

COMPUTER-AIDED ENGINEERING

СИСТЕМЫ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

УДК 004.8.681.3.06

А.Н. Афанасьев, В.С. Хородов

СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ СТРУКТУРНО-ФУНКЦИОНАЛЬНЫХ ЛИНГВИСТИЧЕСКИХ МОДЕЛЕЙ

Афанасьев Александр Николаевич, доктор технических наук, окончил радиотехнический факультет Ульяновского политехнического института, проректор по дистанционному и дополнительному образованию в Ульяновском государственном техническом университете. Имеет книги и статьи в области автоматизированного проектирования программно-аппаратного обеспечения ЭВМ. [e-mail: a.afanasev@ulstu.ru].

Хородов Виталий Сергеевич, аспирант, окончил факультет информационных систем и технологий УлГТУ. Руководитель отдела веб-разработки в компании ООО «Разработка кибернетических систем». Имеет статьи в области распределенного проектирования и многоагентных систем. [e-mail: v.khorodov73@gmail.com].

Аннотация

В данной статье описана система распределенного проектирования, в основе которой лежит многоагентный подход. Представлена архитектура разработанной системы, которая описывает место и роль как системы, так и ее основных подсистем в процессе распределенного проектирования. Описаны типы взаимодействия и функциональное назначение агентов в системе. Рассмотрена структурно-функциональная лингвистическая модель, формирующаяся из описания проектируемого устройства на языке VHDL и являющаяся одним из основных объектов, с которыми взаимодействуют агенты и проектировщики. Описаны основные сценарии взаимодействия между агентами и графовое представление такого сценария. Показаны основные технологии, играющие важную роль в процессе функционирования системы распределенного проектирования. Представлен состав и описание функционального назначения компонентов внутри-системного и внесистемного информационного обеспечения разработанной системы. Предлагаемое решение в виде системы распределенного проектирования направлено на повышение качества и сокращение затрат в ходе создания проектных решений.

Ключевые слова: САПР, распределенное проектирование, многоагентная система, VHDL, структурно-функциональная лингвистическая модель, CORBA.

COMPUTER-AIDED DESIGN SYSTEM OF STRUCTURAL AND FUNCTIONAL LINGUISTIC MODELS

Alexander Nikolaevich Afanasiev, Doctor of Engineering; graduated from the Faculty of Radio Engineering of Ulyanovsk Polytechnic Institute; Vice-Rector for Distance and Additional Education in Ulyanovsk State Technical University, an author of books and articles in the field of computer-aided design of soft-hardware. e-mail: a.afanasev@ulstu.ru.

Vitaliy Sergeevich Khorodov, Post-Graduate Student; graduated from the Faculty of Information Systems and Technologies of Ulyanovsk State Technical University; Head of the Department for Web-Development in 'Cyber

Systems Development' LLC; an author of articles in the field of distributed design and multi-agent systems. e-mail: v.khorodov73@gmail.com.

Abstract

The article describes a distributed design system, which is based on the multi-agent approach. A developed system architecture that describes the place and role of both system and its main subsystems in the distributed design process is presented. The interaction modes and the functionality of agents in the system are described. A structural and functional linguistic model generated from a designed device description by means of the VHDL language, one of the main objects with which agents and designers interact is considered. The main scenarios of interaction between the agents and a graph representation of such scenario are described. Basic technologies playing an important role in the process of the distributed-design system functioning are shown. A composition and a description of the internal and external dataware component functionality of the developed system are represented. A proposed solution in a form of a distributed design system is aimed at the quality improvement and the reduction of costs during a design solution creation.

Key words: CAD, distributed design, multi-agent systems, VHDL, structural and functional linguistic models, CORBA.

ВВЕДЕНИЕ

Рост сложности технических объектов типа «система на кристалле», к которым относятся современные сверхбольшие интегральные схемы (СБИС), обуславливает поиск новых подходов к проектированию продуктов, обеспечивающих требуемое качество проекта при сокращении затрат. Методологическую, технологическую и инструментальную основу создания СБИС составляют в настоящее время высокоуровневые языки VHDL, Verilog и др. Организация коллективных работ по проектированию сложных VHDL-программ может быть реализована на базе распределенной системы, в основу функционирования которой положены Internet-технологии.

