

## Раздел II. Автоматизация проектирования

УДК 004.94

**В.Н. Гридин, Г.Д. Дмитриевич, Д.А. Анисимов**

### **АРХИТЕКТУРА РАСПРЕДЕЛЕННЫХ СЕРВИС-ОРИЕНТИРОВАННЫХ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ**

*Актуальной проблемой в области информационных технологий является децентрализация архитектуры САПР путем перехода к распределенным системам проектирования, построенным на основе Интернет-технологий, реализующих задачи коммуникации и информационного обмена между приложениями. Такие независимо управляемые приложения являются автономными и могут взаимодействовать друг с другом в процессе выполнения общей задачи. Протоколы Интернет-технологий представляют надежную базу для связывания подсистем и не требуют согласованного использования ресурсов, находящихся в разных узлах сети, что существенно упрощает процесс построения и эксплуатации распределенной САПР. Основным требованием для возможности реализации такой распределенной системы является согласованность интерфейсов, посредством которых связаны отдельные подсистемы. При выполнении этого требования отдельные компоненты распределенной САПР могут создаваться различными разработчиками и поддерживаться на различных сайтах, откуда они будут поставляться (возможно, на коммерческой основе) потребителям. Самым эффективным методом объединения подсистем в распределенное приложение следует считать организацию удаленного вызова процедур на базе сервис-ориентированной архитектуры с использованием веб-сервисов.*

*Системы автоматизированного проектирования; веб-технологии; сервис-ориентированная архитектура; распределенные системы; веб-сервисы.*

**V.N. Gridin, G.D. Dmitrevich, D.A. Anisimov**

### **ARCHITECTURE OF DISTRIBUTED SERVICE-ORIENTED SYSTEM COMPUTER-AIDED DESIGN**

*Urgent problem in the field of information technology is the decentralization of the architecture of CAD by moving to distributed systems design, built on top of Internet technology, the goal of communication and information exchange between applications. Such independently managed applications are self-contained and can interact with each other in the process of implementation of the general problem. Internet Protocols and technologies represent a solid basis for the binding of subsystems and require coordinated use of resources in different network nodes, which greatly simplifies the process of constructing and operating a distributed CAD. The main requirement for the feasibility of such a distributed system is the consistency of the interfaces through which the individual subsystems are connected. In fulfilling this requirement of individual components distributed CAD can be created and maintained by different developers at different sites, where they will be shipped (perhaps on a commercial basis) to consumers. The most effective method of combining subsystems in a distributed application should be considered organization of the remote procedure call based on service-oriented architecture using web services.*

*Computer-aided design; web technology; ser-bismuth-oriented architecture; distributed systems; web services.*

**Введение.** К числу основных задач в области информационных технологий относится внедрение в системы автоматизированного проектирования веб-технологий, что позволяет реализовать построение систем с сервисно-ориенти-

рованной архитектурой, при которой информационные ресурсы доставляются потребителям посредством сетевых сервисов [1–3]. При этом в отличие от традиционных САПР, распределенная система автоматизированного проектирования может состоять из отдельных модулей, функционирующих независимо и взаимодействующих друг с другом посредством одного из SOA-ориентированных протоколов по схеме «запрос–ответ» [4, 5].

Системы, основанные на сервис-ориентированной архитектуре, независимы от технологий разработки и платформ, при этом приложения, работающие на одних платформах, могут вызывать стандартным способом сервисы, работающие на других платформах. Основная нагрузка по выполнению вычислительных операций при такой архитектуре ложится на веб-сервисы, решающие все задачи моделирования проектируемых систем, на клиентские приложения возлагаются только простейшие функции подготовки данных и отображения результатов моделирования.

Поскольку веб-сервисы могут функционировать на более высоком уровне абстракции, анализируя и обрабатывая типы данных динамическим образом, то отдельным компонентам программного обеспечения предоставляется возможность взаимодействовать более открыто. При использовании универсально описанных интерфейсов появляется возможность использовать программные компоненты повторно, что позволяет снизить трудоемкость разработки САПР и более эффективный возврат инвестиций в программное обеспечение. Учитывая высокую стоимость программного обеспечения современных САПР, указанная возможность имеет большое экономическое значение.

