

На правах рукописи



Хородов Виталий Сергеевич

**РАЗРАБОТКА МЕТОДОВ И СРЕДСТВ
МНОГОАГЕНТНОГО РАСПРЕДЕЛЕННОГО
АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ
СТРУКТУРНО-ФУНКЦИОНАЛЬНЫХ
ЛИНГВИСТИЧЕСКИХ МОДЕЛЕЙ
ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ**

Специальность: 05.13.12 – Системы автоматизации
проектирования (промышленность)

Автореферат

диссертации на соискание ученой степени
кандидата технических наук

Ульяновск – 2015

Работа выполнена на кафедре «Вычислительная техника»
ФГБОУ ВПО «Ульяновский государственный технический университет»

Научный руководитель – **Афанасьев Александр Николаевич**,
доктор технических наук, доцент, профессор
кафедры «Вычислительная техника»
ФГБОУ ВПО Ульяновского государственного
технического университета

Официальные оппоненты:

Глушань Валентин Михайлович,
доктор технических наук, профессор, профессор
кафедры «Дискретная математика и методы
оптимизации» Южного федерального университета

Арсентьев Алексей Владимирович,
кандидат технических наук,
доцент кафедры «Полупроводниковая электроника
и наноэлектроника» Воронежского
государственного технического университета

Ведущая организация – **ФГАОУ ВО «Санкт-Петербургский
государственный электротехнический
университет «ЛЭТИ» им. В.И. Ульянова
(Ленина)»**

Защита состоится «30» сентября 2015 г. в 15 часов 00 минут на заседании диссертационного совета Д 212.277.01 при Ульяновском государственном техническом университете по адресу: 432027, г. Ульяновск, ул. Северный Венец, д. 32 (ауд. 211, Главный корпус).

С диссертацией можно ознакомиться в библиотеке Ульяновского государственного технического университета. Также диссертация и автореферат размещены в Internet на сайте УлГТУ - <http://www.ulstu.ru/>

Автореферат разослан «08» июля 2015 г.

Ученый секретарь
диссертационного совета,
доктор технических наук, профессор



Смирнов Виталий Иванович

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Возрастание требований к функциональным характеристикам вычислительных устройств приводит к увеличению их сложности, в частности, микросхемы СБИС являются одними из самых сложных технических объектов. Для проектирования таких устройств в настоящее время используются высокоуровневые языки описания аппаратуры HDL (Hardware Description Language): VHDL, Verilog, SystemC, System Verilog и др. При этом объем HDL-кода для сложных устройств может достигать нескольких сотен тысяч строк.

Для повышения успешности реализации таких проектов применяются современные парадигмы коллективного проектирования, повторного использования кода, паттернов и др. Большое внимание уделяется инструментам организации среды взаимодействия для совместной работы, используются системы автоматизации процессов управления проектами в соответствии с требованиями PMI (Project Management Institute) и стандартами ISO. Ведутся работы по созданию специализированных сред проектирования, позволяющих проводить распараллеливание проектных процедур, организовывать взаимодействие между проектировщиками путем обмена данными и сообщениями в режиме online, в короткие сроки выполнить прототипирование и др.

Российскими учеными в этой области являются Камаев В.А., Глушань В.М., Лаврик П.В., Анисимов В.И., Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А., Ильичев Н.Б., Пантелеев Е.Р., Пекунов В.В., Целищев Е.С., Адамов А.З., зарубежными – Indrusiak L.S., Udwardia F., Törlind P. и др.

В основе организации и использования большинства HDL-языков лежит объектно-ориентированный подход. Такие языки удобно использовать как на этапах проектирования, так и верификации проекта, а созданные проектные решения могут быть использованы в течение длительного промежутка времени без внесения изменений в код. Язык описания аппаратуры VHDL, обладая возможностью структурно-функционального представления проектируемого устройства, служит основой для формирования структурно-функциональных лингвистических моделей (СФЛМ), которые являются методологической и практической основой проектирования сложных вычислительных устройств. Под СФЛМ понимаются объекты, представленные на языке VHDL, состоящие из структурной и функциональной частей. Использование СФЛМ обеспечивает решение задач проектирования вычислительных устройств по разработке интерфейсной части устройств, структуры и функционирования устройств, а также верификации.

Основными трудностями в процессе коллективного распределенного проектирования сложных HDL-проектов являются организация эффективного взаимодействия проектировщиков на основе современных web-технологий при решении взаимосвязанных задач проекта, управление задачами проекта, накопление базы проектных решений для ее последующего эффективного использования.

В современных системах коллективной разработки сложных программных продуктов (например, GIT, SVN, Mercurial и т.п.) используются механизмы сохранения этапов разработки, создания веток для работы над отдельными задачами с последующим слиянием в автоматическом/пользовательском режиме, ведения версий проекта с возможностью возврата к предыдущим версиям. При этом для управления задачами используются ручное управление потоком задач через встроенный редактор и/или управление задачами по шаблонам бизнес-процессов в системах управления проектами. Практическое кодирование производится в системах другого типа, что затрудняет «привязку» задач к разработанному коду.

Указанные системы контроля версий, работая только с текстом, не могут в полной мере учесть специфику сложных HDL-проектов, заключающуюся в проектировании и верификации единой сущности «структура устройства + функционирование устройства». HDL-языки предназначены для сквозного функционально-логического проектирования, поддерживают различные уровни абстракции проекта, включая поддержку особенностей архитектуры (architecture-specific design), являясь основным средством проектирования, моделирования и документирования от уровня вентилях до уровня цифровых систем.

Как правило, повторное использование кода при проектировании вычислительных устройств предполагает его частичное изменение, а не просто использование по принципу «As is» («Как есть»). Процесс написания reusable (многократно используемого) кода очень важен. Использование готового кода, который является работоспособным и был тщательно протестирован, оказывается экономически более выгодным в 90% случаев для компаний производителей. В случае использования готового модуля сторонний разработчик может вообще не знать о внутреннем устройстве, при этом ему важен лишь интерфейс и принцип работы.

Таким образом, в области автоматизированного проектирования сложных HDL-проектов **актуальной и имеющей большое практическое значение научно-технической задачей** является разработка методов и средств распределенного проектирования СФЛМ, позволяющих организовать взаимодействие между проектировщиками, которые находятся на удаленном

расстоянии друг от друга, для решения проектных задач и получения проектных решений.

Разрабатываемые методы должны обеспечить возможность распределенного формирования проектных решений коллективом проектировщиков путем их создания как с «нуля» в виде VHDL-кода, так и из шаблонов СФЛМ с частичной доработкой. В процессе формирования проектного решения применение методов управления проектными задачами, поиска и синтеза проектных решений должны способствовать оптимизации управления сложными процессами проектирования VHDL-программ, сократить затраты разработки и повысить качество проектирования.

