

Першин Иван Митрофанович – Северо-Кавказский федеральный университет, филиал в г. Пятигорске; e-mail: ivmp@yandex.ru; 357500, г. Пятигорск, просп. 40 лет октября, 56; тел.: 88793973927; кафедра управления в технический и биомедицинских системах; зав. кафедрой; профессор.

Веселов Геннадий Евгеньевич – Южный федеральный университет; e-mail: gev@sfedu.ru; 347900, г. Таганрог, ул. Чехова, 2; тел.: +78634360450; Институт компьютерных технологий и информационной безопасности; директор.

Першин Максим Иванович – e-mail: maksimpershin@bkmail.ru; тел.: 89280093030; Институт компьютерных технологий и информационной безопасности; аспирант.

Pershin Ivan Mitrofanovich – North-Caucasian Federal University, a branch in the town of Pyatigorsk; e-mail: ivmp@yandex.ru; 357500, Pyatigorsk, ave. 40 years on October 56; phone: +78793973927; the department of management of technical and biomedical systems; head of department; professor.

Veselov Gennady Evgen'evich – Southern Federal University; e-mail: gev@sfedu.ru; 2, Chekhov street, Taganrog, 347900, Russia; phone: +78634360450; Institute of Computer Technology and Information Security; director.

Pershin Maksim Ivanovich – e-mail: maksimpershin@bkmail.ru; phone: +79280093030; Institute of Computer Technology and Information Security; postgraduate student.

УДК 004.94

В.Н. Гринин, В.И. Анисимов, А.А. Осман

МЕТОДЫ ПОСТРОЕНИЯ ВЕБ-СЛУЖБ .NET ДЛЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ АВТОМАТИЗАЦИИ СХЕМОТЕХНИЧЕСКОГО ПРОЕКТИРОВАНИЯ*

Основной целью статьи является рассмотрение особенностей решения задачи внедрения Интернет-технологий при построении распределенных систем автоматизации схмотехнического проектирования. Показывается, что при построении веб-служб для систем автоматизации схмотехнического проектирования необходимо организовать эффективную передачу и получение большого числа аргументов с простыми и сложными типами данных. Основной проблемой организации клиент-серверных взаимодействий в этом процессе является необходимость работы с многомерными массивами, которые относятся к сложным типам данных. Отмечается, что при этом необходимо учитывать возможную работу веб-службы в гетерогенных средах, когда клиентские и серверные приложения, написаны на различных языках. Для обеспечения гарантированного совместного функционирования клиентских и серверных приложений в гетерогенных средах, предлагается отказаться от передачи информации между такими приложениями через многомерные массивы. Поскольку для описания всех компонентов электронных схем используются двумерные массивы, то единственным решением проблемы является их предварительная упаковка в одномерные массивы с последующей передачей упакованных массивов в качестве аргументов методу веб-службы. Аналогичную процедуру упаковки многомерных массивов следует выполнить перед передачей результатов работы веб-службы клиентскому приложению. Приводится методика приведения сложных типов данных к единообразной форме, что обеспечивает надежность клиент-серверных взаимодействий в гетерогенных средах. Показывается, что WSDL-документ веб-службы не содержит всей необходимой информации для построения клиентских приложений и устанавливается структура дополнительного информационного метода, который должен входить в состав веб-службы. Использование предлагаемой методики построения веб-служб для распределенных систем автоматизации схмотехнического проектирования позволяет существенно повысить эффективность функционирования таких систем.

Системы автоматизированного проектирования; автоматизация схмотехнического проектирования; Интернет-технологии; веб-службы; распределенные системы.

* Исследование выполнено при финансовой поддержке Программы 2 ОНИТ РАН «Научные основы создания гетерогенных телекоммуникационных и локационных систем и их элементной базы».