Использование веб-технологий при разработке программного обеспечения систем автоматизированного проектирования позволяет:

- ◆ Перейти к концепции описания интерфейсов и взаимодействий на основе XML, объединяя любой тип приложения с другим приложением, и предоставляя свободу изменения и развития с течением времени до тех пор, пока поддерживается соответствующий интерфейс.
- ◆ Использовать более высокий уровень абстракции программного обеспечения, при котором оно может быть задействовано пользователями, работающими только на уровне бизнес-анализа.
- ◆ Взаимодействовать между различными сервисами на любой платформе, написанными на любом языке программирования;
- ◆ Учитывать слабосвязанность программного обеспечения, благодаря которой взаимодействие между приложениями сервиса не нарушается каждый раз, когда меняется дизайн или реализация какого-либо сервиса.
- ◆ Предоставлять существующему или унаследованному программному обеспечению сервисный интерфейс без изменения оригинальных приложений.
- ◆ Адаптировать существующие приложения к меняющимся условиям проектирования и потребностям заказчика.

**1. Сервис-ориентированная архитектура распределенной системы автоматизированного проектирования.** Сервис-ориентированная архитектура (SOA) основана на модульном подходе к построению программного обеспечения, со стандартными интерфейсами. SOA использует принципы унификации типовых процессов, неоднократного использования элементов, и организацию на базе выбранной платформы. Компоненты программного обеспечения можно распределять по различным узлам, и они предлагаются для применения как слабо связанные, независимые приложения. Хотя архитектура SOA не привязывается к какой-либо одной технологии удаленного вызова методов (COM, DCOM, COM+, .NET REMUTING, Java RMI, CORBA [6]), программные комплексы, построенные со-

гласно SOA, как правило, реализуются как некоторая совокупность веб-сервисов, которые интегрированы согласно с известными стандартными протоколами (WSDL, SOAP). Веб-сервис описывает некоторый набор методов, каждый из которых может быть вызван в сети через стандартизированные XML-сообщения. На основании таких сообщений можно описать требуемые данные способом, не зависящим от платформы, и реализовать обмен информацией между приложениями, что обеспечивает слабосвязанность приложений.

Работающие на одной платформе SOA-приложения, могут вызывать стандартным способом сервисы, функционирующие на иных платформах. Важно отметить, что сервисы не располагают информацией о вызывающем их приложении, а клиентские приложения не имеют информации, о способе, которым сервисы выполняют свои задачи. SOA-программы предоставляет инкапсуляцию всех деталей реализации каждого приложения от иных компонентов, предоставляя гибкий способ неоднократного использования и комбинирования приложений для построения распределённых информационных систем.

Системы на основе сервис-ориентированной архитектуры относятся к классу мультиагентных систем (МАС), которые образованы несколькими взаимодействующими интеллектуальными агентами, обеспечивающими автономность, ограниченность представления и децентрализацию отдельных подсистем распределённой информационно-вычислительной системы.

При разработке систем автоматизированного проектирования с использованием веб-сервисов могут быть применены следующие типы клиентских приложений: приложение консольного типа, приложение оконного типа и веб-приложение.

Особенностью консольных приложений является отсутствие графического интерфейса, однако их использование может оказаться полезным при реализации простейших САПР для карманных компьютеров с небольшой площадью экрана.

Приложения оконного типа дают возможность в наилучшей степени реализовать графические средства и наилучшим образом подходят для разработки распределённых систем на базе веб-сервисов. Для любого веб-сервиса предоставляется возможность построения произвольного числа клиентских приложений с различными способами реализации диалогового взаимодействия.

Веб-приложения обеспечивает возможность целиком разместить все используемое программное обеспечение САПР в сети. Достоинством приложения этой структуры является открытый доступ к использованию распределённой САПР через браузер любого типа [7, 8], недостатком приложения такого типа является увеличение времени, которое требуется для описания компонентов проектируемой системы из-за ожидания реакции на отдельных шагах ввода данных.

