

*На правах рукописи*

**ПАРАМОНОВ Андрей Владимирович**

**ПРОЕКТИРОВАНИЕ ПРИКЛАДНОГО ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ  
КОНТРОЛЛЕРОВ НА ОСНОВЕ ВИЗУАЛЬНЫХ МОДЕЛЕЙ**

Специальность:

2.3.5 – Математическое и программное обеспечение  
вычислительных систем, комплексов и компьютерных сетей

**АВТОРЕФЕРАТ**

диссертации на соискание учёной степени  
кандидата технических наук

Уфа – 2024

Работа выполнена в ФГБОУ ВО «Оренбургский государственный университет» на кафедре управления и информатики в технических системах

Научный руководитель: доктор технических наук, профессор  
**Боровский Александр Сергеевич**  
заведующий кафедрой управления и информатики в технических системах ФГБОУ ВО «Оренбургский государственный университет»

Официальные оппоненты:

Ведущая организация:

Защита диссертации состоится                      в            на заседании диссертационного совета 24.2.479.07 на базе ФГБОУ ВО «Уфимский университет науки и технологий» по адресу: 450008, г. Уфа, ул. Карла Маркса, д. 12

С диссертацией можно ознакомиться в библиотеке ФГБОУ ВО «Уфимский университет науки и технологий» и на сайте <https://uust.ru/>.

Автореферат разослан                     

Учёный секретарь  
диссертационного совета  
доктор технических наук

Вульфин Алексей Михайлович

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы.** Современные предприятия характеризуются растущей потребностью в разработке, внедрении и сопровождении программного обеспечения (ПО) на различных уровнях, которая, в частности, объясняется стремлением к цифровизации производства и концепции «Индустрия 4.0». На уровне технологического управления это стремление оказывает влияние на рост использования программируемых устройств – программируемых логических контроллеров (ПЛК) и их прикладного ПО. Вслед за необходимостью в прикладном ПО ПЛК растут и расходы предприятий и проектных организаций на его разработку, внедрение и сопровождение. Актуальной становится задача снижения затрат по данной статье с сохранением или повышением показателей качества ПО. Эффективная разработка подразумевает получение качественного ПО (ГОСТ 28195-89 «Оценка качества программных средств. Общие положения») с минимальными трудозатратами.

Сохранение или повышение качества ПО с сокращением трудозатрат особенно актуально на технологическом уровне производств, где ошибки и аварии могут привести к значительным затратам на этапе эксплуатации, в том числе связанным с простоями и ремонтом дорогостоящего оборудования. В отдельном случае опасных производственных объектов высокие требования к качеству прикладного ПО ПЛК критически важны и обуславливаются угрозой жизни и здоровью людей. Таким образом, повышение эффективности разработки прикладного ПО ПЛК не только снижает расходы предприятий и проектных организаций, но и имеет значение для общества.

Об актуальности повышения эффективности разработки прикладного ПО ПЛК свидетельствует также большое количество исследований в данном направлении как российских, так и зарубежных ученых.

Исследование существующих работ показало, что основное внимание сосредоточено на нескольких аспектах, а именно: улучшение согласованности проектных решений со всеми участниками разработки, обеспечение возможности автоматически проверить полноту и непротиворечивость проектных решений и расширение возможностей повторного использования готовых проектных решений. Все указанные аспекты имеют влияние на эффективность разработки. Под проектным решением в области разработки принято понимать проект архитектуры ПО в целом и детальный проект каждого отдельного компонента этой архитектуры (ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология (ИТ). Системная и программная инженерия. Процессы жизненного цикла программных средств»).

Исследование существующих подходов к реализации прикладного ПО ПЛК также позволило выявить отсутствие таких форм представления детального проекта, которые бы полно и однозначно интерпретировались всеми участниками в процессе согласования. Для согласованности с заказчиком распространено представление архитектурных проектных решений в виде визуальных моделей управляемых объектов. Данные модели являются эффективным ин-

струментом, однако вопрос их использования для детального проектирования остается нерассмотренным.

Таким образом, разработка визуальных моделей и алгоритма детального проектирования с целью повышения эффективности разработки прикладного ПО ПЛК является актуальной задачей.

**Степень разработанности темы исследования.** Вопросам использования формальных спецификаций алгоритмов управления посвящены работы А. Д. Закревского, В. Н. Захарова, Д. А. Поспелова, В. Е. Хазацкого, А. А. Шалыто, Е. В. Кузьмина, В. А. Соколова, В. Д. Червенчука, А. С. Хадеева, Г. Фрайтаг, В. Годе, Э. Хамби, С. Cheng, Н. Ruess, Н. Kress-Gazit, I. Majzik. Методы использования визуальных моделей представлены в работах А. В. Рыбакова, О. А. Большакова, а также зарубежных специалистов – R. A. Havner, S. D. Ryan, J. F. Toman, C. J. Whitehead, Jr., K. P. Yuen, M. P. Bender, J. P. McAndrew, M. J. Schwenke, J. A. Sinclair, R. J. Staron, W. Hoefler, M. Tiegelkamp.

Работа выполнена в ФГБОУ ВО «Оренбургский государственный университет» в рамках следующих научных направлений №01200902662 «Анализ и синтез информационных и технических систем»; №121102900209-9 «Развитие методологических основ принятия решений на основе современных информационных технологий и методов синтеза сложных систем в автоматизированных системах промышленных объектов (промышленности)».

**Объектом исследования** является процесс проектирования прикладного программного обеспечения программируемых логических контроллеров.

**Предмет исследования** составляют модели и алгоритмы проектирования прикладного программного обеспечения программируемых логических контроллеров.

**Цель и задачи исследования.** Цель работы состоит в повышении эффективности проектирования прикладного программного обеспечения программируемых логических контроллеров на основе визуальных моделей.

Для достижения поставленной цели необходимо решить следующие основные задачи:

1. Разработать предметно-ориентированные визуальные модели объектов управления как человеко-машинный интерфейс при детальном проектировании прикладного программного обеспечения программируемых логических контроллеров (соответствует п. 7 паспорта специальности 2.3.5).

2. Разработать алгоритм проектирования прикладного программного обеспечения программируемых логических контроллеров на основе использования предметно-ориентированных визуальных моделей объектов управления и автоматов Мура (соответствует п. 1 паспорта специальности 2.3.5).

3. Реализовать предложенные модели и алгоритм в виде структуры программного комплекса, интегрированной среды разработки и утилиты проверки корректности моделей (соответствует п. 2 паспорта специальности 2.3.5).

4. Провести анализ эффективности применения разработанных моделей, алгоритма и программного комплекса (соответствует пп. 1, 2 паспорта специальности 2.3.5).

**Положения, выносимые на защиту:**

1. Предметно-ориентированные визуальные модели объектов управления, обеспечивающие человеко-машинный интерфейс взаимодействия при проектировании прикладного программного обеспечения программируемых логических контроллеров.

2. Алгоритм проектирования прикладного программного обеспечения программируемых логических контроллеров, позволяющий реализовать совместное использование предметно-ориентированных визуальных моделей объектов управления и автоматов Мура.

3. Структура программного комплекса, позволяющая обеспечить корректность прикладного программного обеспечения программируемых логических контроллеров с использованием адаптированного алгоритма минимизации и отрицания дизъюнктивных нормальных форм и продукционных экспертных систем.

4. Анализ эффективности применения разработанных моделей, алгоритма и программного комплекса.