Целью диссертационной работы являются сокращение затрат на разработку и повышение качества формирования проектных решений путем реализации методов и средств коллективного распределенного проектирования, позволяющих управлять процессом проектирования и создавать базу проектных решений, элементами которой являются СФЛМ для их повторного использования.

В соответствии с поставленной целью в работе формулируются и решаются следующие **задачи исследования**.

1. Анализ методологий, методов и средств построения распределенных и многоагентных систем коллективного проектирования.

2. Разработка многоагентной системы коллективного распределенного проектирования на основе СФЛМ.

3. Разработка модели системы распределенного проектирования (СРП) СФЛМ на базе цветных сетей Петри и проведение анализа на ее основе с целью верификации СРП, проверки сценарного взаимодействия агентов СРП.

4. Разработка методов управления проектными задачами и процессом получения проектных решений.

5. Разработка методов формирования, синтеза и поиска СФЛМ в СРП.

6. Разработка программно-информационного обеспечения СРП.

Объектом исследования является автоматизация процесса формирования проектного решения при распределенной работе нескольких проектировщиков с проектными задачами.

Предметом исследования являются модели, методы и средства распределенного проектирования, используемые для организации коллективной работы и обеспечивающие автоматизацию процесса формирования проектного решения, представленного в виде СФЛМ.

Методы исследования основаны на использовании теории многоагентных систем, теории графов, теории сетей Петри, теории построения web-ориентированных САПР, теории баз данных.

Научная новизна полученных в диссертации результатов теоретических и экспериментальных исследований определяется разработанными методами и средствами, лежащими в основе организации и функционирования СРП СФЛМ. В результате исследований получены следующие результаты.

1. Предложена новая архитектура многоагентной системы СРП, включающая структуру подсистем и агентов, отличающаяся составом, типами и функциональностью агентов, и позволяющая обеспечить коллективное распределенное проектирование сложных VHDL-объектов.

2. Предложен новый метод управления задачами в СРП, отличающийся использованием разработанной ассоциативно-ориентированной параллельной сетевой схемой задач (ПССЗ) и позволяющий оптимально организовать выполнение проектных задач.

3. Предложен новый метод формирования библиотек VHDL-программ, которые позволяют многократно использовать проектные решения или модифицировать их с учетом новых задач, применяя концепцию повторного использования (Reuse). Данный метод, в отличие от существующих, позволяет наполнять библиотеку СФЛМ, которые были созданы с применением концепции MVC для разделения интерфейсов, описаний функционирования, структур проектируемых устройств и представлений.

Практическая ценность

Практическими результатами диссертационной работы являются.

1. Разработана модель архитектуры системы распределенного проектирования на базе цветной сети Петри, позволяющая провести имитационное моделирование с целью анализа и верификации нового функционала при масштабировании системы или внесении изменений в существующий функционал.

2. Разработано программное обеспечение web-ориентированной многоагентной СРП.

3. Разработан web-ориентированный транслятор, позволяющий обнаруживать ошибки в описании на VHDL при проектировании устройства и формировать СФЛМ из этого описания.

4. Разработан распределенный поиск проектных решений, представленных в виде СФЛМ.

5. Разработан аппарат управления процессом распределенного проектирования путем формирования операторов в ПССЗ.

На защиту выносятся следующие новые и содержащие элементы новизны основные результаты.

1. Организация распределенного проектирования сложных VHDL-объектов в виде многоагентной системы, содержащей 8 типов агентов с их ролевой функциональностью.

2. Механизмы управления процессом проектирования через аппарат ПССЗ.

3. Применение концепции MVC на этапе формирования проектируемого устройства из совокупности СФЛМ.

4. Разработанное программно-информационное обеспечение СРП.

Реализация и внедрение. Результаты работы внедрены в АО «Ульяновский механический завод» путем использования в корпоративной компьютерной среде обучения предприятия, в ООО «Разработка кибернетических систем» путем использования в процессе разработки проектов, а также в учебный процесс кафедры «Вычислительная техника» УлГТУ.

Апробация работы. Основные положения диссертационной работы докладывались и обсуждались на следующих Международных и Всероссийских конференциях: VIII Международной научно-практической конференции «Объектные Системы-2014», г. Шахты, 2014; VIII Международной научно-практической конференции «Системы проектирования, моделирования, подготовки производства и управление проектами CAD/CAM/CAE/PDM», г. Пенза, 2014; Всероссийской научно-технической конференции «Информатика и вычислительная техника», г. Ульяновск, 2014; Всероссийской школе-семинаре «Информатика, моделирование, автоматизация проектирования», г. Ульяновск, 2013-2014; Конгрессе по интеллектуальным системам и информационным технологиям «IS&IT'14», п. Дивноморское, 2014; XXXVII Международной научно-практической конференции «Технические науки – от теории к практике», г. Новосибирск, 2014.

Публикации. По теме диссертации опубликованы 15 печатных работ, в том числе 5 статей в изданиях, входящих в перечень ВАК РФ. Получено 1 СВИДЕТЕЛЬСТВО (РОСПАТЕНТ) об официальной регистрации программы для ЭВМ №2015610356.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав с выводами, заключения, списка литературы, изложенных на 210 страницах машинописного текста, а также приложения на 38 страницах машинописного текста, содержит 55 рисунков и 34 таблицы. Список литературы включает 157 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность исследуемой проблемы, сформулированы цель и задачи диссертационной работы, перечислены полученные в диссертации новые результаты, их научная и практическая ценность, представлены положения, выносимые на защиту, и описана структура диссертации.

В первой главе «Методологии, методы и средства распределенного проектирования СФЛМ» проводится анализ предметной области исследований. Рассмотрены ее основные понятия: «распределенное проектирование», «многоагентные системы», «структурно-функциональные лингвистические модели», а также проанализированы обобщенные подходы к проектированию, структурные элементы процесса проектирования и существующие методологии распределенного решения задач и построения многоагентных систем.

Проанализирована структура языка VHDL, описаны его основные функциональные блоки. Исследованы особенности иерархического проектирования, которое используется для больших проектов с участием коллектива разработчиков.

Рассмотрена методология «потоков работ» Workflow и ее использование для описания и реализации распределенных вычислительных процессов, в рамках которых производится координированное взаимодействие набора распределенных ресурсов.

Рассмотрена методология сопряженного проектирования, описывающая процесс параллельного проектирования аппаратных и программных средств, при котором происходит оценка целесообразности выбора аппаратной или программной реализации определенной части проекта.

Рассмотрена методология параллельного проектирования в распределенной управляемой среде разработки проектов, построенной в виде совокупности объединенных в сеть автоматизированных рабочих мест, осуществляющих доступ к общей базе проектов.

Рассмотрена методология сервис-ориентированной архитектуры, как один из подходов к созданию современных корпоративных информационных систем, представляющая собой набор бизнес-методов, организационных методов, методов управления и технических методов для создания гибкой бизнес-среды.