V.N. Gridin, V.I. Anisimov, A.A. Osman

METHODS FOR CONSTRUCTING WEB SERVICES .NET FOR DISTRIBUTED AUTOMATION SYSTEMS CIRCUIT DESIGN

The main purpose of the article is to examine the features of the solution of the problem of implementation of Internet technologies in the construction of distributed automation systems of circuit design. It is shown that when building Web Services for automation of circuit design necessary to organize efficient transmission and reception of a large number of arguments with simple and complex data types. The main problem of the organization of client-server interactions in this process is the need to work with multi-dimensional arrays that are complex data types. It is noted that while it is necessary to consider the possible work of web services in heterogeneous environments, when client and server applications are written in different languages. In order to ensure interoperability of client and server applications in heterogeneous environments, it is proposed to abandon the transmission of information between these applications through multi-dimensional arrays. As for the description of all the components of electronic circuits using two-dimensional arrays, the only solution is their pre-packaging in one-dimensional arrays with the subsequent transfer of packed arrays as arguments to a Web service method. A similar procedure package multidimensional arrays should be performed prior to transmitting the results of the Web service client application. The technique of bringing the complex data types in a uniform manner that ensures the reliability of client-server interactions in heterogeneous environments. It is shown that the WSDL-document web service does not contain all the necessary information to build client applications and the structure of the additional information set method that should be part of the web service. Using the proposed method of building Web services for distributed automation of circuit design can significantly improve the efficiency of such systems.

Computer-aided design; circuit design automation; Internet technologies; web services; distributed systems.

Введение. Использование Интернет-технологий при построении систем автоматизированного проектирования требует разработки методов реализации этих технологий с учетом специфики предметной области проектирования и особенностей технологии построения веб-служб в конкретных средах их разработки и функционирования [1–9]. Характеристики предметной области существенно влияют на структуру веб-службы и требуют учета ее особенностей при реализации программного обеспечения веб-службы. Из всех предметных областей следует выделить область схемотехнического проектирования, для которой характерна работа с большим разнообразием компонентов, в число которых входят резисторы, конденсаторы, индуктивности, управляемые частотно-зависимые источники различных типов (типа ИТУН, ИНУН, ИТУТ, ИНУТ), трансформаторы, биполярные и униполярные транзисторы, операционные усилители [10–16]. При построении веб-служб для таких систем необходимо организовать эффективную передачу и получение большого числа аргументов с простыми и сложными типами данных. При этом необходимо учитывать возможную работу веб-службы в гетерогенных средах, когда клиентские и серверные приложения, написаны на различных языках, например, на C# и Java. В настоящей статье рассматривается методика построения веб-служб в среде .NET с учетом особенностей их работы в гетерогенных средах.

Целью статьи является рассмотрение методики построения веб-служб .NET с учетом возможности их взаимодействия с клиентскими приложениями в гетерогенных средах.

1. Реализация веб-служб .NET. Реализация веб-служб в среде .NET осуществляется путем использования инструментальных средств, предоставляемых системой Framework .NET, что позволяет построить веб-службу, не анализируя детально структуру SOAP и WSDL [17–19]. При этом создается .NET-класс с атрибутами, которые идентифицируют класс как веб-службу с некоторыми функциями.

Веб-службы строятся на основе трех взаимосвязанных технологий: языка описания веб-служб (WSDL), протокола подключения (HTTP-GET, HTTP-POST или SOAP) и службы обнаружения (disco). Веб-служба строится из тех же типов, что и любая сборка .NET, т.е. из классов, интерфейсов, перечислений и структур, которые играют для клиента роль приложения, отвечающего на запросы. Единственное ограничение связано с тем, что веб-службы предназначены для обработки удаленных вызовов и поэтому в них не следует применять типы для работы с графическим интерфейсом.

При построении веб-служб в среде .NET используются пространства имен, в которых определен ряд основных типов:

- ◆ System.Web.Services – пространство имен, в котором определен минимально-достаточный набор типов для построения веб-службы.
- ◆ System.Web.Services.Description – набор типов для программного взаимодействия с WSDL.
- ◆ System.Web.Services.Protocols – в этом пространстве имен определены типы, предназначенные для работы с протоколами HTTP-GET, HTTP-POST и SOAP.
- ◆ System.Web.Services.Discovery – в этом пространстве имен определены типы, обеспечивающие клиенту веб-служб возможность программно обнаруживать веб-службы, установленные на конкретном компьютере.