**Научная новизна.** В работе получены следующие результаты, отличающиеся научной новизной:

1. Предметно-ориентированные визуальные модели объектов управления, отличающиеся от известных наличием операций детального проектирования спецификаций программных блоков и средств визуальной интерпретации спецификаций в виде совокупности кадров, что позволяет обеспечить согласованность проектных решений.

2. Алгоритм проектирования прикладного программного обеспечения программируемых логических контроллеров на основе визуальных моделей объектов управления и автоматов Мура, отличающийся использованием имитационного моделирования операторского управления для детального проектирования, позволяющий сократить трудозатраты на реализацию и повысить качество программного обеспечения.

3. Структура программного комплекса в виде совокупности утилиты проверки корректности моделей и интегрированной среды разработки прикладного программного обеспечения программируемых логических контроллеров, отличающаяся использованием адаптированного для автоматов Мура алгоритма минимизации и отрицания дизъюнктивных нормальных форм и продукционных экспертных систем, обеспечивающая повышение качества программного обеспечения.

**Теоретическая значимость** исследования состоит в том, что предложены новые модели и алгоритм проектирования прикладного программного обеспечения программируемых логических контроллеров на основе использования данных моделей. Положения и выводы, содержащиеся в данной работе, могут быть использованы в развитии средств разработки прикладного программного обеспечения программируемых логических контроллеров.

**Практическая значимость** заключается в реализации предложенных моделей и алгоритма в виде программного комплекса, обеспечивающего повы-

шение качества прикладного программного обеспечения программируемых логических контроллеров при детальном проектировании. На компоненты программного комплекса получены три свидетельства о государственной регистрации в реестре Федеральной службы по интеллектуальной собственности. Результаты исследования могут быть использованы в научно-исследовательских и проектных организациях, занимающихся разработкой прикладного программного обеспечения автоматизированных систем управления технологическими процессами.

**Методы исследования.** Использованы методы теории реляционных моделей данных, теории логического управления, технологии разработки программного обеспечения, искусственного интеллекта, теории формальных языков.

**Реализация и внедрение результатов работы.** Результаты диссертационной работы, представленные программным комплексом в виде программного обеспечения «Среда разработки программ логического управления SWSYN Control Builder v 1.0», «Библиотека проверки корректности визуальных моделей автоматизированных технологических комплексов Plant Model Checker v 1.0» и «Программный модуль проверки корректности визуальных моделей автоматизированных технологических комплексов Plant Model Checker CLI v 1.0», а также методическим обеспечением и практическими рекомендациями, были приняты к использованию в АО «МК ОРМЕТО-ЮУМЗ» (г. Орск) и ПАО «Орскнефтеоргсинтез» (г. Орск) для разработки и обслуживания программного обеспечения АСУТП.

**Апробация результатов диссертации.** Основные результаты диссертационной работы докладывались и обсуждались на следующих конференциях: Всероссийская научно-практическая конференция «Компьютерная интеграция производства и ИПИ-технологии» (г. Оренбург, 2015, 2017 гг.), XIII Областной конкурс инженерного искусства (г. Оренбург, 2016 г.), VII Национальная научно-техническая конференция (г. Москва, 2017 г.), Межрегиональный конкурс инженерного искусства «Оренбург – Уфа» (г. Уфа, 2017 г.), Всероссийская научно-практическая конференция «Актуальные проблемы автоматизации управления на предприятии и в организации» (г. Орск, 2017 г.), Всероссийская научно-практическая конференция «Теоретические вопросы разработки, внедрения и эксплуатации программных средств» (г. Орск, 2018 г.), IX Всероссийская научно-техническая конференция «Энергетика: состояние, проблемы, перспективы» (г. Оренбург, 2018 г.), Научно-практическая конференция «Роль нефтегазового сектора в технико-экономическом развитии Оренбуржья» (г. Оренбург, 2021 г.).

**Публикации.** По теме диссертации опубликовано 16 печатных работ, в том числе пять статей в журналах из «Перечня ...» ВАК РФ, получены три свидетельства о государственной регистрации программ для ЭВМ. В работах, опубликованных в соавторстве и приведенных в конце автореферата, лично автором получены следующие результаты: [15, 16] – анализ существующих подходов к проектированию прикладного ПО ПЛК; [5] – алгоритм проектирования

прикладного ПО ПЛК с использованием визуальных моделей; [13] – определения, связанные с синтаксической и семантической корректностью; [4] – подход к обеспечению семантической корректности спецификаций программных блоков на основе продукционных экспертных систем; [7, 8] – программная реализация библиотеки и утилиты проверки визуальных моделей; [14] – реализация прикладного ПО ПЛК с использованием предлагаемых в диссертационной работе моделей и алгоритма.

**Структура и объем работы.** Диссертационная работа состоит из введения, четырех глав, заключения и приложений. Работа изложена на 179 страницах машинописного текста, включая 68 рисунков, 12 таблиц и список литературы из 122 наименований.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

**Во введении** приведены обоснование актуальности выбранной темы, степень ее разработанности, цель и задачи, объект и предмет исследования, методы исследования, научная новизна, теоретическая и практическая значимость работы, реализация и апробация результатов.

**В первой главе** проведен анализ существующих языков, процессов и методов реализации прикладного ПО ПЛК в системах логического управления (СЛУ).

Описана предметная область рассматриваемого в данной работе процесса проектирования прикладного ПО – ПЛК, реализующие СЛУ.

Проведен анализ стандартных языков программирования ПО ПЛК МЭК 61131-3, который показал, что несмотря на нацеленность некоторых языков на использование специалистами различных областей, ни один из них не позволяет привлекать всех участников к согласованию одновременно. Кроме того, отсутствует возможность верификации.

Проанализирован процесс реализации прикладного ПО ПЛК при использовании в СЛУ по ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология (ИТ). Системная и программная инженерия. Процессы жизненного цикла программных средств». Выделены следующие основные проблемы, сопровождающие процесс реализации прикладного ПО ПЛК:

1. Семантический разрыв между разработчиком (в лице аналитика и программиста) и заказчиком.

2. Отсутствие способа описания требований, полностью и однозначно понимаемого всеми участниками процесса реализации. Данное обстоятельство затрудняет обеспечение согласованности проектных решений с заказчиком.

3. Сложность декомпозиции требований, вытекающая из первых двух проблем.

4. Низкая эффективность ручного кодирования. Под эффективностью здесь следует понимать отношение численного значения показателя качества ПО к количеству затраченного времени.

Анализ существующих методов реализации прикладного ПО ПЛК в СЛУ показал, что для решения проблем 1 и 2 могут быть использованы визуальные языки программирования LD, FBD, SFC, а также различные формальные языки алгоритмизации. При этом оба подхода применимы при детальном проектировании, обладают полнотой и однозначностью описания алгоритмов, обеспечивают возможность согласования с разработчиком. Однако, с точки зрения заказчика, данные подходы не переводят алгоритмы в плоскость его предметной области, а лишь повышают уровень абстракции описания последних. Для решения проблемы сложности декомпозиции используются компьютерные визуальные модели. Использование визуальных объектов из предметной области обеспечивает возможность согласования с заказчиком, но не с разработчиком. Кроме того, существующие модели не интегрированы с проектными процедурами детального проектирования программных блоков. Для решения проблемы низкой эффективности ручного кодирования широко применяется практика использования библиотек готовых программных блоков (например, библиотека APL фирмы Siemens). Данный подход обладает преимуществами и недостатками, аналогичными характерным формальным спецификациям.

Проведенный анализ существующих методов показал отсутствие таких визуальных моделей, которые могли бы использоваться при детальном проектировании, обеспечивали возможность для согласования, как с заказчиком, так и с разработчиком, и при этом опирались на математический аппарат формальных спецификаций.