Авторами предлагается система распределенного автоматизированного проектирования (СРАП) объектно-ориентированных моделей, представленных на языке VHDL. Такие структурно-функциональные лингвистические модели (СФЛМ) описывают структурные и функцио-

нальные части объекта [1]. Процедура синтеза включает обработку языкового описания функциональной модели на основе грамматики языка, при этом структурно-функциональное представление рассматривается как совокупность связанных функциональных объектов [2].

В основе методологии распределенного проектирования СФЛМ лежит принцип повторного использования шаблонов СФЛМ, которые были сохранены в базе знаний.

АРХИТЕКТУРА

Обобщенная структура СРАП СФЛМ представлена на рисунке 1. В ее основе лежит технология CORBA (Common Object Request Broker Architecture), позволяющая организовать единую информационную среду, компоненты которой могут взаимодействовать друг с другом вне зависимости от их конкретной реализации. На основе формирования промежуточного программного обеспечения (ППО, middleware) производится связывание програм-

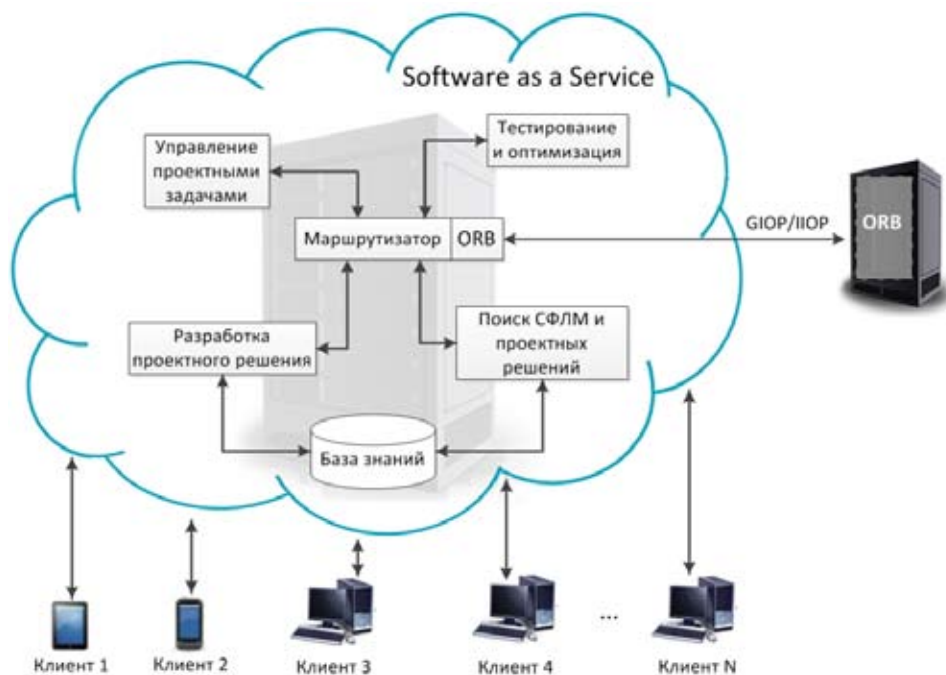


Рис. 1. Структура системы распределенного проектирования СФЛМ

мных приложений, которые находятся на разных серверах. Организация мостов между приложениями производится с помощью технологии брокера объектных запросов (ORB, Object Request Broker) [3].

В рассматриваемой архитектуре используется частное облако как средство организации процесса распределенного проектирования с централизованным хранением проектных решений. Под частным облаком будем понимать инфраструктуру, предназначенную для использования одной организацией, включающей несколько подразделений.

В качестве функциональных модулей выделены следующие подсистемы: управления проектными задачами, маршрутизации, разработки проектного решения, поиска шаблона СФЛМ, тестирования и оптимизации [4]. На рисунке 1 показаны связи между подсистемами и серверами с указанием протокола взаимодействия.

МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Математическое обеспечение СРАП СФЛМ основано на применении многоагентных систем.

В [5] рассматриваются следующие типы агентов: рефлексный агент; рефлексный агент, основанный на модели; агент, основанный на цели; агент, основанный на полезности.

В СРАП к рефлексным агентам относятся интерфейсный агент и агент базы знаний; к рефлексным агентам, основанным на модели, – агент рабочей памяти; к агентам, основанным на полезности, – агент управления проектными задачами, поисковый агент и агент разработки проектного решения; к агентам, основанным на цели, – агент синтеза проектных решений и агент маршрутизации.