Рассмотрена методология агентно-ориентированного анализа и проектирования.

Исследованы основные направления разработки распределенных, многоагентных систем и описаны технологии, используемые при создании подобных систем.

Проанализированы современные, имеющие широкое применение на практике средства работы с VHDL-кодом в процессе проектирования вычислительных устройств (ВУ), их функциональные возможности с точки зрения коллективной работы, создания библиотек VHDL-программ и синтеза проектных решений. Рассмотрены организация и функционирование систем коллективного распределенного программирования GIT, SVN, Mercurial, выявлен ряд их недостатков при возможном использовании для создания HDL-проектов.

Обобщены достоинства и недостатки существующих подходов и методов коллективного проектирования при разработке сложных устройств ВУ на языках описания аппаратуры.

Сформулирована постановка задачи, в которой выделяются существующие проблемы в рассматриваемой предметной области и намечаются пути их решения в рамках диссертационной работы.

Во второй главе «Разработка многоагентной структуры системы распределенного проектирования СФЛМ» рассмотрены вопросы организации и функционирования многоагентной СРП. Представлена типовая схема процесса проектирования ВУ на HDL. Описаны этапы и задачи процесса, предложено 8 типов агентов, разработано их функциональное назначение. Определены роли и место агентов в процессе проектирования.

Общая структура многоагентной СРП представлена на рис. 1. Предложены следующие типы агентов.

Интерфейсный агент (interface agent [INA]) выполняет связующую роль агентов.

Агент управления проектными задачами (management agent project tasks [МАРТ]) выполняет формирование ПССЗ и распределение проектных задач между проектировщиками.

Агент разработки проектного решения (agent designer project solution [ADPS]) выполняет операции, связанные с созданием проектного решения на языке VHDL, т.е. формированием СФЛМ, шаблона СФЛМ и проведением лексического и семантического анализа кода.

Агент синтеза проектных решений (agent synthesis project solution [ASPS]) выполняет поиск готовых к объединению проектных решений, созданных проектировщиками, а также их синтез в единое проектное решение.

Агент маршрутизации (router agent [ROA]) выполняет связующую роль между локальными или распределенными агентами, расположенными на других серверах, которые выполняют роль хранения или разработки проектных решений.

Поисковый агент (search agent [SEA]) выполняет формирование запроса на поиск проектного решения и кластеризацию данных с целью сокращения времени поиска данных.

Агент базы данных (agent data base [ADB]) выполняет операции по работе с базой данных.

Агент рабочей памяти (agent working memory [AWM]) управляет состоянием системы и распределением нагрузки как на агентов, так и на всю систему.

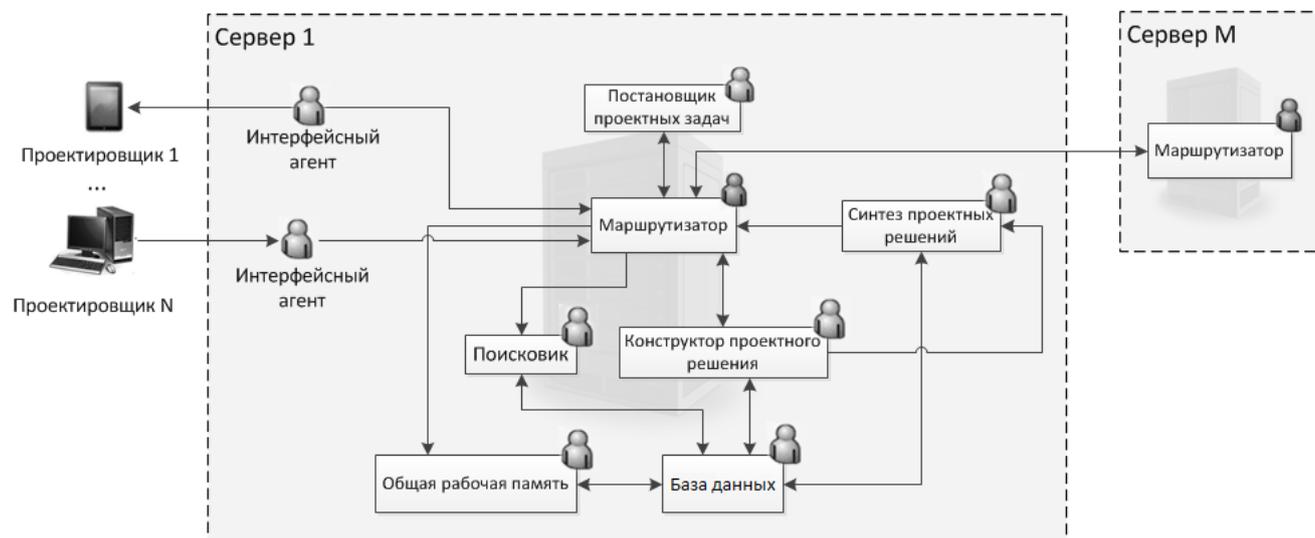


Рис. 1. Структура многоагентной СРП

Формальное определение многоагентной СРП имеет следующий вид:

$MAS = (Ag, R, En)$, где множество агентов Ag состоит из подмножеств агентов, относящихся к разным типам.

$$Ag = Ag_{INA} \cup Ag_{МАРТ} \cup Ag_{ADPS} \cup Ag_{ASPS} \cup Ag_{ROA} \cup Ag_{SEA} \cup Ag_{ADB} \cup Ag_{AWM},$$

$R = \{R_1, R_2, \dots, R_M\}$ – множество взаимодействий между агентами,

$En = \{Dev, T, SFLM, PNST\}$ – внешняя среда, $Dev = \{Dev_1, Dev_2, \dots, Dev_F\}$ – множество проектировщиков.

Работая в системе СРП, проектировщик совершает определенные действия.

$ActDev = \{ActDev_1, ActDev_2, \dots, ActDev_L\}$ – множество действий проектировщика.

$T = \{T_1, T_2, \dots, T_L\}$ – множество проектных задач в СРП,

$T_j = \{Dev, Name, Descr, Time, Prj, GitBranch\}$ – проектная задача,

$Name$ – название проектной задачи, $Descr$ – описание проектной задачи, $Time$ – время реализации проектной задачи, Prj – привязка к проекту,

$GitBranch$ – ветка разработки проектной задачи в системе контроля версий.

$SFLM = \{SFLM_1, SFLM_2, \dots, SFLM_G\}$ – множество структурно-функциональных лингвистических моделей.

$SFLM_i = \{Obj, Map\}$ – структурно-функциональная лингвистическая модель,

$Obj = \{Obj_1, Obj_2, \dots, Obj_V\}$ – множество объектов проектируемого устройства,

$Obj_j = \{Dec, Proc\}$ – объект структурно-функциональной лингвистической модели, где Dec – декларативная часть, $Proc$ – процедурная часть, Map – карта назначения входных и выходных сигналов проектируемых устройств.