Таким образом, на основании указанных результатов предложено использование визуальных моделей в качестве человеко-машинного интерфейса взаимодействия между пользователем и разрабатываемыми алгоритмами в виде автоматов Мура.

**Вторая глава** посвящена разработке предметно-ориентированных визуальных моделей и алгоритма проектирования прикладного ПО ПЛК.

В качестве человеко-машинного интерфейса для проектирования прикладного ПО предложены модели, визуально схожие с автоматизированными технологическими комплексами (АТК). Визуальная модель komponуется из заранее подготовленных изображений частей АТК – иконок. Аналитически визуальная модель представляется в виде двух объектов:

$$\langle L, M \rangle,$$

где  $L$  – формальный язык моделирования,  $M$  – реляционная составляющая. Реляционная составляющая  $M$ , в свою очередь, определяется как совокупность из 14 отношений:

$$M = (R_{ED}, R_{EH}, R_{PD}, R_{PI}, R_{TTD}, R_{TTH}, R_{TI}, R_{RD}, R_{SD}, R_{SM}, R_{AD}, R_{AM}, R_{ST}, R_{TT}).$$

Отношение  $R_{ED}$  представляет элементы АТК (узлы механизма, исполнительные механизмы, датчики), а отношение  $R_{EH}$  – список связей между элементами.  $R_{PD}$  – отношение, содержащее данные об иконках, а  $R_{PI}$  – отношение, сопоставляющее некоторым элементам иконки, а также координаты их расположения на некоторой плоскости и геометрические размеры. Отношение  $R_{TTD}$  описывает группы элементов. Отношение  $R_{TTH}$  представляет собой список свя-

зей между парами групп (по аналогии с  $R_{EH}$ ). Отношение  $R_{TI}$  указывает, к какой группе относится элемент и позволяет более формально отличать элементы друг от друга, группировать их. Последующие отношения используются для интеграции спецификаций программных блоков в модель. Отношение  $R_{PI}$  задает соответствия между элементами и иконками и позволяет получить визуальную интерпретацию модели на плоскости. На рисунке 1а изображена визуальная интерпретация модели АТК передвижной тележки.

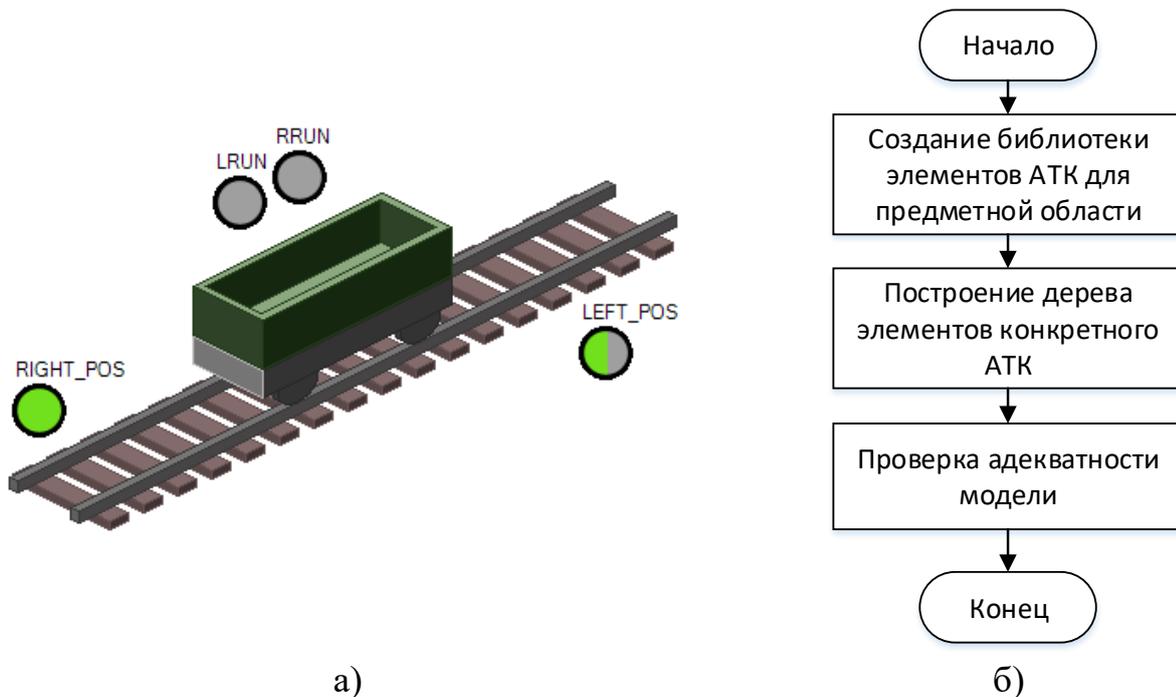


Рисунок 1 – Пример визуальной модели АТК передвижной тележки (а) и алгоритм построения визуальной модели (б)

Для формальной проверки моделей введено понятие конфигурации модели и предложено использование формальных грамматик для описания языка моделирования  $L$ . Конфигурация модели есть дерево элементов, вершины которого помечены идентификаторами групп соответствующих элементов. Порождающая процедура (формальная грамматика) языка моделирования имеет вид:

$$G = \langle N, T, P, S \rangle;$$

$$T = \{ (, ), , \} \cup R_{TTD},$$

где  $N$  – множество нетерминалов,  $T$  – множество терминалов,  $P$  – множество правил грамматики,  $S$  – начальный нетерминал. Класс необходимой порождающей грамматики по Хомскому определяется исходя из особенностей предметной области и условий применения моделей. Для каждой сферы применения индивидуально разрабатывается дерево групп ( $R_{TTD}$  и  $R_{TTH}$ ), порождающая грамматика и распознающая программа. Входными данными распознающей процедуры являются левые скобочные представления конфигураций.

Разработан алгоритм построения визуальной модели, основные шаги которого представлены на рисунке 1б.

Модель содержит как минимум три группы элементов: входы ПЛК (или программного блока), выходы ПЛК (или программного блока) и прочие элементы. Каждому входу в каждый момент времени сопоставлено одно из трех изображений, соответствующих его значениям – ложь, истина, неопределенность. Каждому выходу в каждый момент времени сопоставляется одно из двух изображений – ложь, истина. Модель, отличающаяся от исходной только иконками входов и выходов, но не их количеством или названием, называется кадром. Любой кадр показывает определенные значения входов и выходов.

Для интеграции спецификаций программных блоков в модели предложен математический аппарат автоматов Мура. Введено определение автомата Мура (АМ) в рамках разработанных моделей:

$$A = \langle Q, q_0, \{M_S\}, \{M_R\}, \Phi, \Psi \rangle;$$

$$\Phi: Q \times \{M_S\} \rightarrow Q;$$

$$\Psi: Q \rightarrow \{M_R\},$$

где

$Q = (q_1, q_2, \dots, q_p)$  – множество внутренних состояний АМ.

$q_0 \in Q$  – начальное состояние.

$\{M_S\}$  – множество кадров визуальной модели, соответствующих комбинациям значений входных переменных программного блока.

$\{M_R\}$  – множество кадров визуальной модели, соответствующих комбинациям значений выходных переменных программного блока.

$\Phi$  – функция переходов.

$\Psi$  – функция выходов.