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

Агент управления проектными задачами (Design Tasks Management Agent [DTMA]) выполняет формирование параллельной сетевой схемы задач (ПССЗ) и распределение проектных задач между проектировщиками.

Агент разработки проектного решения (Design Development Agent [DDA]) 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 (Design Synthesis Agent [DSA]) 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 знаний (Knowledge Base Agent [KBA]) выполняет операции по работе с базой знаний.

Агент рабочей памяти (Working Memory Agent [WMA]) управляет состоянием системы и распределением нагрузки как на агентов, так и на всю систему.

В СРАП для управления потоками задач использована ассоциативно-ориентированная модель управления потоками работ [6] в виде ПССЗ, позволяющая эффективно организовать коллективное проектирование VHDL-объектов.

ВЗАИМОДЕЙСТВИЕ МЕЖДУ АГЕНТАМИ

В зависимости от ситуации, возникающей при работе в СРАП, в сообщениях агентов содержится различная информация, поэтому для анализа процесса выделим следующие типы взаимодействий.

1. Receive Approval (RA) – позволяет агенту принимать сообщения от другого агента.
2. Accept Requests (AR) – позволяет агенту принимать запросы от другого агента.
3. Send Approval (SA) – позволяет агенту посылать сообщения другому агенту.

Согласно спецификации FIPA (Foundation for Intelligent Physical Agents) можно выделить протоколы взаимодействия.

1. Request Interaction Protocol (IP) – протокол взаимодействия, позволяющий агенту запросить другого агента выполнить некоторое действие.
2. Query Interaction Protocol (QIP) – протокол взаимодействия, позволяющий одному агенту запросить выполнение некоторого действия на другом агенте.
3. Request When Interaction Protocol (RIP) – протокол взаимодействия, позволяющий агенту запросить выполнение некоторого действия, учитывая определенные условия.

Для наглядного представления взаимодействия агентов сформирован граф, вершинам которого соответствуют агенты, а дугам – обмен сообщениями в процессе коммуникации. Последовательность посылаемых агентами сообщений образует сценарий их взаимодействия. Например, при поиске проектного решения сценарий взаимодействия

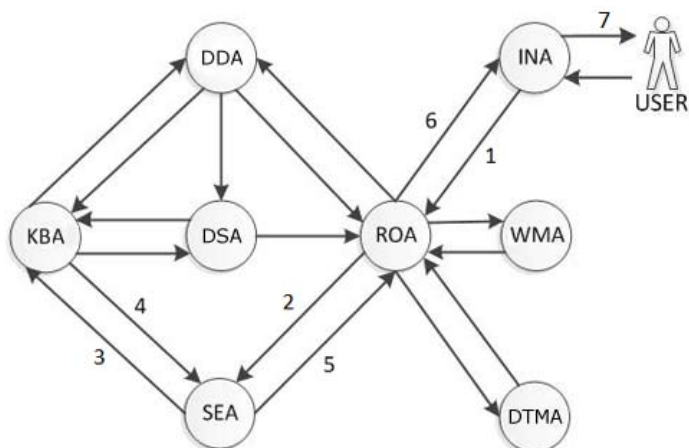


Рис. 2. Граф сценария взаимодействия агентов при поиске проектного решения

агентов может быть представлен в виде графа, изображенного на рисунке 2. Так как сценарий представляет собой последовательность сообщений, передаваемых агентами, то над дугами в графе указываются номер сообщения.

В таблице 1 представлен сценарий поиска проектного решения с указанием взаимодействий агентов, отражающих соответствия между типом сообщения, порядком его передачи и действиями агента.

Взаимодействие между агентами производится посредством формирования ACL-сообщения с последующей его отправкой. ACL-сообщение имеет вид:

```
(inform
  : sender INA
  : receiver ROA
  : content search_query_data
  : in-reply-to prev_message_id
  : ontology projectSolution
  : reply-by 29.08.2014 12:00
).
```

СТРУКТУРА СФЛМ

СФЛМ представляет собой совокупность объектов и связей между ними. Структура СФЛМ представлена на рисунке 3. В описываемой структуре встречаются квантификаторы, которые показывают, сколько раз могут встречаться элементы в описываемой структуре. Например, (*) предполагает неограниченное количество элементов, а (+) – хотя бы один элемент. Обработываемые данные хранятся во входных и выходных сигналах, ячейках памяти, сигналах связи и регистрах. При поступлении данных на входы обрабатываемого объекта в соответствии с функцией преобразования происходит формирование выходных результатов. При формировании шаблона из описанной модели СФЛМ можно управлять настраиваемыми параметрами и, таким образом, получить новое проектное решение. На основе описанной структуры организуется поиск СФЛМ по параметрам, которые являются важными атрибутами модели.