Для клиентского приложения любого вида вызов веб-сервисов осуществляется одинаковым способом, и для каждого веб-сервиса возможно использование любых способов реализации клиентских приложений, написанных на различных языках. Если необходимо, такие клиентские приложения можно легко модифицировать согласно с изменяющимися условиями проектирования, возможно также расширение веб-сервиса за счет включения в него дополнительных методов.

Возможная архитектура распределённой сервис-ориентированной системы автоматизированного проектирования на основе технологии веб-сервисов приведена на рис. 1.

Согласно этой архитектуре все веб-сервисы, решающие вычислительные задачи моделирования путем удаленного вызова соответствующих методов, располагаются на сервере веб-сервисов, а клиентские приложения в зависимости от их типа, размещаются либо непосредственно на пользовательских компьютерах ПК (локальные клиентские приложения), либо на сервере веб-приложений (клиент-

ские веб-приложения). При этом, вся работа, связанная с вводом и отображением данных, осуществляется клиентскими приложениями, что позволяет осуществлять их модификацию и адаптацию к условиям проектирования независимо от веб-сервисов. В свою очередь, при условии сохранения выбранного интерфейса возможно развитие и совершенствование методов веб-сервисов, выполняющих основные задачи моделирования [9–13].

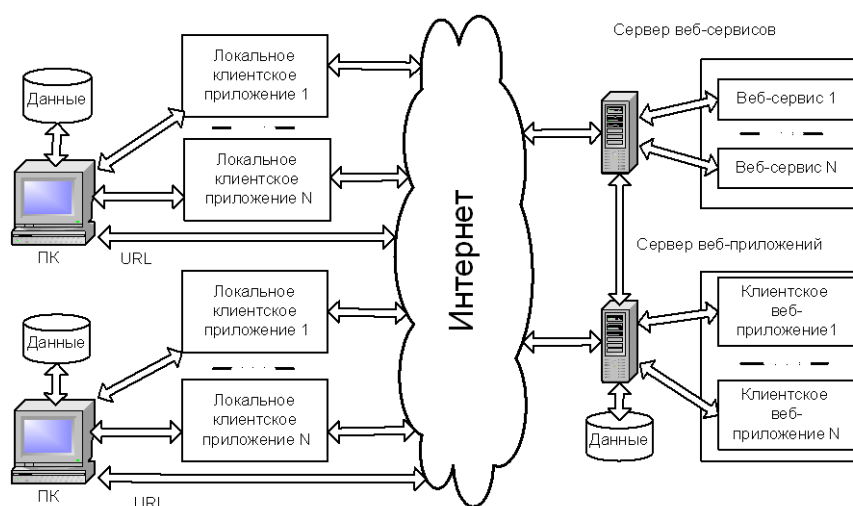


Рис. 1. Архитектура распределенной сервис-ориентированной системы автоматизированного проектирования

**2. Основные протоколы веб-сервисов.** Веб-сервисы основаны на стандарте XML и позволяют обеспечивать работу с программными средствами в сети Интернет. Технически веб-сервисы являются модульными приложениями, имеющими стандартный интерфейс, обеспечивающий работу в сети, и являются слабосвязанными компонентами программы, которые доступны для использования при помощи стандартных Интернет-протоколов. При этом сервис функционирует как закрытая система, задачей клиента является только нахождение требуемого для выполнения требуемой задачи сервиса, а конкретная реализация сервиса для клиентского приложения не имеет никакого значения.

Поскольку веб-сервисы функционируют на достаточно высоком уровне абстракции, динамически обрабатывая и анализируя все типы данных, то отдельным компонентам системы предоставляется возможность взаимодействовать с другими приложениями более открыто. При использовании таких универсальных интерфейсов появляется возможность использовать программные компоненты повторно, что позволяет снизить трудоемкость разработки распределенных систем автоматизированного проектирования. Учитывая высокую стоимость программного обеспечения современных САПР, указанная возможность имеет большое экономическое значение. Важно отметить, что услуга веб-сервиса, в отличие от любой услуги, оказываемой клиенту-пользователю через связанный с ним браузер, предоставляется клиентской программе.