Обеспечена интеграция визуальной модели и формальных спецификаций программных блоков введением в модель отношений с  $R_{RD}$  по  $R_{TT}$ . Отношение  $R_{RD}$  – задает множество спецификаций, интегрированных в модель,  $R_{SD}$  – соответствует перечню комбинаций входов (ситуаций), без указания значений входов,  $R_{SM}$  – список соответствий значений входов ситуациям (так называемая карта ситуаций),  $R_{AD}$  – множество комбинаций выходов (действий), без указания значений выходов,  $R_{AM}$  – карта действий,  $R_{ST}$  – соответствует таблице состояний,  $R_{TT}$  – таблица переходов.

Введены определения для операций детального проектирования спецификаций в моделях: СМЕНА – позволяет пользователю переназначать для входа или выхода значение, СОСТОЯНИЕ – позволяет пользователю получить по текущему кадру вектор действия  $r$  и увязать его с новым состоянием  $q$ , ПЕРЕХОД – позволяет выбрать исходное состояние из  $Q$ , получить вектор ситуации  $s$  по кадру и выбрать результирующее состояние из  $Q$ . Операция СМЕНА выполняется кликом мыши по соответствующим изображениям входов и выходов визуальной модели.

Таким образом, разработаны предметно-ориентированные визуальные модели объектов управления, отличающиеся от известных наличием операций детального проектирования спецификаций программных блоков и средств визуальной интерпретации спецификаций в виде совокупности кадров, что позволяет обеспечить согласованность проектных решений.

Предложен алгоритм проектирования прикладного ПО ПЛК, заключающийся в следующем:

1. Для предметной области разрабатывается дерево групп ( $R_{TID}$  и  $R_{TIN}$ ), порождающая грамматика и распознающая программа.
2. Для конкретного АТК строится визуальная модель.
3. Путем имитации работы оператора с данным АТК определяется спецификация каждого программного блока.
4. По спецификациям автоматически генерируется программный код.

Разработан автоматизированный алгоритм проектирования спецификаций, представленный на рисунке 2.

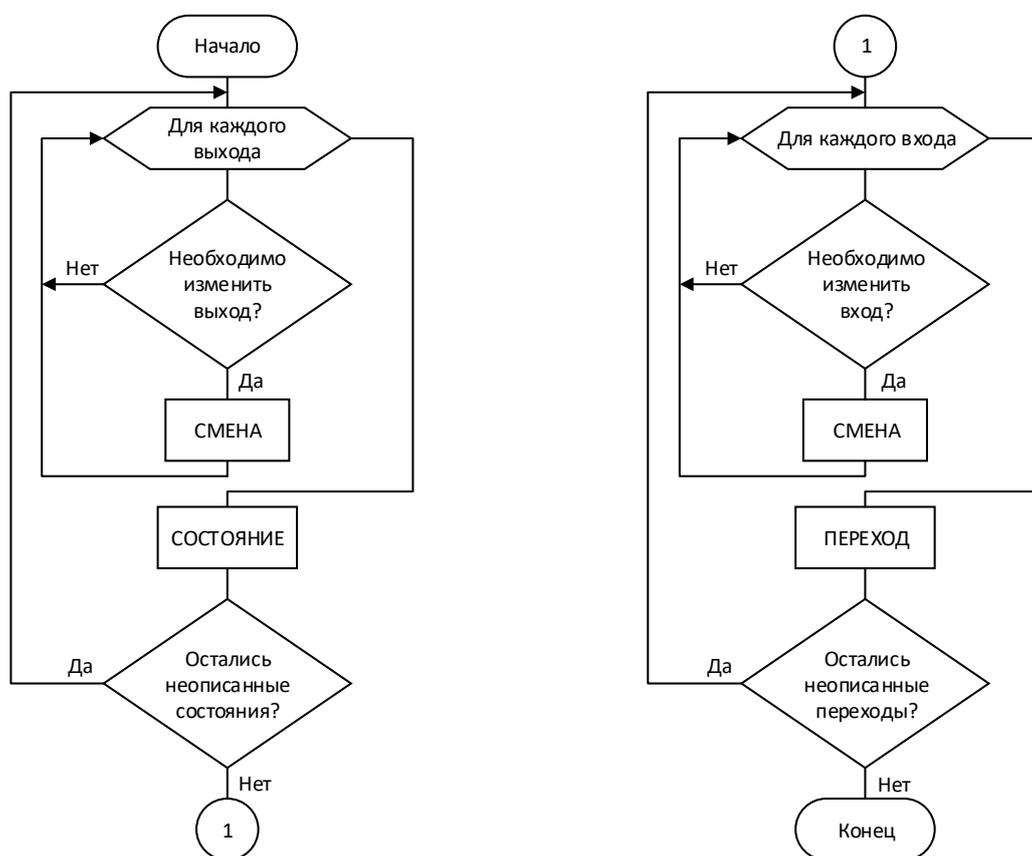


Рисунок 2 – Алгоритм проектирования спецификаций

Даны практические рекомендации для повышения уровня абстракции работы с аналоговыми сигналами, временными задержками и счётчиками в управляющих алгоритмах.

Таким образом, разработан алгоритм проектирования, отличающийся использованием имитационного моделирования операторского управления для детального проектирования программных блоков на основе визуальных моде-

лей. Предложенные модели и алгоритм позволяют обеспечить согласованность спецификаций как с заказчиком (в виде совокупности визуальных кадров), так и с разработчиком (в виде автоматов Мура), что в свою очередь дает возможность выявлять и устранять смысловые ошибки на ранних стадиях реализации ПО, разрабатывать ПО, максимально соответствующее замыслу заказчика.

**Третья глава** посвящена разработке структуры и реализации программного комплекса с использованием продукционных экспертных систем и адаптированного к автоматам Мура алгоритма минимизации и отрицания дизъюнктивных нормальных форм (ДНФ).

Даны определения, связанные с синтаксической и семантической корректностью, на основе рассуждений А. П. Еремеева в работе «О корректности продукционной модели принятия решений на основе таблиц решений».

Множеством синтаксически возможных простых ситуаций (универсумом ситуаций) называется множество  $S$  всех  $2^n$  возможных наборов значений входов модели.

Множеством допустимых простых ситуаций называется подмножество  $S'$  множества  $S$ :

$$S' \subseteq S.$$

Обозначим  $\Phi_p$  множество переходов из заданного состояния  $q_p$ .

Ситуация  $s = (x_1, x_2, \dots, x_n)$  распознаётся переходом  $\varphi = ((q, s'), q')$ ,  $q, q' \in Q$ ,  $\varphi \in \Phi_p$ , если:

$$s' = (x'_1, x'_2, \dots, x'_n);$$

$$(\forall i)[x'_i \neq \lambda \Rightarrow x'_i = x_i].$$

Тот факт, что ситуация  $s$  распознаётся переходом  $\varphi$ , будем обозначать как  $s \rightarrow \varphi$  (когда это не будет приводить к противоречиям со знаком отображения). А если необходимо понимать при этом, в какое состояние перейдёт АМ, то используется следующая запись:

$$(s \rightarrow \varphi) = q'.$$

Множество переходов  $\Phi_p$  состояния  $q_p$  является непротиворечивым относительно множества ситуаций  $S'$ , если:

$$(\forall s \in S', \varphi_j, \varphi_l \in \Phi_p) \left[ (s \rightarrow \varphi_j \wedge s \rightarrow \varphi_l) \Rightarrow ((s \rightarrow \varphi_j) = (s \rightarrow \varphi_l)) \right].$$

Множество переходов  $\Phi_p$  состояния  $q_p$  является полным относительно множества ситуаций  $S'$ , если:

$$(\forall s \in S' \exists \varphi \in \Phi_p) [(s \rightarrow \varphi) = q, q \in Q].$$

Множество переходов  $\Phi_p$  состояния  $q_p$  является корректным относительно множества ситуаций  $S'$ , если оно полно и непротиворечиво относительно  $S'$ .