В ходе процесса проектирования формируется шаблон СФЛМ. Особенность шаблона в том, что его можно представить как заготовку, из которой создается новое проектное решение путем изменения редактируемых параметров. Размерность ячеек памяти, отвечающих за структурные элементы объектов и связи между ними, рассчитываются относительно предварительно указанных входных переменных шаблона.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Для организации взаимодействия между распределенными клиент-серверными системами проектирования VHDL-объектов применяется технология CORBA. Отличительным признаком CORBA является архитектура Object Management Architecture (OMA) [7]. В основе OMA лежит концепция брокера объектных запросов. Брокер объектных запросов (ORB) – основной компонент CORBA. Все объекты, которые используют брокер объектных запросов, могут взаимодействовать с любыми другими объектами, подключенными через брокер [3].

СРАП рассматривается как объект, который может одновременно играть роль и клиента, и сервера. Если объект является инициатором вызова метода у другого объекта, то он выполняет роль клиента. Если объект является исполнителем запрашиваемого метода, то он выполняет роль сервера. Большинство объектов одновременно исполняют роль и клиентов, и серверов, попеременно вызывая и исполняя методы. Применение технологии CORBA способствует созданию более гибких систем, чем клиент-сервер, которые основаны на двухуровневой и трехуровневой архитектурах.

В основу распределенного проектирования СФЛМ положена технология MVC (Model-View-Controller). Разделяя бизнес-логику (модель) и визуализацию (представление, вид), осуществляется принцип повторного использования. Применение подобной концепции в описываемой системе необходимо в случае просмотра проектировщиком одних проектных решений в различном визуальном представлении или на разных стадиях разработки.

Таблица 1

Описание сценария взаимодействия агентов при поиске проектного решения

Агент-отправитель (sender)	Агент-получатель (receiver)	Тип	№ сообщения	№ пред. сообщения	Описание взаимодействия
INA	ROA	IP	1	–	посылает запрос с данными для поиска проектного решения
		–	7	6	предоставление пользователю результатов поиска
ROA	INA	SA	6	5	поиск запроса, сделанного агентом, и передача ему результатов
	SEA	QIP	2	1	перенаправляет запрос с данными для поиска проектного решения
SEA	ROA	SA	5	4	передача результатов выполнения запроса
	AKB	QIP	3	2	передача сформированного поискового запроса
KBA	SEA	SA	4	3	передача результатов выполнения поискового запроса к базе знаний

В СРАП MVC разделяет данные, представление и обработку действий проектировщика на три отдельных компонента.

1. Модель, реагируя на запросы, предоставляет данные (проектные решения, задачи и т. д.) и методы для работы с данными, изменяет свое состояние.

2. Представление формирует вывод данных в различных форматах, таких как html, json, xml и т. д., в зависимости от обращения в системе проектирования.

3. Контроллер получает запросы от проектировщиков и обеспечивает соответствующую реакцию, характеризующуюся контролем входных данных, получением модели и рендером представления сформированных данных.

Применение веб-технологий, таких как php, html, javascript, позволяет сформировать веб-интерфейс. Посредством этого интерфейса проектировщики могут работать в одной системе, управляя проектными задачами и создавая проектные решения через браузер.

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

Информационное обеспечение СРАП включает в себя внутрисистемное и внесистемное информационное обеспечение.

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

В состав внутрисистемного информационного обеспечения входят следующие компоненты: сборщик дан-

ных, осуществляющий сборку проектных решений в виде VHDL-кода от поставщиков данных с последующим преобразованием их в СФЛМ и сохранением в базе знаний; хранилище данных, представляющее собой базу знаний, а также базу данных агентов в системе распределенного проектирования. Хранилище данных предназначено для хранения данных системы, проектных решений, проектных задач, данных агентов и обмена информацией между агентами.

Внутрисистемное информационное обеспечение организовано по принципу создания центрального хранилища данных. Ядром системы является база знаний, содержащая всю информацию о сформированных проектных решениях в виде СФЛМ. Каждое проектное решение в системе представлено отдельной сущностью в базе знаний, которая имеет уникальный идентификатор.