Параллельной сетевой схемой задач будем называть кортеж $PNST = (G(P, A), COM, CON, F, N)$, где $G(P, A)$ – граф ПССЗ, в котором P – множество вершин-операторов, A – множество дуг. В рассматриваемом графе выделяются вершины выполнения проектных задач P^T , декомпозиции P^D и синтеза P^S , а также дуги разветвления A^C , декомпозиции A^D и синтеза A^S .

$COM = \{COM_1, COM_2, \dots, COM_X\}$ – множество составных операторов.

Отображение вершин графа на множество составных операторов имеет вид:
 $F = N \rightarrow COM$.

$COM_j = \{T, OpList_{prev}, OpList_{next}, Type, Mask\}$ – составной оператор ПССЗ.

$CON = \{CON_1, CON_2, \dots, CON_F\}$ – множество условных операторов.

Отображение дуг графа на множество условных операторов имеет вид:
 $N = A \rightarrow CON$.

$CON_j = \{T, OpList_{prev}, OpList_{next}, Type, Cond, Mask\}$ – условный оператор ПССЗ,

где $OpList_{prev} = \{Op_1, Op_2, \dots, Op_Z\}$ – список предыдущих операторов,

$OpList_{next} = \{Op_1, Op_2, \dots, Op_Z\}$ – список следующих операторов,

$Op_j \in (COM \mid CON)$,

$Type = \{task, decomposition, synthesis\}$, где $task$ – проектные задачи, $decomposition$ – декомпозиция, $synthesis$ – синтез,

$Cond$ – условие, при котором будет осуществлен переход к следующему оператору. В случае безусловного перехода $Cond = \tau$,

$Mask = \{CodeM_1, CodeM_2, \dots, CodeM_T\}$ – маска оператора, описывает атрибуты проектных задач, которые участвуют/не участвуют в поиске.

$$CodeM_i = \begin{cases} 0, & \text{не участвует в поиске} \\ 1, & \text{участвует в поиске} \end{cases}.$$

Ввиду того, что в СРП выделяется ряд модулей, не являющихся агентами, но которые предоставляют возможность взаимодействовать проектировщику с самой системой, введем в рассмотрение формальное определение СРП: $SDD = \{SysSt, In, Out, Act, DB\}$, где $SysSt = \{SysSt_1, SysSt_2, \dots, SysSt_O\}$ – множество состояний, которые характеризуют ключевые этапы получения проектных решений через модули СРП, In – запрос к СРП, Out – сформированный ответ на запрос, DB – база данных СРП, $Act = \{Act_1, Act_2, \dots, Act_P\}$ – множество внутренних действий СРП.

Математическая модель агента многоагентной системы имеет следующий вид: $Ag = \{In_{Ag}, Out_{Ag}, AgSt, ActAg\}$, где $AgSt = \{AgSt_1, AgSt_2, \dots, AgSt_K\}$ – множество возможных состояний агента, $ActAg = \{ActAg_1, ActAg_2, \dots, ActAg_R\}$ – множество действий агента.

$ActAg_i = \{name, reciever, type_mess\}$, где *name* – наименование действия, *receiver* – получатель, *type_mess* – тип отправляемого сообщения.

$In_{Ag} = \{Mes_{acl}, Query\}$ – рецепторный вход агента, который может быть в виде запроса (*Query*) или сообщения ACL (*Mes_{acl}*).

$Out_{Ag} = \{Mes_{acl}, Request\}$ – эффекторный выход агента, который может быть в виде ответа (*Request*) или сообщения ACL (*Mes_{acl}*).

Рассматривая основные сущности СРП, такие как проектировщики, компоненты СРП, агенты, база проектных решений СФЛМ и ПССЗ, выделим следующие взаимодействия между ними.

1. Между проектировщиком и компонентами СРП.

$Dev \times T \times SysSt \rightarrow ActDev \times SysSt \times [Init(Ag) = In_{Ag}]$ – обращение проектировщика к СРП, где *Init(Ag)* – функция инициализации агента путем формирования и передачи данных на его вход *In_{Ag}*.

$InData = (Find(Dev, DB) \mid Pars(In))$, где функция *Find(Dev, DB)* ищет сообщения для проектировщика в базе данных СРП, а функция *Pars(In)* обрабатывает входные данные.

$SysSt \times InData \rightarrow Act \times Out \times SysSt$ – внутрисистемное функционирование.

$SysSt \times InData \times Out \rightarrow (Provide(Out, Dev) = \{view, false\}) \times SysSt$, где функция *Provide(Out, Dev)* предоставляет данные проектировщику.

Рассматривая более детально процесс внутрисистемного функционирования по концепции MVC, предыдущая формула преобразуется следующим образом:

$SysSt \times (InData_{controller} \times InData_{method} \times InData_{data}) \rightarrow Act_{model} \times (Out_{controller} \times Out_{method} \times Out_{data}) \times SysSt$,

$SysSt \times (InData_{controller} \times InData_{method} \times InData_{data}) \times (Out_{controller} \times Out_{method} \times Out_{data}) \rightarrow (Provide(Out_{data}, Dev) = Out_{view}) \times SysSt$.

2. Между агентами, базой проектных решений и ПССЗ.

Внешняя среда агента представлена окружающими его объектами, в том числе и другими агентами. На рецепторный вход агента поступают сообщения, которые необходимо разобрать с помощью функции *Pars(In_{Ag})* и получить

необходимые данные, или агент может сформировать входные данные сам на основе текущего состояния через функцию $Create(AgSt)$.

$In_{Ag_data} = (Create(AgSt) \mid Pars(In_{Ag}))$ – формирование входных данных.

Функционирование агента описывается следующим образом:

$AgSt \times In_{Ag_data} \times [FuncDB_SFLM(In_{data}, DB_{SFLM}) = DB_{SFLM_RES}] \times$
 $\times [FuncPNST(In_{Ag_data}, PNST) = PNST_{RES}] \times [Change(In_{Ag_data}) = Data] \rightarrow$
 $\rightarrow ActAg \times Out_{Ag} \times AgSt,$

где DB_{SFLM} – база проектных решений СФЛМ, функция $Change(In_{Ag_data})$ преобразует входные данные в соответствии с внутренним состоянием агента, $FuncDB_SFLM(In_{data}, DB_{SFLM})$ представляет собой совокупность функций, название и результат работы (DB_{SFLM_RES}) которых представлены в табл. 1, $FuncPNST(In_{Ag_data}, PNST)$ представляет собой совокупность функций, название и результат работы ($PNST_{RES}$) которых представлены в табл. 2.