Алгоритм (АМ) называется семантически корректным, если множества переходов каждого его состояния корректны относительно множества допустимых ситуаций  $S'$ . Алгоритм (АМ) называется синтаксически корректным, если множества переходов каждого его состояния корректны относительно множества синтаксически возможных ситуаций  $S$ .

Для обеспечения синтаксической корректности адаптирован алгоритм минимизации и отрицания ДНФ, предложенный В. Д. Червенчуком применительно к таблицам решений. Алгоритм минимизации и отрицания ДНФ заключается в построении ДНФ, соответствующей множеству учтенных ситуаций и выполнении логического отрицания над ней. При этом конъюнкции новой ДНФ соответствуют ситуациям, для которых не определены переходы. Преимущество данного алгоритма состоит, во-первых, в том, что он позволяет не только фиксировать факт наличия неполноты, но и определить все неучтенные ситуации. Во-вторых, роль пользователя ограничивается назначением парам состояний и обобщенных ситуаций новых состояний, а обеспечение корректности осуществляется автоматически в режиме диалога.

Для разработки семантически корректных спецификаций на основе знаний предметной области управляемого объекта, предложено использование продукционных экспертных систем (ЭС), в среде разработки CLIPS.

Продукционные правила ЭС разделяются на основные и вспомогательные. Основные правила выполняют анализ кортежей составляющей  $M$  и добавляют по результатам анализа факты, характеризующие наборы значений входов и выходов, требующие внимания и являющиеся потенциальным источником нарушений в работе алгоритма. Факты такого рода называются ограничивающими наборами. На рисунке 3 представлены форматы описания ограничивающего набора и основных правил ЭС.

```
(deftemplate bounding-set
  (slot id)
  (slot element-id)
  (multislot variables)
  (multislot values)
  (slot message-text))
```

а)

```
(defrule <название правила>
  <выражение, характеризующее
  состав и структуру модели АТК>
  =>
  (assert <ограничивающий набор>))
```

б)

Рисунок 3 – Формат ограничивающего набора (а) и основных правил (б)

Вспомогательные правила информируют пользователя в том случае, если определяемый им в данный момент вектор ситуации или действия пересекается с одним из ограничивающих наборов. Формат использования вспомогательных правил представлен на рисунке 4.

Для обеспечения дополнительного контроля семантической корректности разработан подход к тестированию спецификаций на основе моделирования управляемых объектов. Для моделирования алгоритмов поведения использован тот же механизм, что и для создания управляющих алгоритмов. Возможность различать алгоритм управления и алгоритм поведения технологического объекта управления обеспечивается введением понятия инвертированного алгоритма.

```

(defrule <название правила>
  <ограничивающий набор>
  <совпадение значений сигналов модели со значениями сигналов
ограничивающего набора>
  =>
  <уведомление пользователя>)

```

Рисунок 4 – Формат описания вспомогательных правил ЭС

В инвертированном алгоритме выходы ПЛК становятся входами, а некоторые входы ПЛК – выходами.

Разработана структура интегрированной среды разработки (ИСР) прикладного ПО ПЛК, включающая компонент проверки корректности спецификации с использованием алгоритма минимизации и отрицания ДНФ и продукционных экспертных систем. Разработана структура программного комплекса, включающая ИСР и утилиту проверки корректности визуальных моделей (рисунок 5).

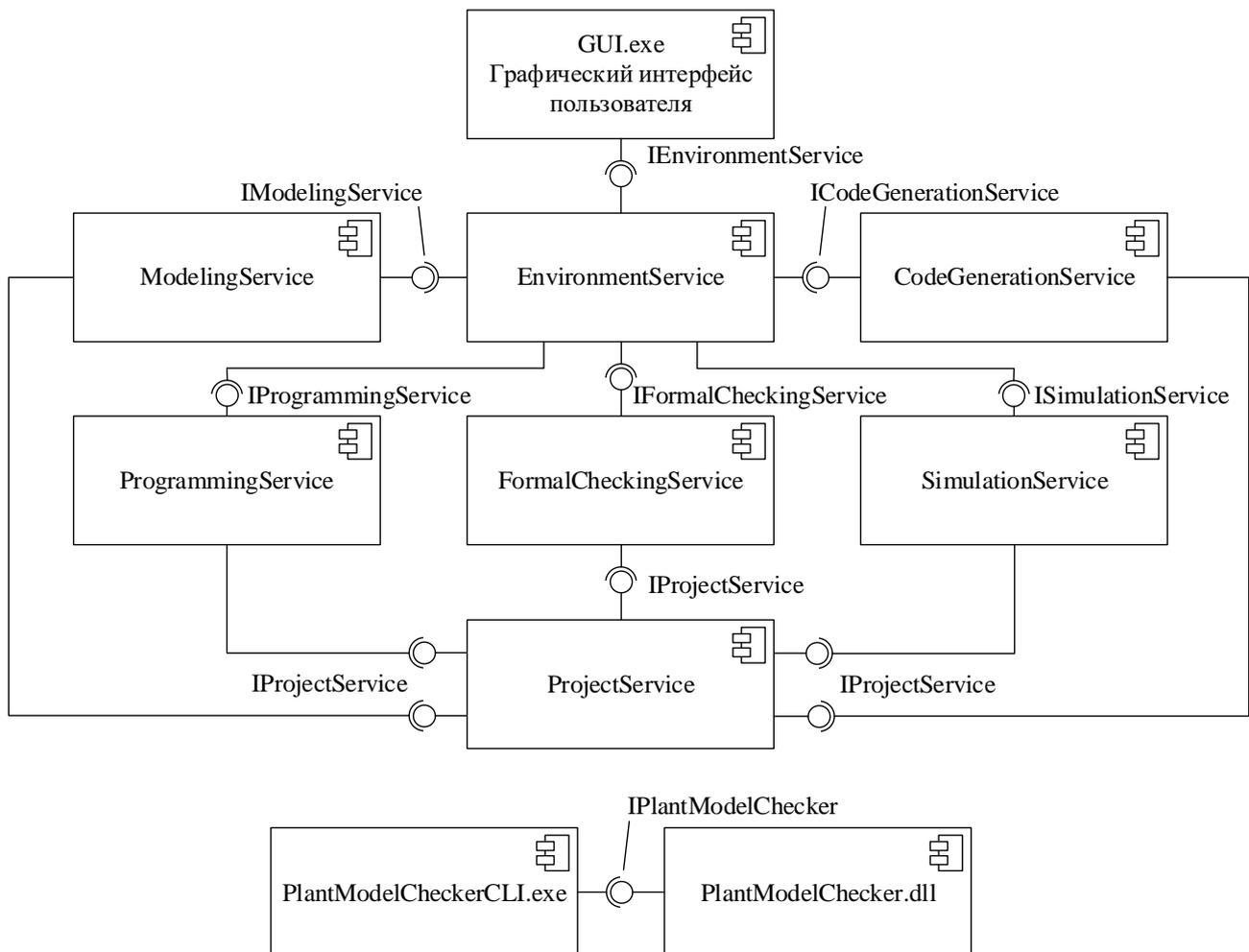


Рисунок 5 – Структура программного комплекса разработки прикладного ПО ПЛК

Компоненты PlantModelChecker.dll и PlantModelCheckerCLI.exe составляют программное средство проверки корректности визуальных моделей, а совокупность всех остальных компонентов – ИСП.