Поиск в базе знаний организуется следующим образом. Запрос формируется на ограниченном естественном языке и подвергается анализу, в рамках которого выделяются дескрипторы, присутствующие в словаре. В соответствие запросу ставится совокупность дескрипторов, которая участвует в получении релевантности, путем сопоставления с поисковыми образами. Если было установлено семантическое соответствие поискового запроса и поискового образа проектного решения, из образа извлекается идентификатор проектного решения, которое необходимо передать проектировщику, сделавшему запрос. Ответом на запрос является множество проектных решений, соответствующих отобранному в процессе поиска идентификаторам.



Рис. 3. Структура СФЛМ

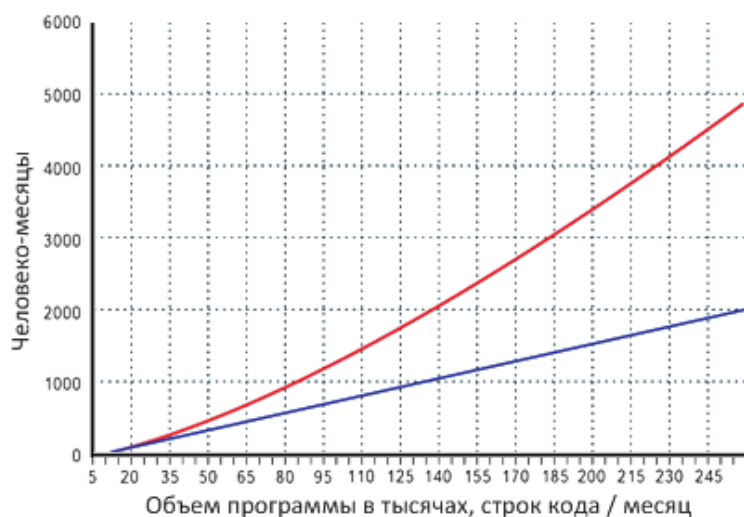


Рис. 4а. График зависимости времени разработки от объема программы

Проектные решения СФЛМ в базе знаний сохраняются в унифицированной форме. Это позволяет для проектных решений, представленных на различных языках описания, иметь единое представление в хранилище данных.

Хранение параметров и описаний СФЛМ в базе знаний позволяет реализовать автоматизированный поиск необходимой модели при помощи языка запросов SQL.

Оценки эффективности

В [1] рассматриваются четыре подхода разработки VHDL-проектов на основе: прямого программирования, программирования с подстановкой, автоматической генерации СФЛМ по шаблону, автоматической генерации СФЛМ с подстановкой. Каждый из методов направлен на повышение производительности (примерно в 1,5–2 раза) проектирования и качества получения проектного решения. Распределенное проектирование характерно, в основном, для 3 и 4 подходов.

На рисунке 4а показана зависимость объема работ по разработке программного кода на высокоуровневых языках описания аппаратных средств от количества строк кода (LOC) проекта. При этом верхняя кривая представляет собой график, построенный на основании модели издержек разработки COCOMO (Constructive Cost Model). Нижняя кривая соответствует распределенному проектированию, при котором потоки проектных работ разбиваются на модули объемом 20 тыс. строк (модель Kilo Source Lines Of Code (KSLOC)) [8].

На рисунке 4б показан качественный график зависимости производительности и качества разработки от используемых подходов.

Для оценки трудозатрат исследован проект разработки параллельного ассоциативного микропрограммного устройства управления [1]. Анализ трудозатрат производился по количеству строк кода KSLOC и длительности разработки [9]. Варьируя такие факторы, как изменение логики работы проектируемого устройства, опыта и коли-

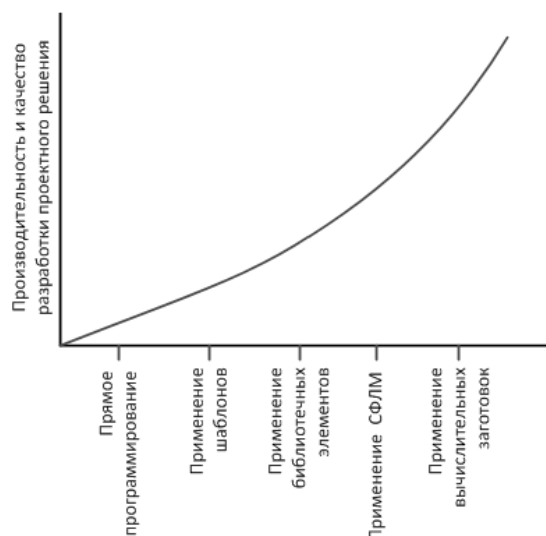


Рис. 4б. График зависимости производительности и качества разработки от применяемых подходов

чества проектировщиков, получен выигрыш во времени проектирования в 95% при внесении незначительных изменений в СФЛМ и 24% при существенной модификации логики работы проектируемого устройства.