Таблица 1. Описание функций по работе с СФЛМ и их результатов

$FuncDB_SFLM$	DB_{SFLM_RES}
Create	{true, false}
Find	{SFLM, false}
Update	{true, false}
Delete	{true, false}

Таблица 2. Описание функций по работе с ПССЗ и их результатов

$FuncPNST$	$PNST_{RES}$
CreateOperator	{true, false}
FindTask	{Op, false}
FindPNST	{PNST, false}
FindDev	{Dev, false}
UpdateOperator	{true, false}
DeleteOperator	{true, false}

Опишем возможные данные на рецепторных входах агентов.

1. В случае когда на вход агента поступает запрос:

$In_{Ag_data} : \{uid, action, controller, content\}$, где uid – id пользователя, $action$ – действие, $controller$ – контроллер, $content$ – содержимое.

2. В случае когда на вход агента поступает aсl сообщение:

$In_{Ag_data} : \{uid, sender, reciever, content, reply_with, reply_to, envelope, language, ontology, reply_by, protocol, conversation_id\}$,

где *uid* – идентификатор пользователя в системе; *sender* – идентификатор агента-отправителя сообщения; *receiver* – кортеж идентификаторов агентов-получателей; *content* – содержимое сообщения или объект действия; *reply_with* – выражение, которое будет использоваться агентом, чтобы определить исходное сообщение при ответе на полученное сообщение; *reply_to* – выражение, ссылающееся на раннее действие, к которому текущее сообщение является ответом; *envelope* – конверт, содержащий полезную информацию о сообщении, представляет собой список пар значений ключевых слов; *language* – обозначает кодировку содержимого; *ontology* – онтология, которая используется, чтобы дать смысл содержимому; *reply_by* – крайний срок отправки ответа на сообщение в виде времени и/или даты; *protocol* – идентификатор протокола, который использует агент-отправитель (протокол предназначен для формирования дополнительного контекста, служащего для интерпретации сообщения); *conversation_id* – выражение для идентификации последовательности коммуникативных актов, которые вместе образуют «разговор»

3. Между агентом и СРП:

$$AgSt \times In_{Ag_data} \times Out_{Ag} \rightarrow (SaveDB(Out_{Ag}, DB) = \{true, false\}) \times AgSt,$$

где $SaveDB(Out_{Ag}, DB)$ – функция сохранения в базе СРП данных, полученных агентом, которые необходимо предоставить проектировщику.

На основе формульных зависимостей разработана структура моделей в виде цветной сети Петри и назначения ее позиций и переходов.

Процесс получения проектного решения, начиная с формирования проектной задачи, описывается совокупностью сценариев: вход в СРП, поиск проектных задач, поиск проектировщика, поиск проектного решения, создание и верификация проектного решения, формирование модели СФЛМ, интеграция проектных решений от проектировщиков, информирование проектировщиков. Сценарии представлены в виде моделей цветной сети Петри.

В СРП возникает задача управления потоками проектных работ. Ниже предлагается ассоциативно-ориентированная модель управления задачами (работами), позволяющая эффективно организовать коллективное проектирование VHDL-объектов.

В ПССЗ составной оператор проектных задач представляет собой совокупность задач, выполнение которых, происходит последовательно. Составной оператор синтеза представляет собой совокупность проектных решений, являющихся атрибутами в операторах, и оператора синтеза, выполнение которого приводит к формированию нового проектного решения и разрешает переход к выполнению последующих задач только тогда, когда все предыдущие будут выполнены. Составной оператор декомпозиции представляет собой

совокупность проектных задач и оператора декомпозиции, выполнение которого приводит к полуавтоматическому разбиению задачи на подзадачи. Такие задачи могут выполняться независимо друг от друга с произвольным сдвигом во времени. Обобщенный условный оператор выполнения проектных задач представляет собой совокупность условий и маски.

Таким образом, передача управления в ПССЗ может быть организована одновременно по многим выходным дугам операторной вершины декомпозиции с последующим появлением условий инициирования процесса выполнения составного оператора задач. При наличии передачи управления одновременно по нескольким входным дугам операторной вершины синтеза также запускается процесс выполнения составного оператора задач. На рис. 2 приведен пример ПССЗ.

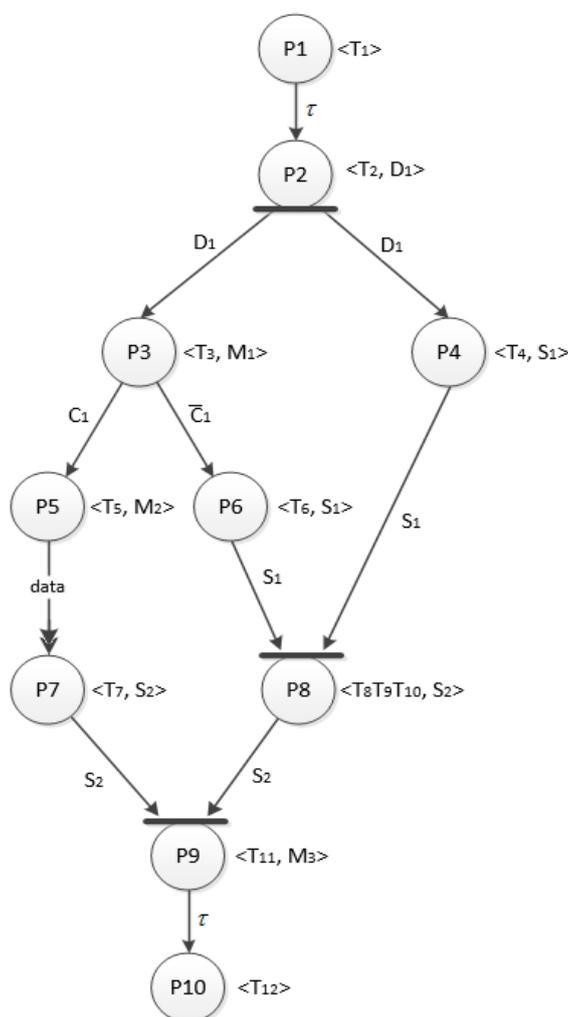


Рис. 2. Пример графа ПССЗ

При распределенном проектировании важная роль отводится процессу формирования и назначения проектных задач. Аппарат ПССЗ позволяет классифицировать множество проектных задач, описывающих проектные ситуации по контексту и признакам. Такая возможность позволяет

оптимизировать управление сложными процессами проектирования VHDL-программ, сократить затраты на разработку и повысить качество проектирования.

Во второй главе также проводится моделирование работы СРП, рассматривается назначение и практическое применение модели в виде цветной сети Петри при проектировании и анализе системы.

Такое применение рассматривается по двум направлениям: использование модели в качестве инструмента анализа до разработки программного обеспечения с целью проверки работоспособности, выявления ошибок и узких мест в системе, из-за которых производительность или пропускная способность системы ограничена одним или несколькими компонентами или ресурсами; использование возможностей, которые предоставляет анализ модели во время функционирования системы. Цветные сети Петри позволяют однозначно идентифицировать проектировщика в системе. Решение задачи достижимости позволяет определить доступные состояния системы из текущего состояния. Типовыми проектными ситуациями при работе с системой являются: поиск проектного решения, генерирование VHDL-кода, генерирование графического представления, генерирование СФЛМ, генерирование шаблона СФЛМ и др.