Перечисленные типы могут быть использованы как для построения самих веб-служб, так и для создания требуемой инфраструктуры. В проектах по созданию веб-служб основным является пространство имен System.Web.Services. В этом пространстве содержатся тип WebMethodAttribute, добавляющий атрибут [WebMethod] в определение каждого метода веб-службы, который может быть вызван удаленным клиентом по HTTP, тип WebService, определяющий базовый класс для веб-службы, тип WebServiceAttribute, который может быть использован для размещения информации о веб-службе.

При построении веб-службы наиболее целесообразно выбрать шаблон File|New|WebSite|ASP.NET Empty Web Site, где создать проект с выбранным именем (например ModService), затем для сборки ...ModService открыть в меню WebSite| Add New Item и выбрать шаблон Web Service. Среда .NET обеспечивает большую степень языковой независимости и позволяет создавать проекты при помощи C#, VB.NET и других языков программирования, однако для построения веб-служб в среде .NET следует рекомендовать язык C#.

Построенный таким образом каркас проекта будет включать в себя файл WebService.asmx в XML-совместимом формате WSDL, на основе которого среда выполнения ASP.NET автоматически генерирует код HTML. Кроме того, в проекте создается исходный файл WebService.cs, в котором и будет реализована вся программная логика веб-службы.

В файле WebService.cs автоматически создается класс WebService, описанию которого предшествует атрибут [WebService(Namespace = "http://tempuri.org/)]. В список аргументов этого атрибута следует добавить аргументы Description и Name, значение первого из которых должно содержать информацию о назначении веб-службы (например, "Моделирование систем в частотной области"), а второй – некоторый псевдоним для краткого имени приложения (например, "MFD"). При наличии аргумента Name его значение будет определять имя класса создаваемого прокси-объекта (при отсутствии аргумента Name имя класса прокси-объекта будет совпадать с именем класса WebService, что не всегда удобно вследствие возможного конфликта имен).

Как и для любого проекта ASP.NET, при создании проекта Web Service среда VisualStudio.NET автоматически создаст виртуальный каталог для функционирования этого проекта в среде VisualStudio.NET. Однако следует отметить, что при использовании веб-службы вне среды VisualStudio.NET, необходимо создавать вручную в IIS виртуальный каталог с некоторым псевдонимом (в операционных системах Windows 7 и более поздних – именованное при помощи псевдонима приложение). В качестве псевдонима при этом следует выбирать имя, совпадающее со значением аргумента Name в атрибуте WebService (например, "MFD").

В подготовленном таким образом каркасе проекта следует выполнить описание всех переменных и методов класса WebService, при этом все методы веб-службы обязательно должны иметь модификатор public и описанию каждого метода должен предшествовать атрибут [WebMethod(Description =...)], значение аргумента Description которого должно содержать информацию о назначении метода (например, "Выполняет формирование математического описания схемы и расчет передаточных функций").

Характер программной логики веб-службы определяется предметной областью и конкретными задачами, возлагаемыми на веб-службу. Применительно к построению распределенной системы схмотехнического проектирования одной из таких задач является моделирование электронных схем в частотной области.

VisualStudio.NET автоматически создает документ WSDL (Web Service Description Language, язык описания веб-служб). WSDL – это XML-совместимый язык, который полностью описывает для внешних клиентов возможности веб-служб, методы, которые клиенты могут вызвать, а также поддержку протоколов подключения к веб-службам (HTTP-GET, HTTP-POST и SOAP).

Документ WSDL открывается и закрывается корневым элементом <definitions>, при этом в открывающемся дискрипторе содержатся различные атрибуты xmlns, которые задают пространство имен XML, определяющие подчиненные элементы. Обязательным в описании является пространство имен, где определены сами элементы WSDL (<http://schemas.xmlsoap.org/wsdl>). В пределах корневого элемента <definitions> помещаются пять подчиненных элементов, при этом общий вид WSDL-документа имеет структуру:

```
<?xml versions= "1.0" encoding= "utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap"
.....
  <wsdl:types>
    <!-- Список типов, доступных для данной веб-службы -->
  </wsdl:types> <wsdl:message>
    <!-- Формат сообщений -->
  </wsdl:message> <wsdl:portType>
    <!-- Информация портов -->
  </wsdl:portType> <wsdl:binding>
    <!-- Информация связывания -->
  </wsdl:binding> <wsdl:service>
    <!-- Информация о самой веб-службе-->
  </wsdl:service>
</wsdl:definitions>
```