С учетом разработанной структуры реализована ИСП SWSYN Control Builder с использованием Windows Forms и языка программирования C#, а также библиотека Plant Model Checker и утилита Plant Model Checker CLI. Главное окно SWSYN Control Builder представлено на рисунке 6.

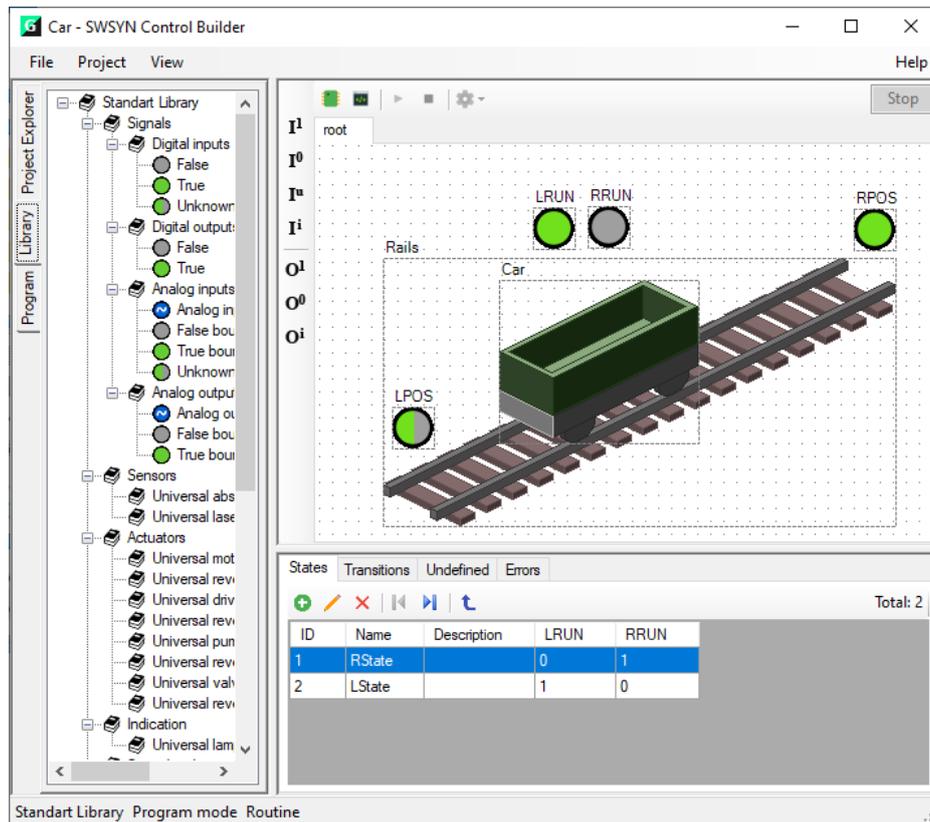


Рисунок 6 – Главное диалоговое окно реализованной ИСП SWSYN Control Builder

Таким образом, разработана структура программного комплекса, реализована ИСП SWSYN Control Builder с использованием Windows Forms и языка программирования C#, а также библиотека Plant Model Checker и утилита Plant Model Checker CLI.

**В четвёртой главе** диссертации представлены результаты экспериментальной оценки эффективности применения разработанных моделей, алгоритма и программного комплекса.

Проведено исследование эффективности программной процедуры поиска неучтенных ситуаций компонента проверки корректности спецификаций. В исследовании использовались четыре ЭВМ, различающихся по характеристикам. На рисунке 7 представлены графики зависимости среднего времени выполнения программы от количества входных переменных в трех случаях.

В худшем случае определен переход только для одной произвольной простой ситуации. В среднем случае определены переходы ровно для половины всех возможных ситуаций. В лучшем случае определены переходы для всех возможных простых ситуаций.



Рисунок 7 – Графики зависимости среднего времени выполнения программы от количества входных переменных

Как видно из графиков, наблюдается резкое увеличение времени выполнения процедуры в худшем случае при изменении числа входов с 11 до 12. "Вычистка" одного состояния от неучтенных ситуаций при числе входов менее 12, в среднем, на практике, требует не более 10 повторений процедуры. При этом время работы процедуры при 11 входах практически остаётся приемлемым и не отражается негативно на пользовательском опыте, тогда как 12 входов могут существенно его ухудшить.

Опираясь на результаты исследования можно сделать вывод о том, что реализованное программное обеспечение остается эффективным при числе входов спецификации меньше, либо равным 11. Исходную задачу необходимо декомпозировать вручную программисту на блоки с количеством входов меньше, либо равным 11, что также будет способствовать более строгому соблюдению чистоты архитектуры и принципа разделения ответственности.

Рассмотрено применение предложенных в диссертационной работе моделей, алгоритма и программного комплекса к задаче разработки программы управления трансбордером – платформой для поперечного перемещения полувагонов между двумя железнодорожными путями. В качестве основного технического средства был использован ПЛК серии S7-300 фирмы Siemens, в качестве целевой среды – Simatic Step 7.

Для предметной области разработано дерево групп из 41 наименования, формальная грамматика и распознающая программа, база знаний из двух продукций, библиотека элементов из 50 наименований.

Выполнена декомпозиция исходной задачи на четыре основные подзадачи: управление в ручном режиме, управление в полуавтоматическом режиме, логика работы контакторов и логика работы частотного преобразователя.

Для каждой подзадачи разработана визуальная модель и спецификация программного блока, сгенерирован программный код на языке SCL, проведено тестирование и экспериментальная проверка. На рисунке 8 представлен фрагмент визуальной модели блока ручного управления и интерфейс блока контактора частотного преобразователя.

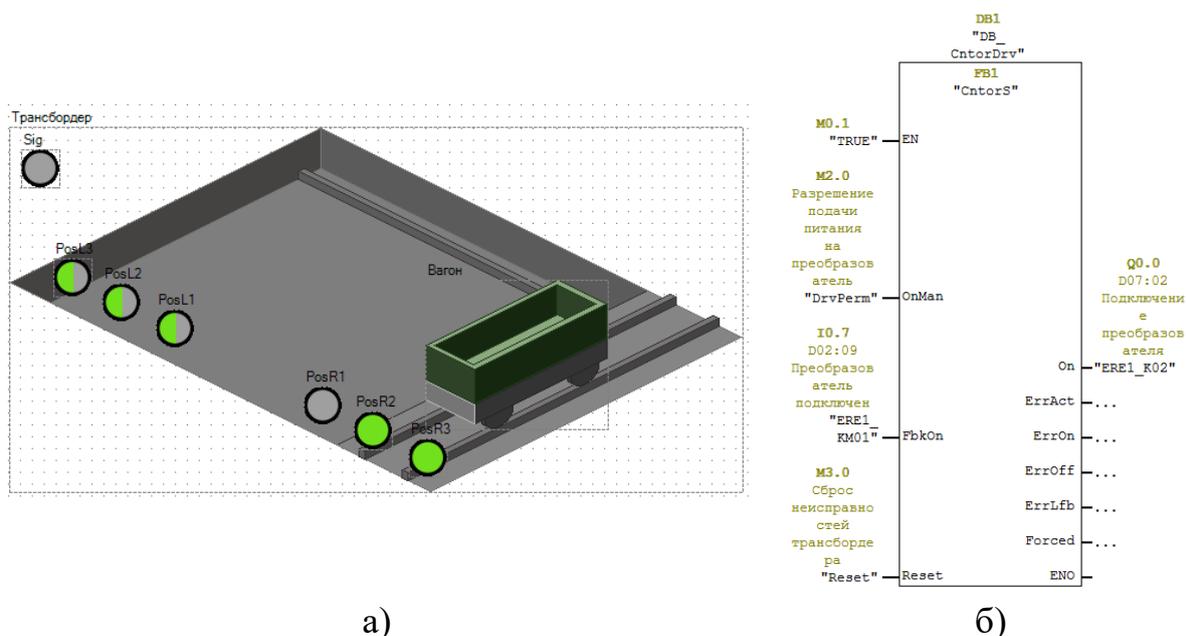


Рисунок 8 – Фрагмент визуальной модели блока ручного управления (а) и интерфейс блока контактора частотного преобразователя (б)

Эффективность проектирования ПО определяется отношением значений показателей качества ПО к трудозатратам.