При построении веб-сервисов применяется ряд спецификаций, основанных на открытых стандартах, при этом текст протоколов выполняется в следующей последовательности: поиск веб-сервиса при помощи протокола UDD, описание веб-сервиса на основе WSDL, вызов веб-сервиса через протокол SOAP, кодирование данных (XML), транспортировка (HTTP).

Основным стандартом технологии веб-сервисов является протокол WSDL. Этот протокол используется для создания описания интерфейса веб-сервиса при этом WSDL-документ описывает информационные блоки сервиса и имеет информацию об интерфейсе и методах, доступных для общего пользования, дает описание типов данных, используемых при передаче в запросах и ответах по соответствующему протоколу, а также информацию о транспортном протоколе, который используется для работы с сервисом и адресную информацию о расположении описываемого сервиса. Следует заметить, что стандарт документа WSDL является независимым от какого-либо определенного протокола обмена XML-данными, но содержит встроенные средства поддержки SOAP.

Протокол SOAP – это межплатформенный стандарт, используемый для форматирования сообщений, которыми обмениваются веб-сервис и клиентское приложение. SOAP описывает XML-конверт для сообщений веб-сервиса, модель обработки и алгоритм кодирования информации перед ее отправкой. В технологии веб-сервисов используются два типа SOAP-сообщений. Первый тип является сообщением клиентского запроса, которое клиент направляет веб-сервису для инициализации выбранного метода веб-сервиса. Второй тип является ответным сообщением, которое веб-сервис возвращает клиентскому приложению.

Транспортный протокол HTTP применяется для осуществления обмена информацией с веб-сервисом через SOAP-сообщения. Возможно также в качестве транспортного протокола использование стандарта SMTP.

Протокол UDDI используется для создания бизнес-журналов, фиксирующих имена компаний, предоставляющих веб-сервис и соответствующие URL-адреса их WSDL-контрактов. В качестве альтернативы можно использовать стандарт DISCO, который применяется для создания документов поиска, содержащих ссылки на множество конечных пунктов веб-сервисов.

Необходимость в использовании протокола UDDI (или DISCO) возникает только тогда, когда необходимо осуществлять поиск требуемого для решения задач проектирования веб-сервиса. Если же URL-адрес веб-сервиса уже известен, то его можно просто жестко закодировать или поместить в файл конфигурации, а потом вызвать сервис и осуществить обмен информацией, применяя стандартный формат обмена сообщениями.

**3. Взаимодействие протоколов веб-сервисов.** Взаимодействие протоколов UDDI, WSDL, SOAP, HTTP при использовании веб-сервисов иллюстрируется приведенной на рис. 2 структурой, где сверху изображены процессы регистрации службы и получения ее описания WSDL, а в нижней части – процессы вызова метода службы и возвращения на клиентскую сторону результатов работы сервера.

Обращение к реестру UDDI осуществляется единственный раз и после получения адреса веб-сервиса повторно обычно не выполняется. Единственный раз выполняется также и процесс получения WSDL-документа. Если URL адрес веб-сервиса известен, то обращение к реестру выполнять не обязательно, так как файл WSDL можно вызвать непосредственно из веб-сервиса. Поскольку в реальных условиях работы с распределенными системами на основе веб-сервисов их URL-адреса, как правило, известны, обращение к UDDI осуществляется сравнительно редко.

Вызов метода веб-службы в распределенной системе осуществляется из клиентского приложения через прокси-класс, объект которого реализует требуемую заглушку (stub). Прокси-класс осуществляет преобразование передаваемых методу аргументов в SOAP-сообщение и направляет это сообщение веб-сервису. После выполнения сервисом вызываемого метода, прокси-класс получает соответствующий SOAP-ответ и преобразует его в соответствующие типы данных, что позволяет клиентскому приложению отобразить результаты выполнения метода веб-сервиса.