Поскольку код WSDL генерируется автоматически, детальное рассмотрение кода WSDL создаваемой web-службы не обязательно. Наиболее важная информация содержится в пределах элемента <wsdl:service>, где содержится адрес размещения сервиса в сети. Вместе с тем все содержание WSDL-документа может рассматриваться как «контракт» между клиентским приложением и самой веб-службой. В частности, WSDL используется для описания основных характеристик методов веб-службы, к которым относятся:

- ◆ Имя метода веб-службы.
- ◆ Число, тип и порядок следования параметров.
- ◆ Тип возвращаемого методом значения.
- ◆ Условия вызова HTTP GET, HTTP POST, SOAP.

Существенной является возможность получения WSDL-документа до построения клиентского приложения. Для этого необходимо добавить суффикс ?wsdl к адресу URL, указывающему на файл *.asmx веб-службы. Иллюстрацией является приводимый на рис. 1 фрагмент полученного таким образом WSDL-документа.

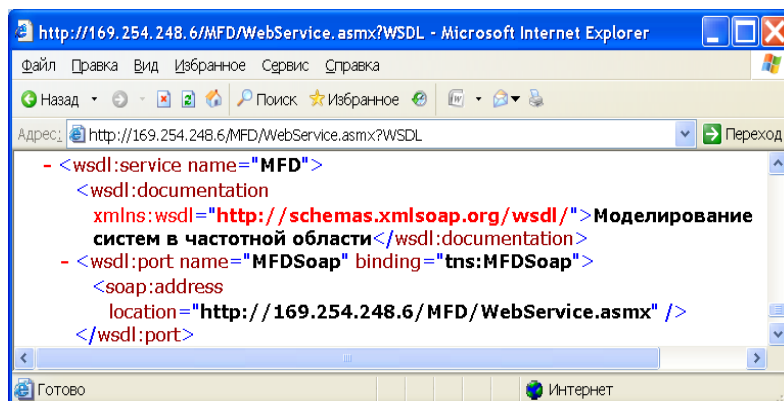


Рис. 1. Отображение в браузере WSDL-документа

Полезно отметить, что WSDL-документ содержит значения введенных при создании проекта аргументов Description = "Моделирование систем в частотной области" и Name = "MFD", а также адрес URL расположения веб-службы.

2. Особенности работы веб-служб со сложными типами данных. Особенностью систем автоматизации схемотехнического проектирования является большое разнообразие используемых в них компонентов. Так, компонентами таких систем являются резисторы, конденсаторы, индуктивности, управляемые частотно-зависимые источники различных типов (типа ИТУН, ИНУН, ИТУТ, ИНУТ), трансформаторы, биполярные и униполярные транзисторы, операционные усилители и др. Схемы включения и параметры компонентов должны быть описаны в клиентском приложении в некоторых двумерных массивах, содержание которых должно быть передано на сервер, где размещена веб-служба. Методы этой веб-службы получают описание компонентов через передаваемые аргументы, выполняют формирование математического описания схемы и ее расчет и возвращают результаты расчета клиентскому приложению, в общем случае также в виде многомерных массивов.

Основной проблемой организации клиент-серверных взаимодействий здесь является необходимость работы с многомерными массивами, которые относятся к сложным типам данных. С одной стороны, благодаря стандартам XML и SOAP среда .NET позволяет передавать веб-службе сложные типы данных и получать их в качестве выходных данных. В частности, наборы данных DataSet, которые могут содержать множество таблиц, колонок, являющиеся ключевым компонентом технологии доступа в ADO.NET компании Microsoft, могут быть преобразованы в XML и обратно, и следовательно могут передаваться веб-службам .NET и возвращаться от них клиенту. Однако такие возможности предоставляются только в случае, когда клиентское и серверное приложения реализованы в среде .NET, когда обеспечиваются единообразные подходы к организации работы со сложными типами данных.