ЗАКЛЮЧЕНИЕ

Разработанная на основе многоагентного подхода и использования базы знаний проектных решений СРАП позволяет оптимизировать управление сложными процессами проектирования VHDL-программ, сократить затраты разработки и повысить качество проектирования. Веб-интерфейс, лежащий в основе работы с СРАП, обеспечивает реализацию современного подхода взаимодействия с веб-сайтом или приложением через браузер и организацию коллективного распределенного проектирования в сетях Internet и Intranet. Повышение качества проектирования достигается за счет применения современной парадигмы повторного использования СФЛМ с целью получения нового проектного решения. Применение паттернов, представляющих собой типовые структуры СФЛМ в виде метаданных, обеспечивает более точный и быстрый поиск проектных решений. В настоящее время ведутся работы по интеграции СРАП с системой обучения автоматизированному проектированию технических объектов [10–12].

СПИСОК ЛИТЕРАТУРЫ

1. Афанасьев А.Н., Игонин А.Г. Обработка лингвистических структурно-функциональных моделей на основе нейросемантических сетей: монография. – Ульяновск: УлГТУ, 2007. – 227 с.
2. Афанасьев А.Н., Игонин А.Г. Применение нейросемантического подхода для анализа и синтеза функциональных моделей в системах проектирования // Вестник Ижевского государственного технического университета. – 2007. – № 1. – С. 66–69.
3. Хородов В.С., Игонин А.Г. Технологии распределенного проектирования // Вестник Ульяновского го-

сударственного технического университета. – 2014. – № 1. – С. 55–59.

4. Афанасьев А.Н., Хородов В.С. Распределенное проектирование структурно-функциональных моделей представленных на языке VHDL // Вестник Ульяновского государственного технического университета. – 2014. – № 2 (66). – С. 41–45.

5. Рассел С., Норвиг П. Искусственный интеллект: современный подход. – 2-е изд. – М.: Вильямс, 2006. – 1408 с.

6. Афанасьев А.Н. Методология графо-аналитического подхода к анализу и контролю потоков работ в автоматизированном проектировании сложных компьютеризованных систем // Вестник Ульяновского государственного технического университета. – 2011. – № 3 (55). – С. 48–52.

7. Малышева Е.Ю. Учебно-методический комплекс по дисциплине «Распределенные информационные системы». – Тольятти: Изд-во ПВГУС, 2013.

8. Каршенбойм И. Квадрига Аполлона и микропроцессоры // Компоненты и технологии. – 2006. – № 5. – С. 112–115.

9. Гольфанд И.Я., Хлебутин П.С. Оценка трудозатрат разработки программной компоненты // Труды ИСА РАН. – 2005. – Т. 15. – С. 125–135.

10. Афанасьев А.Н., Войт Н.Н. Разработка компонентно-сервисной платформы обучения: диаграммы использования и деятельности программного компонента сценария на UML-языке // Вестник Ульяновского государственного технического университета. – 2012. – № 1 (57). – С. 66–68.

11. Афанасьев А.Н., Войт Н.Н. Интеллектуальная обучающая система концептуальному проектированию автоматизированных систем // Известия Самарского научного центра Российской академии наук. – 2010. – Т. 12, № 4 (2). – С. 465–468.

12. Афанасьев А.Н., Войт Н.Н. Компонентная автоматизированная обучающая система САПР на основе гибридной нейронной сети // Автоматизация и современные технологии. – 2009. – № 3. – С. 14–18.

REFERENCES

1. Afanasiev A.N. and Igonin A.G. *Obработка lingvisticheskikh strukturno-funktionalnykh modeley na osnove neyrosemanticheskikh setey: monografiya* [The Processing of Structural and Functional Linguistic Models on the basis of Neurosemantic Networks: Monograph]. Ulyanovsk, UISTU Publ., 2007. 227 p.