Результат анализа свойства живости показывает степень работоспособности системы, помогает выявлять невозможные состояния (в частности, неисполняемые ветви функционирования системы), например, ситуации, в которых блокируются действия пользователя на одной странице в web-системе, или работа в которых не может завершиться. В цветной сети Петри анализ этого свойства проводится относительно каждого цвета (т.е. относительно каждого проектировщика, находящегося в системе). Таким образом осуществляется проверка работы в системе нескольких проектировщиков, выполняющих различные действия.

Модель системы на базе аппарата сетей Петри реализована в среде CPN Tools, которая позволяет смоделировать и проанализировать работу системы с учетом ее архитектуры, выявить наиболее загруженные элементы системы. Сетевая модель СРП показана на рис. 3.

В системе выделены следующие блоки, моделирующие процесс работы проектировщиков в системе: system entry (SEN) – система входа; system search (SSE) – система поиска; system constructor project solution (SCPS) – система конструктора проектного решения; system personal cabinet (SPC) – система личного кабинета; system managing projects (SMP) – система управления проектами; system managing project tasks (SMPT) – система управления проектными задачами; messenger (MES) – система уведомлений.

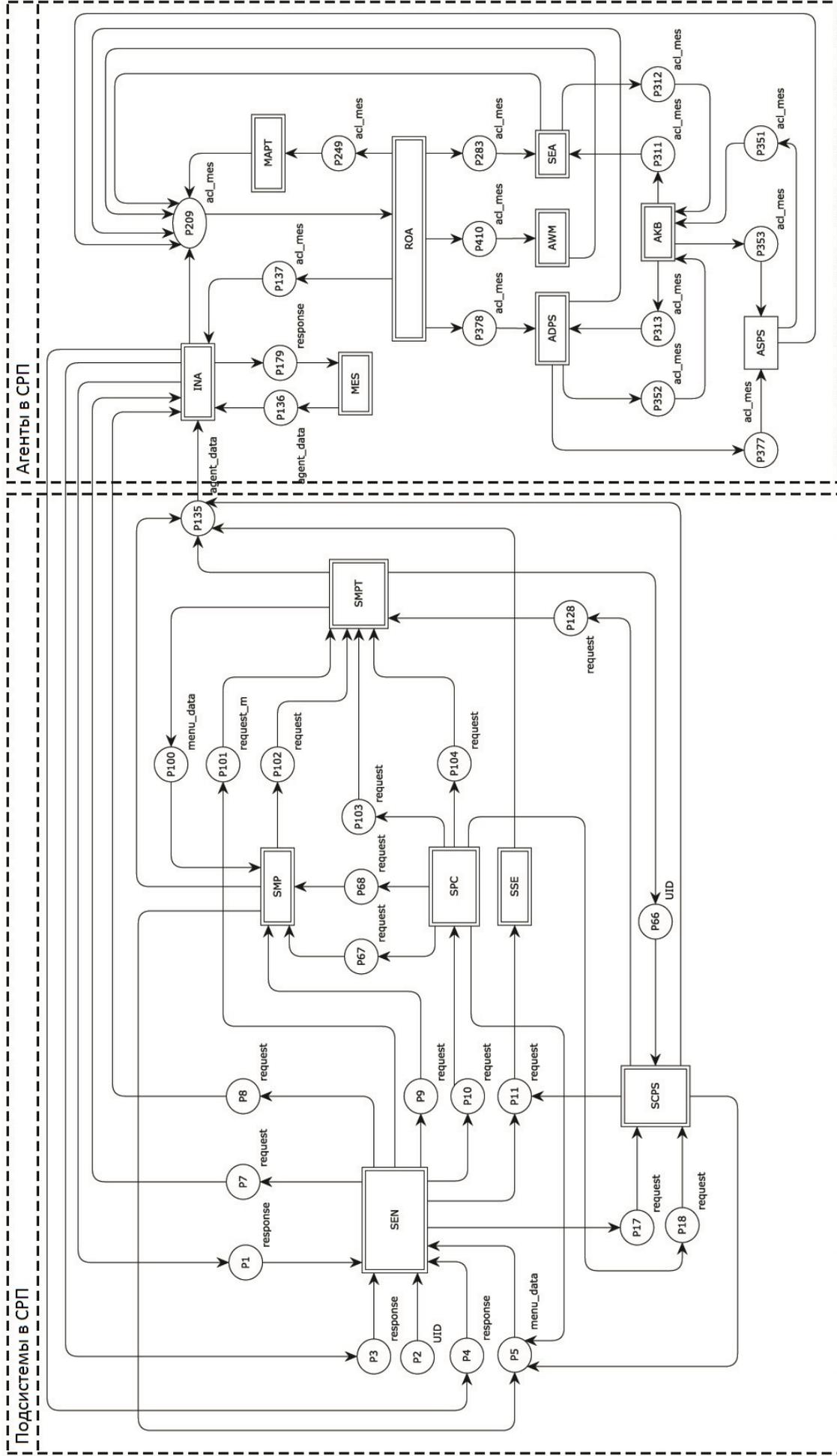


Рис. 3. Модель СРП на базе сети Петри

При моделировании работы системы задача достижимости интерпретируется как возможность перехода к некоторой типовой ситуации, под которой понимается совокупность последовательных маркировок, необходимых для перехода в интересующее состояние системы. Для этого строится дерево типовых ситуаций, переходы по которому переводят систему в выбранные состояния, минуя определенное количество промежуточных этапов. Используя свойство достижимости сети Петри, строится карта внутрисистемных переходов. На ней отображаются все переходы, но активны будут только те, в которые можно перейти из текущего состояния.

Определение свойства безопасности позволяет анализировать процесс создания проектного решения «наблюдением» за ходом выполнения проектных задач, закрепленных за проектировщиками. Так как в текущий момент у проектировщика может быть только одна активная задача, которую он делает и по которой ведется учет времени, анализ свойства безопасности позиций токенов определенного цвета позволяет выявить временные нарушения решения задач.

Обязательное наличие позиции-очереди у агента создает необходимость проверять данную позицию на свойство ограниченности. В случае превышения предельно допустимого количества задач, установленного в системе, запускается механизм создания копии агента и переназначения на него части задач. Освобождение ресурсов удалением копии агента из системы происходит при отсутствии задач в позиции-очереди агента.