Для оценки качества разработанного ПО, в соответствии с ГОСТ 28195-89 была выбрана номенклатура показателей качества. При выборе учитывалось влияние предлагаемого подхода на тот или иной показатель. С использованием экспертного метода определены значения оценочных элементов программ управления перегрузочно-усреднительным оборудованием, разработанных в отечественных проектных подразделениях традиционным подходом в период с 2016 по 2023 годы.

Определены значения оценочных элементов разработанной программы управления трансбордером. В таблице 1 представлены полученные значения оценочных элементов.

Таблица 1 – Значения оценочных элементов

	H0103	C0304	C0301	Y0101	Y0307	K0303
Базовые значения	0,73	0,5	0,53	0,24	0,81	0,73
Предлагаемый подход	0,85	0,96	1	0,59	1	1

По формулам, представленным в ГОСТ 28195-89, рассчитаны значения метрик, критериев и факторов качества. На рисунке 9 представлена гистограмма полученных факторов.

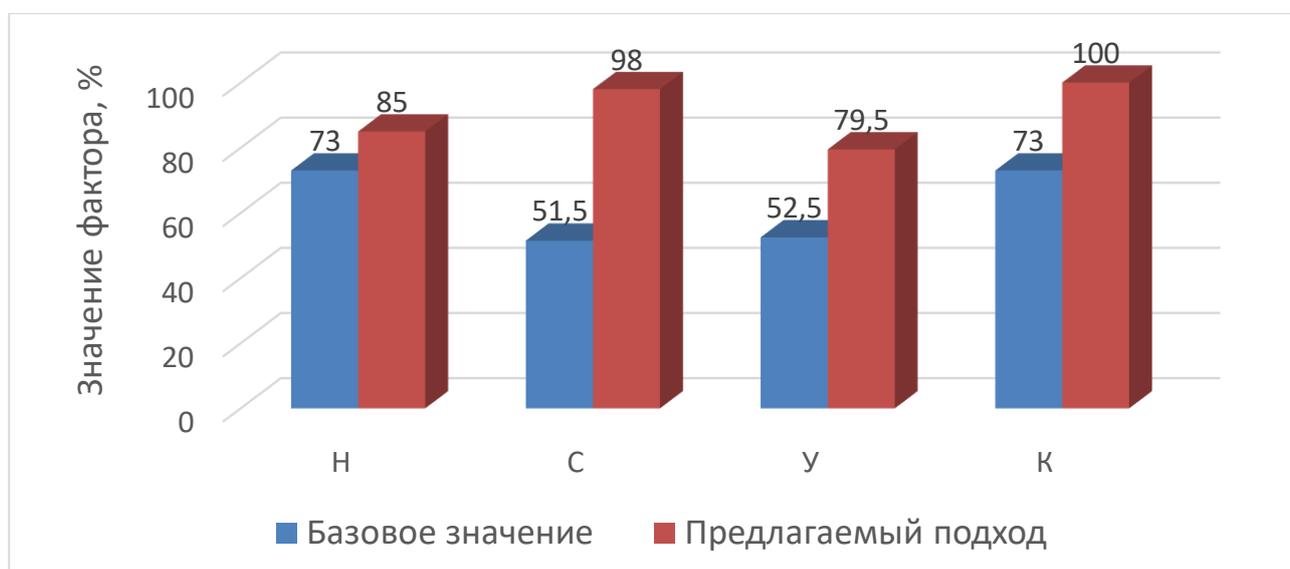


Рисунок 9 – Гистограмма факторов качества (Н – надежность, С – сопровождаемость, У – удобство применения, К – корректность)

На рисунке 10 представлены графики распределения трудозатрат по процессам ЖЦ.

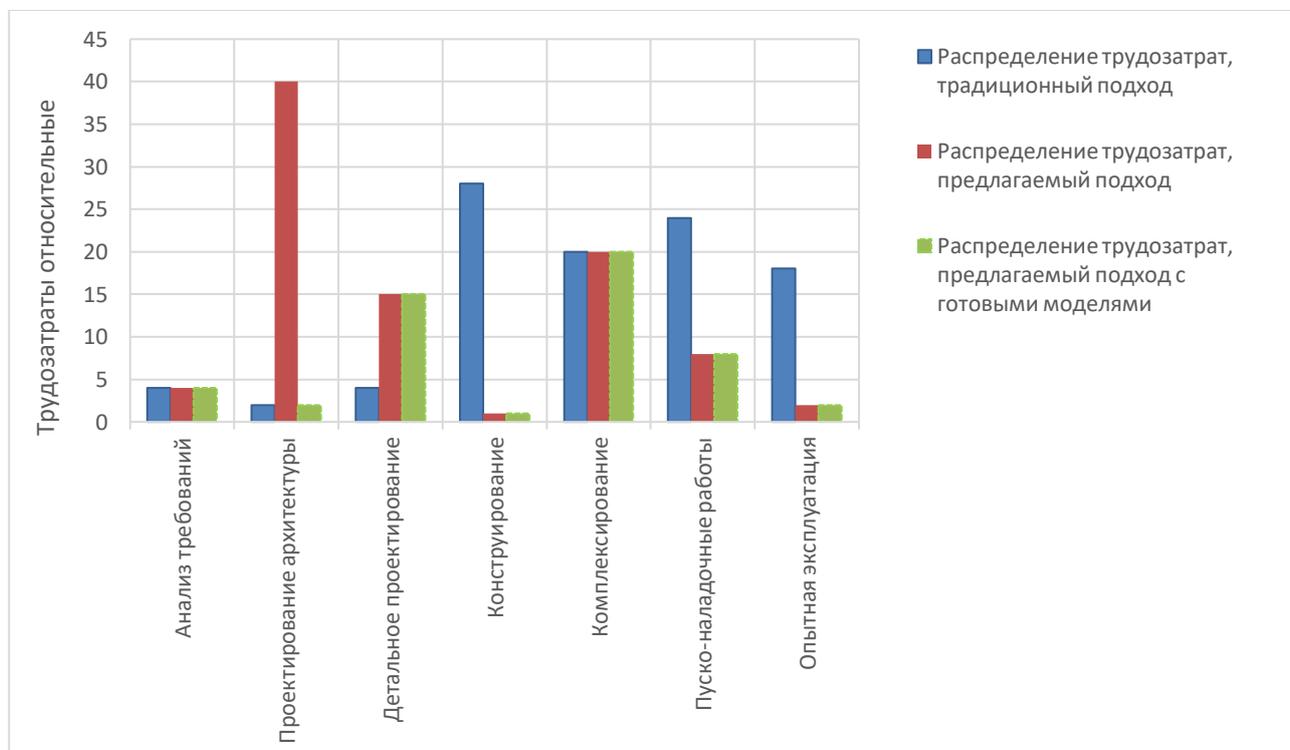


Рисунок 10 – Графики распределения трудозатрат по процессам ЖЦ

На этапе архитектурного проектирования появляется необходимость в разработке групп элементов, грамматик, библиотек элементов и визуальных моделей, что существенно увеличивает трудозатраты. В среднем, предлагаемый подход позволяет сократить трудозатраты от 10% до 20% в зависимости от широты охвата предметной области. При наличии реализованной предметной области предлагаемый подход позволяет снизить трудозатраты от 40% до 60% в зависимости от числа нетиповых визуальных моделей. Чем чаще возникает потребность в разработке новой нетиповой модели, тем выше трудозатраты.