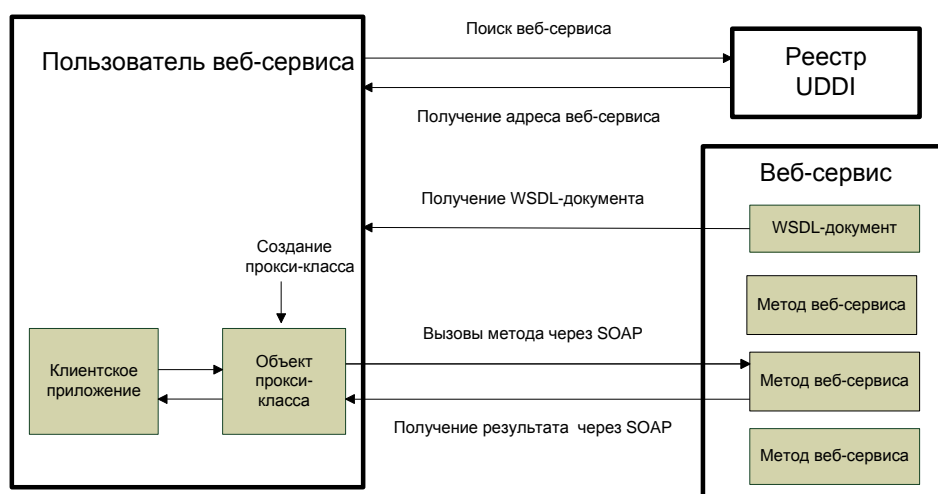


Рис. 2. Взаимодействие протоколов UDDI, WSDL, SOAP, HTTP при использовании веб-сервисов

**Заключение.** В распределенных системах автоматизированного проектирования веб-сервисы функционируют на достаточно высоком уровне абстракции, динамически обрабатывая и анализируя все типы данных. Поэтому отдельным компонентам системы предоставляется возможность взаимодействовать с другими приложениями более открыто. Использование таких универсальных интерфейсов дает возможность использовать программные компоненты повторно, что позволяет снизить трудоемкость разработки распределенных систем автоматизированного проектирования.

Важно отметить – вся работа, связанная с вводом и отображением данных, осуществляется клиентскими приложениями, что позволяет осуществлять их модификацию и адаптацию к условиям проектирования независимо от веб-сервисов. В свою очередь, при условии сохранения выбранного интерфейса возможно развитие и совершенствование методов веб-сервисов, выполняющих основные задачи моделирования

Практическая реализация изложенных концепций существенно повысит эффективность разработки систем автоматизированного проектирования и обеспечит гибкую возможность последующей модификации программного обеспечения в процессе эксплуатации САПР.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Морган С., Райан Б., Хорн Ш., Бломсма М. Разработка распределенных приложений на платформе Microsoft.Net. – СПб.: Питер, 2008. – 608 с.
2. Дей Н., Мандел Л., Райман А. Eclipse: Платформа Web-инструментов: Пер. с англ. – М.: КУДИН-ПРЕСС, 2008. – 688 с.
3. Гридин В.Н., Анисимов В.И. Методы построения систем автоматизированного проектирования на основе Интернет-технологий и компактной обработки разреженных матриц // Информационные технологии в проектировании и производстве. – 2009. – № 1. – С. 3-7.
4. Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А. Построение систем автоматизированного проектирования на основе Web-сервисов // Автоматизация в промышленности. – 2011. – № 1. – С. 9-12.
5. Анисимов Д.А. Методы построения систем автоматизации схемотехнического проектирования на основе веб-сервисов // Известия СПбГЭТУ «ЛЭТИ». – 2012. – № 10. – С. 56-61.