В третьей главе «Разработка методов синтеза и анализа распределенного проектирования СФЛМ» исследуются методы и организация работы с СФЛМ в ходе распределенного проектирования. Предлагается схема применения концепции MVC для организации СФЛМ в виде трех отдельных компонентов таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Описывается функциональное назначение каждого компонента и приводится пример СФЛМ. Процесс работы в системе с СФЛМ по концепции MVC представлен на рис. 4.

В контроллер (SFLMController) выносятся методы для получения данных интерфейса, функциональной части и структуры из модели (SFLM). Также контроллер отвечает за получение данных о проектных задачах из ПССЗ (Parallel network scheme tasks – PNST). В качестве поисковых параметров рассматриваются: идентификаторы веток разработки (ids_pnst_branch); идентификаторы операторов ПССЗ (ids_pnst_operator); идентификаторы проектировщиков, которые решают проектные задачи (ids_designer); временные рамки разработки (start_developing, end_develop); стадии проектирования, такие как проектная задача, проектное решение в виде описания на языке VHDL, СФЛМ, шаблон СФЛМ, графическое представление проектного решения

(stages_development). Представление (SFLMView) отвечает за формирование вывода проектных решений.

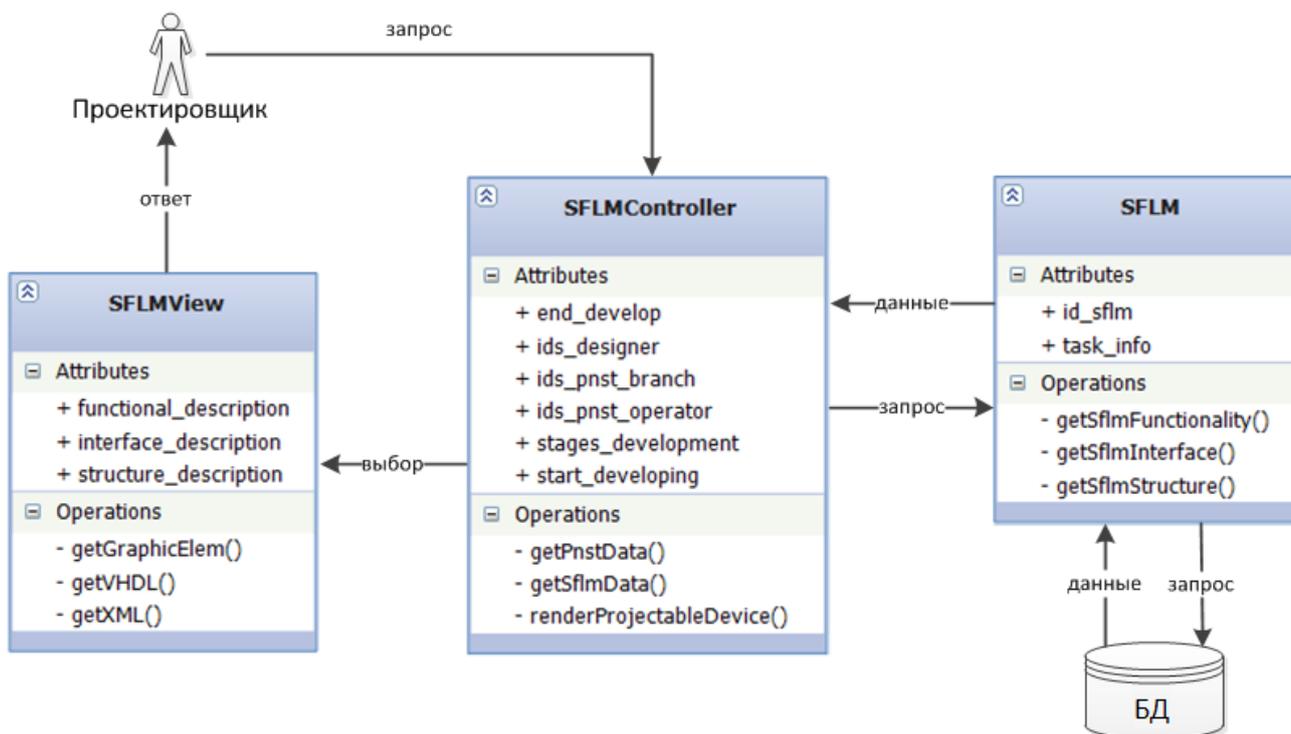


Рис. 4. Процесс работы с СФЛМ по концепции MVC

Исследованы способы представления проектных решений, такие как описания на HDL-языке, шаблоны, библиотеки элементов. Сформулированы их достоинства и недостатки.

Описаны сценарии поиска и создания операторов ПССЗ, назначение проектных задач проектировщикам, а также поиск проектных решений СФЛМ в базе данных СРП.

Рассмотрены способы распределенного проектирования СФЛМ, позволяющие организовать взаимодействие между проектировщиками, с целью получения проектных решений.

Проведен расчет коэффициента повторяемости кода при реализации HDL-проектов на основе СФЛМ. В среднем коэффициент повторяемости составил 25-30%, что свидетельствует о сокращении сроков проектирования.

Получена оценка эффективности применения СФЛМ, как средства формирования проектного решения, в виде оценки трудозатрат по проектированию ВУ на языке описания аппаратуры. Анализ проведен на основе метрики SLOC (Source Lines of Code). Получена оценка сокращения времени проектирования в среднем на 10% при использовании СРП и СФЛМ.

В четвертой главе «Реализация многоагентной системы распределенного проектирования» рассмотрена авторская разработка многоагентной web-ориентированной СРП SFLM CAD, в которой операции синтеза и анализа проектных решений возлагаются на агентов. Описано информационное обеспечение системы, включающее в себя внутрисистемное и внесистемное информационное обеспечение.

В составе внесистемного информационного обеспечения выделяются поставщики данных, являющиеся информационными системами, которые собирают проектные решения в виде описания устройств на языке VHDL.

В составе внутрисистемного информационного обеспечения выделяются следующие компоненты: сборщик данных, осуществляющий сборку проектных решений в виде VHDL-кода от поставщиков данных с последующим преобразованием их в СФЛМ и сохранением в базе данных; хранилище данных – предназначено для хранения данных системы, проектных решений, проектных задач, данных агентов и обмена информацией между агентами.

Описаны структура клиентской и серверной части системы, основанной на шаблоне MVC, а также базы данных. Рассмотрена ролевая политика, являющаяся следствием создания команды разработчиков. Приведено описание построения каждой подсистемы и ее функционального назначения.

В качестве примера реализации отдельных модулей web-системы приведен транслятор из VHDL-кода в СФЛМ. Приведены разработанные правила формирования СФЛМ и операторов ПССЗ.

Разработана интегрированная система проектирования и обучения, включающая СРП и систему электронного обучения автоматизированному проектированию ВУ по методологии СФЛМ. Отличительной особенностью этой системы является использование общей базы проектных решений СФЛМ. Система обучения построена на платформе Moodle. Включение проектных решений в виде СФЛМ в процесс обучения выполняет специализированный агент, предоставляющий необходимые данные по ходу индивидуальной траектории обучения. Взаимодействие системы обучения и СРП организовано в виде связи агентов интеграции. Описан API (application programming interface) системы электронного обучения, включающий функции получения обучающих материалов и проектных решений СФЛМ по метаданным.