Вместе с тем, стандартизация XML и SOAP позволяет организовать взаимодействие клиентских и серверных приложений, написанных на разных языках и реализованных в различных средах, что является одним из основных достоинств использования веб-служб при построении распределенных систем. Но такое взаимодействие строго обеспечивается только при использовании простых типов данных (целочисленные и нецелочисленные числовые типы, булевские и символьные типы, а также одномерные массивы с элементами любого поддерживаемого типа). Так, например, наборы данных DataSet не поддерживаются в J2EE [20], что исключает возможность совместного использования таких данных в гетерогенных средах, где приложения, написаны на различных языках, например, на C# и Java.

Таким образом, для обеспечения гарантированного совместного функционирования клиентских и серверных приложений в гетерогенных средах, следует отказаться от передачи информации между такими приложениями через многомерные массивы. Поскольку для описания всех компонентов используются двумерные массивы, то единственным решением проблемы является их предварительная упаковка в одномерные массивы с последующей передачей упакованных массивов в качестве аргументов методу веб-службы. Аналогичную процедуру упаковки многомерных массивов следует выполнить перед передачей результатов работы веб-службы клиентскому приложению.

Следовательно, если в описании компонентов какого либо вида используется двумерный массив некоторого типа T с именем `arr_comp`, число строк которого (число компонентов данного вида) имеет значение `ncomp`, а число столбцов массива имеет значение `ncol`, т.е. `T arr_comp = new T[ncomp+1, ncol]`, где строка с нулевым индексом не используется, то такой массив в клиентском приложении упаковывается по столбцам в одномерный массив с именем `Arr_comp`, т.е. `T Arr_comp = new T[(ncol-1)*ncomp + 1]`, при этом элемент с нулевым индексом не используется.

Аналогичным образом в методах веб-служб упаковываются многомерные массивы, в которых отображаются результаты работы веб-службы.

3. Информационный метод для веб-служб. Технология построения распределенных систем автоматизированного проектирования основана на независимом построении веб-службы и клиентских приложений различными коллективами программистов. При этом каждый коллектив не имеет полной информации о всех деталях структуры приложения, разрабатываемого другим коллективом. Несмотря на то, что при построении веб-службы формируется WSDL-документ с описанием веб-службы, и доступ к этому документу возможен до построения клиентского приложения, проблема не исчезает, так как WSDL-документ обеспечивает требуемую информацию для возможности клиент-серверного взаимодействия, но не содержит достаточной информации для построения клиентского приложения. Так, в частности в этом документе отсутствует информация о выбранных при построении веб-службы способах преобразования многомерных массивов к единообразной форме.

Поэтому каждая веб-служба распределенной системы автоматизации схемотехнического проектирования должна обеспечивать разработчиков клиентских приложений дополнительной информацией, необходимой для независимого от разработчика веб-службы построения этих приложений. Такая задача может быть решена путем построения дополнительного информационного метода (например, `Info()`), не имеющего аргументов, и возвращающего простую строковую переменную. Содержание этой переменной должно включать в себя:

- ◆ Описание компонентов схемы (число компонентов данного вида `ncomp`, массив включения `in_comp`, массив значений параметров `z_comp` и порядков формирования этих массивов для каждого типа компонента).

- ◆ Алгоритм преобразования многомерных массивов описания компонентов схемы к унифицированной форме In_comp , Z_comp , обеспечивающей гарантированное функционирование распределенной системы в гетерогенных средах.
- ◆ Оператор формирования прокси-объекта для взаимодействия веб-службы с клиентскими приложениями (например, `MDF proxy = new MDF();`)
- ◆ Заголовки рабочих методов веб-службы.
- ◆ Описание возвращаемых результатов работы методов веб-службы в унифицированной форме.
- ◆ Описание преобразования унифицированной формы возвращаемых результатов методов веб-службы к виду, требуемому для отображения результатов работы веб-службы в клиентских приложениях.

Содержание этой переменной следует ввести в аргумент `Description` атрибута `WebServices` для созданного метода `Info`, что приведет к ее отображению в браузере без необходимости вызова самого метода `Info` (рис. 2).