6. Цимбал А.А., Анишина М.П. Технология создания распределенных систем. – СПб.: Питер, 2003. – 576 с.
7. Гридин В.Н., Анисимов В.И., Ларистов Д.А. Архитектура схемотехнических САПР со встроенным браузером // Автоматизация в промышленности. – 2009. – № 11. – С. 52-55.
8. Гридин В.Н., Анисимов В.И., Башкатов А.С. Технология построения системы информационной поддержки распределенных процессов проектирования // Автоматизация в промышленности. – 2010. – № 2. – С. 41-43.
9. Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А. Построение веб-сервисов систем автоматизации схемотехнического проектирования // Информационные технологии и вычислительные системы. – 2012. – № 4. – С. 79-84.
10. Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А. Построение систем автоматизированного проектирования на основе Web-технологий // Информационные технологии. – 2011. – № 5. – С. 23-27.
11. Алмаасали С.А., Анисимов В.И. Построение распределенных систем автоматизированного проектирования на основе методов диакоптики // Известия СПбГЭТУ «ЛЭТИ». – 2014. – № 1. – С. 15-19.
12. Дмитриевич Г.Д., Анисимов Д.А. Построение систем автоматизированного проектирования на основе сервис-ориентированной архитектуры // Известия СПбГЭТУ «ЛЭТИ». 2014. – № 2. – С. 14-18.
13. Коваленко О.С., Курейчик В.М. Обзор проблем и состояний облачных вычислений и серверов // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 146-153.

#### REFERENCES

1. Morgan S., Rayan B., Khorn Sh., Blomsma M. Razrabotka raspredelennykh prilozheniy na platforme Microsoft.Net [The development of distributed applications on the platform Microsoft.Net]. Saint-Petersburg: Piter, 2008, 608 p.
2. Dey N., Mandel L., Rayjman A. Eclipse: Platforma Web-instrumentov. Moscow: KUDIN-PRESS, 2008, 688 p.
3. Gridin V.N., Anisimov V.I. Metody postroeniya sistem avtomatizirovannogo proektirovaniya na osnove Internet-tekhnologiy i kompaktnoy obrabotki razrezhennykh matrits [Methods of construction of computer-aided design on the basis of Internet-technologies and compact processing sparse matrices], *Informatsionnye tekhnologii v proektirovanii i proizvodstve* [Information technologies in design and manufacturing], 2009, No. 1, pp. 3-7.
4. Gridin V.N., Dmitrevich G.D., Anisimov D.A. Postroenie sistem avtomatizirovannogo proektirovaniya na osnove Web-servisov [Creation of systems of automated design based on Web services], *Avtomatizatsiya v promyshlennosti* [Automation in industry], 2011, No. 1, pp. 9-12.
5. Anisimov D.A. Metody postroeniya sistem avtomatizatsii skhemotekhnicheskogo proektirovaniya na osnove veb-servisov [Methods of construction of systems of automation of circuit design based on web services], *Izvestiya SPbGEhTU «LEhTI»* [Proceedings of the ETU LETI], 2012, No. 10, pp. 56-61.
6. Tsimbal A.A., Anshina M.P. Tekhnologiya sozdaniya raspredelennykh sistem [The technology of creation of distributed systems]. Saint-Petersburg: Piter, 2003, 576 p.
7. Gridin V.N., Anisimov V.I., Laristov D.A. Arkhitektura skhemotekhnicheskikh SAPR so vstroennym brouzerom [The architecture of circuit CAD with built-in browser] *Avtomatizatsiya v promyshlennosti* [Automation in industry], 2009, No. 11, pp. 52-55.
8. Gridin V.N., Anisimov V.I., Bashkatov A.S. Tekhnologiya postroeniya sistemy informatsionnoy podderzhki raspredelennykh protsessov proektirovaniya [The technology of creating an information support system of distributed design processes], *Avtomatizatsiya v promyshlennosti* [Automation in Industry], 2010, No. 2, pp. 41-43.
9. Gridin V.N., Dmitrevich G.D., Anisimov D.A. Postroenie veb-servisov sistem avtomatizatsii skhemotekhnicheskogo proektirovaniya [Build web services automation systems of circuit design] *Informatsionnye tekhnologii i vychislitelnye sistemy* [Information Technologies and Computing Systems], 2012, No. 4, pp. 79-84.
10. Gridin V.N., Dmitrevich G.D., Anisimov D.A. Postroenie sistem avtomatizirovannogo proektirovaniya na osnove Web-tekhnologiy [Creation of systems of the automated designing on the basis of Web-technologies], *Informatsionnye tekhnologii* [Information technology], 2011, No. 5, pp. 23-27.

