации / Н. Н. Филатова, А. Г. Требухин // Известия ЮФУ. – Технические науки. – 2012. – № 11(136). – С. 206–211. – EDN PIVPNR.
2. Садыков, А. И. Система автоматического проектирования (САПР) – перспективы и недостатки внедрения системы в производство / А. И. Садыков, А. М. Варламова // Технические и математические науки. Студенческий научный форум: Сборник статей по материалам XXVII студенческой международной научно-практической конференции (Москва, 13 апреля 2020 г.). – Т. 4(27). – Москва: Общество с ограниченной ответственностью «Международный центр науки и образования», 2020. – С. 29–34. – EDN DIGPYV.
3. SVG-разметка двумерной графики: Опыт использования SVG в создании двумерной графики / А. И. Телегин, Д. Н. Тимофеев, Д. И. Читалов, С. Г. Пудовкина. – Миасс: Южно-Уральский государственный университет (Национальный исследовательский университет), 2015. – 73 с. – EDN UVMNDP.
4. Иванов, А. Д. Использование SVG для построения 2D-графики / А. Д. Иванов // Студенческий научный форум 2024: Сборник статей X Международной научно-практической конференции в 2 частях (Пенза, 17 января 2024 г.). – Пенза: Наука и Просвещение (ИП Гуляев Г. Ю.), 2024. – С. 96–101. – EDN FKSWAU.
5. Прилищ, А. С. Изучение графической технологии SVG старшеклассниками / А. С. Прилищ // Инновационные подходы к обучению физике, математике, информатике: Материалы Международной студенческой научно-практической конференции (Минск, 18 апреля 2023 г.). – Минск: Белорусский государственный педагогический университет им. М. Танка, 2023. – С. 379–384. – EDN SCBTVO.

INTERPRETER FOR GENERATION OF TYPICAL SCHEMES OF AUTOMATION OF TECHNOLOGICAL PROCESSES BASED ON PROGRAMMABLE SVG

A. R. Geta, 3th year student
E-mail: rtyom.geta@mail.ru
Kaliningrad State Technical University

A. L. Onuchin1, senior lecturer
E-mail: aleksandr.onuchin@klgtu.ru
Kaliningrad State Technical University

N.A. Dolgii, PhD in Engineering
E-mail: dolgi@klgtu.ru
Kaliningrad State Technical University

This article proposes a method for generating typical automation schemes using the Scalable Vector Graphics (SVG) markup language. Using the example of developing a project for automating the supply ventilation of a technological facility, the stages of creating typical templates using a programmable SVG interpreter are shown. The advantages of using a programmable SVG interpreter are presented. Their use will reduce the time required to complete various design stages and reduce the designer's labor costs.

Keywords: *template, programmable SVG interpreter, typical diagram, supply ventilation, parameter, generator.*