2. Afanasiev A.N. and Igonin A.G. *Primenenie neyrosemanticheskogo podkhoda dlya analiza i sinteza funktsionalnykh modeley v sistemakh proektirovaniya* [The Application of a Neurosemantic Approach to Analyze and Synthesize the Functional Models in Design Systems]. *Vestnik Izhevskogo gosudarstvennogo tekhnicheskogo universiteta* [Bulletin of Kalashnikov Izhevsk State Technical University]. 2007, no. 1, pp. 66–69.

3. Khorodov V.S. and Igonin A.G. *Tekhnologii raspredelennogo proektirovaniya* [Distributed Design Technologies]. *Vestnik Ulyanovskogo gosudarstvennogo*

tekhnicheskogo universiteta. [Bulletin of Ulyanovsk State Technical University], 2014, no. 1, pp. 55–59.

4. Afanasiev A.N. and Khorodov V.S. *Raspredelennoe proektirovanie strukturno-funktionalnykh modeley predstavlenykh na yazyke VHDL* [The Distributed Design of Structural and Functional Models in VHDL-Language]. *Vestnik Ulyanovskogo gosudarstvennogo tekhnicheskogo universiteta* [Bulletin of Ulyanovsk State Technical University], 2014, no. 2 (66), pp. 41–45.

5. Russel S. and Norvig P. *Iskusstvennyi intellekt: sovremennyi podkhod. 2-e izd.* [Artificial Intelligence. A Modern Approach, Second Edition]. Moscow, Williams Publ., 2006. 1408 p.

6. Afanasiev A.N. *Metodologiya grafo-analiticheskogo podkhoda k analizu i kontrolyu potokov rabot v avtomatizirovannom proektirovanii slozhnykh kompyuterizovannykh sistem* [Methods of Graph-Analytic Approach to Analysis and Workflow Control in Computer-Aided Design of Complex Computer-Aided Systems]. *Vestnik Ulyanovskogo gosudarstvennogo tekhnicheskogo universiteta* [Bulletin of Ulyanovsk State Technical University], 2011, no. 3 (55), pp. 48–52.

7. Malysheva E.Yu. *Uchebno-metodicheskiy kompleks po distsipline 'Raspredelennye informatsionnye sistemy'* [Distributed Information System, Textbook]. Tolyatti, PVGUS Publ., 2013.

8. Karshenboym I. *Kvadriga Apollona i mikroprotsessory* [Apollon Quadriga and Microprocessors]. *Komponenty i tekhnologii* [Components and Technologies], 2006, no. 5, pp. 112–115.

9. Golfand I.Ya. and Khlebutin P.C. *Otsenka trudozatrata razrabotki programnoy komponenty* [Assessment of Working Hours for Software Component Design]. *Trudy ISA RAN* [Proceedings of Institute for Systems Analysis RAS], 2005, vol. 15, pp. 125–135.

10. Afanasiev A.N., Voyt N.N. *Razrabotka komponentno-servisnoy platformy obucheniya: diagrammy ispolzovaniya i deyatelnosti programmnogo komponenta stsenariya na UML-yazyke* [The Development of Component and Service Training Platform: Use and Operation Diagram of UML-Scenario Software Component]. *Vestnik Ulyanovskogo gosudarstvennogo tekhnicheskogo universiteta* [Bulletin of Ulyanovsk State Technical University], 2012, no. 1 (57), pp. 66–68.

11. Afanasiev A.N. and Voyt N.N. *Intellektualnaya obuchayushchaya sistema kontseptualnomu proektirovaniyu avtomatizirovannykh sistem* [Intelligent Training System to Stady Design of Computer-Aided Systems]. *Izvestiya Samarskogo nauchnogo tsentra Rossiyskoy akademii nauk* [Proceedings of the Samara Scientific Centre of the Russian Academy of Sciences], 2010, vol. 12, no. 4–2, pp. 465–468.

12. Afanasiev A.N. and Voyt N.N. *Komponentnaya avtomatizirovannaya obuchayushchaya sistema SAPR na osnove gibridnoy neyronnoy seti* [Automatic Training System SAPR on the basis of the Neural Network]. *Avtomatizatsiya i sovremennye tekhnologii* [Automation and Modern Technologies], 2009, no. 3, pp. 14–18.