В заключительной части главы рассмотрен процесс распределенного проектирования в созданной САПР.

В заключении сформулированы основные научные и практические результаты, полученные в результате проведенных в диссертационной работе исследований.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Предложена и реализована архитектура web-ориентированной многоагентной системы распределенного автоматизированного проектирования. Доступ к функционалу системы осуществляется посредством web-интерфейса с возможностью удаленной работы над проектируемым объектом. Сокращение времени проектирования в среднем составляет 10%.

2. Предложены методы управления задачами в системе распределенного проектирования и формирования библиотек VHDL-программ, позволяющие повысить качество формирования проектных решений за счет эффективной реализации потоков проектных задач, а также наполнения базы проектных решений объектами в виде СФЛМ, поиска наиболее близких проектных решений и их повторного использования.

3. На основе разработанной модели СРП в виде сети Петри получены практические результаты анализа работы СРП и функционирования агентов.

4. Разработано программно-информационное обеспечение СРП.

Публикации в журналах, входящих в список ВАК

1. Афанасьев А.Н. Система автоматизированного проектирования структурно-функциональных лингвистических моделей / А.Н. Афанасьев, В.С. Хородов // Автоматизация процессов управления. – 2014. – №4(38). – С. 92-98.

2. Афанасьев А.Н. Агентный подход к построению системы распределенного проектирования VHDL-объектов и ее моделирование на базе цветных сетей Петри / А.Н. Афанасьев, В.С. Хородов // Известия Волгоградского государственного технического университета. – 2014. Т.22. – №25(152). – С. 47-53.

3. Афанасьев А.Н. Технологии распределенного проектирования VHDL-объектов / А.Н. Афанасьев, В.С. Хородов // Вестник Ижевского государственного технического университета. – 2014. – №4 (64). – С. 131-134.

4. Afanasev A.N. Software system of distributed design of complex VHDL objects / A.N. Afanasev, V.S. Khorodov // Software & Systems. – 2015. – №1 (109). – pp. 42-47.

5. Афанасьев А.Н. Разработка и моделирование распределенной системы проектирования VHDL-объектов / А.Н. Афанасьев, В.С. Хородов // Автоматизация. Современные технологии. – 2015. – №4. – С. 34-40.

Свидетельства о регистрации ПО для ЭВМ

6. Афанасьев А.Н. Web-ориентированный транслятор VHDL описаний в шаблоны структурно-функциональных лингвистических моделей / А.Н. Афанасьев, В.С. Хородов // Свидетельство о государственной регистрации программы для ЭВМ № 2015610356 от 12 января 2015 г.

Другие публикации

7. Хородов В.С. WEB-ориентированный подход к построению САПР / В.С. Хородов // Информатика, моделирование, автоматизация проектирования: сборник научных трудов Всероссийской школы-семинара. – Ульяновск: УлГТУ, 2013. – С. 184-190.

8. Хородов В.С. Технологии распределенного проектирования / В.С. Хородов, А.Г. Игонин // Вестник Ульяновского государственного технического университета. – 2014. – №1. – С. 55-59.

9. Афанасьев А.Н. Распределенное проектирование структурно-функциональных моделей, представленных на языке VHDL / А.Н. Афанасьев, В.С. Хородов // Вестник Ульяновского государственного технического университета. – 2014. – №2. – С. 41-45.

10. Хородов В.С. Проектирование многоагентной распределенной системы создания проектных решений с использованием структурно-функциональных лингвистических моделей / В.С. Хородов // Объектные системы: материалы VIII Международной научно-практической конференции. – Ростов-на-Дону: ШИ (ф) ЮРГПУ (НПИ) им. М.И. Платова, 2014. – С. 19-23.

11. Хородов В.С. Разработка методов синтеза распределенного проектирования структурно-функциональных лингвистических моделей / В.С. Хородов // Системы проектирования, моделирования, подготовки производства и управление проектами CAD/CAM/CAE/PDM: сборник статей по материалам VIII Международной научно-практической конференции. – Пенза: Приволжский Дом Знаний, 2014. – С. 83-86.

12. Афанасьев А.Н. Моделирование распределенной системы проектирования VHDL-объектов / А.Н. Афанасьев, В.С. Хородов // Информатика и вычислительная техника: сборник научных трудов Всероссийской научно-технической конференции. – Ульяновск: УлГТУ, 2014. – С. 21-33.

13. Афанасьев А.Н. Модель управления потоками работ в системе распределенного проектирования VHDL-объектов / А.Н. Афанасьев, В.С. Хородов // Технические науки – от теории к практике: сборник статей по

материалам XXXVII Международной научно-практической конференции. №8(33). – Новосибирск: СибАК, 2014. – С. 6-11.

14. Афанасьев А.Н. Распределенное проектирование структурно-функциональных лингвистических моделей / А.Н. Афанасьев, В.С. Хородов // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'14». – М.: Физматлит, 2014. – Т. 1. – С. 331-337.

15. Хородов В.С. Разработка методов и средств многоагентного распределенного автоматизированного проектирования структурно-функциональных лингвистических моделей вычислительных устройств / В.С. Хородов // Информатика, моделирование, автоматизация проектирования: сборник научных трудов Всероссийской школы-семинара. – Ульяновск: УлГТУ, 2014. – С. 191-197.

16. Афанасьев А.Н. Разработка модели управления задачами в системе распределенного проектирования VHDL-программ / А.Н. Афанасьев, В.С. Хородов // Вестник Ульяновского государственного технического университета. – 2015. – №2. – С. 33-38.

Принятые сокращения и обозначения

ACL – Agent Communication Language

API – Application Programming Interface

CPN – Coloured Petri Nets

HDL – Hardware Description Language

ISO – International Organization for Standardization

MVC – Model View Controller

PMI – Project Management Institute

VHDL – Very high speed integrated circuits Hardware Description Language

БД – база данных

ВУ – вычислительные устройства

ПССЗ – параллельная сетевая схема задач

САПР – система автоматизированного проектирования

СБИС – сверхбольшая интегральная схема

СРП – система распределенного проектирования

СФЛМ – структурно-функциональная лингвистическая модель

Хородов Виталий Сергеевич

Разработка методов и средств многоагентного распределенного
автоматизированного проектирования структурно-функциональных
лингвистических моделей вычислительных устройств

Автореферат

Подписано в печать 06.07.2015. Формат 60×84/16.

Усл. печ. л. 1,4. Тираж 100 экз. Заказ 599.

ИПК «Венец» УлГТУ, 432027, г. Ульяновск, Северный Венец, д. 32.