11. *Almaasali S.A., Anisimov V.I.* Postroenie raspredelennykh sistem avtomatizirovannogo proektirovaniya na osnove metodov diakoptiki [Construction of distributed systems of automated design-based methods diakoptical], *Izvestiya SPbGhTU «LEhTI»* [Proceedings of the ETU LETI], 2014, No 1, pp. 15-19.
12. *Dmitrevich G.D., Anisimov D.A.* Postroenie sistem avtomatizirovannogo proektirovaniya na osnove servis-orientirovannoy arkhitektury [Creation of systems of automated design based on service-oriented architecture], *Izvestiya SPbGhTU «LEhTI»* [Proceedings of the ETU LETI], 2014, No. 2, pp. 14-18.
13. *Kovalenko O.S., Kureychik V.M.* Obzor problem i sostoyaniy oblachnykh vychisleniy i serverov [An overview of issues and state of cloud computing and servers], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 146-153.

Статью рекомендовал к опубликованию д.т.н., профессор М.С. Куприянов.

**Гридин Владимир Николаевич** – Центр информационных технологий в проектировании РАН; e-mail: info@ditc.ras.ru; г. Одинцово, ул. маршала Бирюзова, 7а; тел.: 84955960212; д.т.н.; профессор; директор ЦИТП РАН.

**Дмитревич Геннадий Данилович** – Санкт-Петербургский Электротехнический университет; e-mail: gddm@inbox.ru; г. Санкт-Петербург, 2-й Муринский пр., 39, кв. 19; тел.: 84955520688; д.т.н.; профессор.

**Анисимов Денис Андреевич** – e-mail: anisimovdenis2009@yandex.ru; г. Санкт-Петербург, Приморский пр. 55, кв. 29; тел.: 84954300212; к.т.н.; м.н.с.

**Gridin Vladimir Nikolaevich** – Center for Information Technology in designing RAS; e-mail: info@ditc.ras.ru; 7a, Marshala Biryuzova street, Odintsovo, Russia; phone: +74955960212; dr. of eng. sc.; professor; director of the Russian Academy of Sciences TSITP.

**Dmitrievich Gennady Danilovich** – St. Petersburg Electrotechnical University; e-mail: gddm@inbox.ru; 39, 2nd Ave Murinsky, kv.19, St. Petersburg, Russia; phone: +74955520688; dr. of eng. sc.; professor.

**Anisimov Denis Andreevich** – e-mail: anisimovdenis2009@yandex.ru; 55, Primorsky Prospect, kv. 29, St. Petersburg, Russia; phone: +74954300212; cand. of eng. sc.; junior researcher.

УДК 621.3.049.771.14

**С.В. Гаврилов, Г.А. Иванова, А.Л. Стемпковский**

### **ТЕОРЕТИКО-ГРАФОВАЯ МОДЕЛЬ СЛОЖНО-ФУНКЦИОНАЛЬНЫХ БЛОКОВ ДЛЯ КМОП ТЕХНОЛОГИЙ С ТРЕХМЕРНОЙ СТРУКТУРОЙ ТРАНЗИСТОРА**

*Данная статья посвящена исследованию и разработке методов проектирования заказных сложно-функциональных блоков в базе элементов с регулярной топологической структурой в слоях поликремния и диффузии. На сегодняшний день ключевые блоки микроэлектронных систем, такие как, ядра микропроцессоров, микроконтроллеров ведущие разработчики микроэлектронной аппаратуры по-прежнему отрабатывают в полностью заказном режиме, в котором окончательный состав библиотечных элементов заранее неизвестен, и проектирование ведется на предельно низком транзисторном уровне. Однако автоматизация процесса логического и топологического синтеза для полностью заказного проектирования затруднена из-за существенного возрастания сложности задачи с увеличением степени интеграции микроэлектронных систем и уменьшением технологических размеров базовых элементов до 22 нм и ниже. В данной статье проведен сравнительный анализ существующих подходов разработки топологии FinFET структур (технология*