Таким образом, предложенные модели, алгоритм и программный комплекс обеспечивают повышение эффективности проектирования прикладного ПО ПЛК, выраженное увеличением надежности программ на 12%, сопровождаемости на 46,5%, удобства применения на 27% и корректности на 27%, а также уменьшением трудозатрат от 40% до 60%.

**В заключении** представлены основные выводы по работе.

## **ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ**

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Разработаны предметно-ориентированные визуальные модели объектов управления, отличающиеся от известных наличием операций детального проектирования спецификаций программных блоков и средств визуальной интерпретации спецификаций в виде совокупности кадров, что позволяет обеспечить согласованность проектных решений.

2. Разработан алгоритм проектирования прикладного программного обеспечения программируемых логических контроллеров на основе визуальных моделей объектов управления и автоматов Мура, отличающийся использованием имитационного моделирования операторского управления, позволяющий сократить трудозатраты на реализацию и повысить качество программного обеспечения.

3. Разработана структура программного комплекса в виде совокупности утилиты проверки корректности моделей и интегрированной среды разработки прикладного программного обеспечения программируемых логических контроллеров, отличающаяся использованием адаптированного для автоматов Мура алгоритма минимизации и отрицания дизъюнктивных нормальных форм и продукционных экспертных систем, обеспечивающая повышение качества программного обеспечения.

4. Исследована эффективность применения разработанных моделей, алгоритма и программного комплекса. Установлено, что применение позволяет увеличить такие показатели качества программного обеспечения как надежность на 12%, сопровождаемость на 46,5%, удобство применения на 27% и корректность на 27%, а также уменьшить трудозатраты от 40% до 60%.

Совокупность указанных результатов диссертационной работы позволяет утверждать, что достигнута цель исследования – повышена эффективность про-

ектирования прикладного ПО ПЛК на основе визуальных моделей и конечных автоматов.

**Перспективы дальнейшего исследования.** В качестве перспективы дальнейшего исследования планируется разработка методов и алгоритмов автоматического анализа и поддержки принятия решений при реализации прикладного ПО ПЛК.

## ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

### *Публикации в изданиях, рекомендованных ВАК РФ*

1. Парамонов, А.В. Графическое моделирование технологического процесса как вспомогательное средство составления алгоритма управления // Программные продукты и системы. – Тверь, 2016. – №4. – С. 89-93.

2. Парамонов, А.В. Методы и средства построения алгоритмов логического управления по графическим моделям // Программные продукты и системы. – Тверь, 2017. – №4. – С. 672-677.

3. Парамонов, А.В. Расширенные средства декларативного программирования логических контроллеров с использованием визуальных моделей // Научно-технический вестник Поволжья. – Казань, 2018. – №5. – С. 239-241.

4. Парамонов, А.В., Шепель, В.Н. О применении экспертных систем в разработке программ логического управления технологическими объектами // Научно-технический вестник Поволжья. – Казань, 2018. – №12. – С. 274-276.

5. Парамонов, А.В., Боровский, А.С. Метод и алгоритмы проектирования прикладного программного обеспечения логических контроллеров в промышленности на основе использования визуальных моделей // Автоматизация процессов управления. – Ульяновск, 2022. – №3 (69). – С. 97-105.

### *Свидетельства о государственной регистрации программ для ЭВМ*

6. Свидетельство № 2019614533 Российская Федерация. Среда разработки программ логического управления SWSYN Control Builder v 1.0: свидетельство о государственной регистрации программы для ЭВМ / А. В. Парамонов; заявитель и правообладатель Парамонов А. В. – № 2019614533; заявл. 18.03.2019; за-регистр. 05.04.2019.

7. Свидетельство № 2022660915 Российская Федерация. Библиотека проверки корректности визуальных моделей автоматизированных технологических комплексов Plant Model Checker v 1.0: свидетельство о государственной регистрации программы для ЭВМ / А. С. Боровский, А. В. Парамонов, Б. К. Жумашева; заявитель и правообладатель – ФГБОУ ВО «Оренбургский государственный университет». – № 2022660915; заявл. 09.06.2022; зарегистр. 10.06.2022.

8. Свидетельство № 2022660916 Российская Федерация. Программный модуль проверки корректности визуальных моделей автоматизированных технологических комплексов Plant Model Checker CLI v 1.0: свидетельство о государственной регистрации программы для ЭВМ / А. С. Боровский, А. В. Парамонов, Б. К. Жумашева; заявитель и правообладатель – ФГБОУ ВО «Оренбург-

ский государственный университет». – № 2022660916; заявл. 09.06.2022; зарегистрир. 10.06.2022.

*Статьи и материалы конференций*

9. Парамонов, А.В. Обзор методов автоматического построения программ ПЛК в задачах логического управления // Materials of the XI International scientific and practical conference “Cutting-edge science”. – Sheffield, UK, 2015. Volume 29. – p. 45-54.

10. Парамонов, А.В. Особенности автоматизации программирования ПЛК в задачах логического управления // Компьютерная интеграция производства и ИПИ-технологии: материалы VII Всероссийской научно-практической конференции. – Оренбург, 2015. – С. 25-31.

11. Парамонов, А.В. О корректности описания алгоритмов логического управления с помощью предметно-ориентированных графических моделей // Компьютерная интеграция производства и ИПИ-технологии: материалы VIII Всероссийской научно-практической конференции. – Оренбург, 2017. – С. 25-31.

12. Парамонов, А.В. Синтез цифровых схем по алгоритмам их функционирования в виде предметно-ориентированных графических моделей // Актуальные проблемы автоматизации управления на предприятии и в организации: сборник материалов всероссийской научно-практической конференции (17 ноября 2017, Орск). – Ставрополь: Логос, 2018. – С. 60-64.

13. Парамонов, А.В., Шепель, В.Н. Синтаксическая и семантическая корректность алгоритмов логического управления технологическими объектами // Теоретические вопросы разработки, внедрения и эксплуатации программных средств: материалы Всероссийской научно-практической конференции. – Орск, 2018. – С. 39-43.

14. Парамонов, А.В., Шепель, В.Н. Использование метода декларативного визуального программирования в задачах логического управления электроприводом // Труды IX Всероссийской научно-технической конференции «Энергетика: состояние, проблемы, перспективы». – Оренбург, 2018. – С. 290-294.

15. Парамонов, А.В., Боровский, А.С. О проблемах процессов реализации прикладного программного обеспечения программируемых логических контроллеров в промышленности // Роль нефтегазового сектора в технико-экономическом развитии Оренбуржья: материалы научно-практической конференции. – Оренбург, 2021. – С. 147-152.

16. Парамонов, А.В., Боровский, А.С. Визуальное программирование логических контроллеров в промышленности // Journal of Advanced Research in Technical Science. – Seattle, USA: SRC MS, AmazonKDP. – 2021. – Issue 25. – p. 71-73.