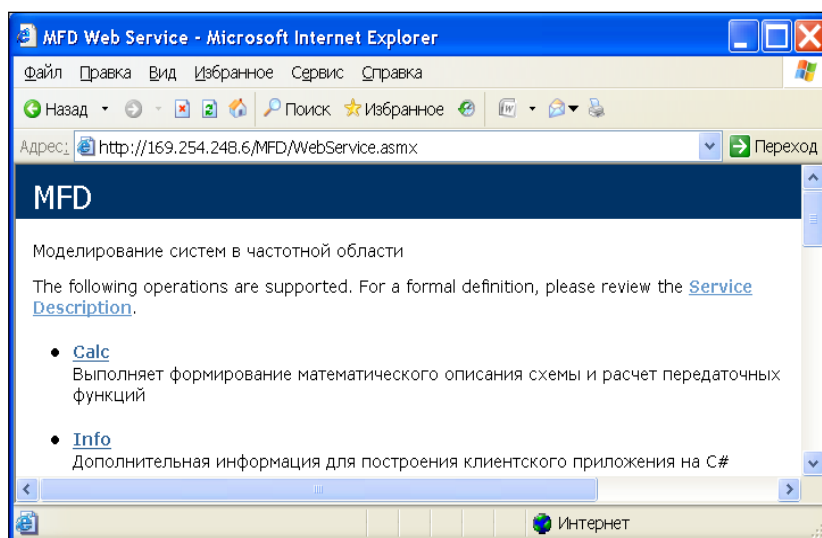


Рис. 2. Вызов информационного метода `Info`

Таким образом, можно получить всю необходимую информацию о веб-службе еще до построения клиентского приложения.

Заключение. Реализация Интернет-технологий при построении распределенных САПР требует учитывать специфику предметной области систем автоматизации схмотехнического проектирования, характеризующихся большим разнообразием компонентов, для описания которых используются сложные типы данных, что является принципиальным отличием систем этого типа. При этом необходимо учитывать возможную работу веб-службы в гетерогенных средах, когда клиентские и серверные приложения написаны на различных языках.

Использование предлагаемой методики построения веб-служб для распределенных систем автоматизации схмотехнического проектирования позволяет существенно повысить эффективность функционирования таких систем и обеспечивает надежное взаимодействие клиентских и серверных приложений в гетерогенных средах.

В результате решена важная научная задача организации взаимодействия клиентских и серверных приложений в гетерогенных средах, а также практическая задача включения в стандартный WSDL–документ дополнительной информации о способе унификации передаваемых и возвращаемых массивов.

В отличие от существующих работ предлагаемая методика обеспечивает полную самодокументированность веб-службы, что позволяет осуществлять построение клиентских приложений автономно от разработчиков веб-службы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *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.
2. *Коваленко О.С., Курейчик В.М.* Обзор проблем и состояний облачных вычислений и серверов // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 146-153.
3. *Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А.* Построение систем автоматизированного проектирования на основе Web-сервисов // Автоматизация в промышленности. – 2011. – № 1. – С. 9-11.
4. *Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А.* Построение систем автоматизированного проектирования на основе Web-технологий // Информационные технологии. – 2011. – № 5. – С. 23-27.
5. *Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А.* Архитектура распределенных сервис-ориентированных систем автоматизированного проектирования // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 51-58.
6. *Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А.* Методика построения веб-сервисов распределенных платформенно-независимых систем автоматизированного проектирования // Системы и средства информатики. – 2014. – № 1. – С. 219-229.
7. *Гридин В.Н., Анисимов В.И., Абухазим М.М.* Методы построения высокопроизводительных систем на основе сжатия данных // Информационные технологии и вычислительные системы. – 2015. – № 2. – С. 16-21.
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. *Dmitrevich G.D., Anisimov D.A.* Platform-Independent Computer-Aided Circuit Design Systems Based on Web Services Technology // Proceedings of the IEEE, North West Section. – 2014. – Vol. 6. – P. 49-53.
10. *Anisimov V.I., Almaasali S.A.* Methods of Organization of Computations During the Simulation of Grand Systems Based on Diacoptics // Proceedings of the IEEE, North West Section. – 2014. – Vol. 6. – P. 44-48.
11. *Алмаасали С.А., Анисимов В.И.* Построение распределенных систем автоматизированного проектирования на основе методов диакоптики // Известия СПбГЭТУ. – 2014. – № 1. – С. 15-19.
12. *Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А.* Построение веб-сервисов систем автоматизации схемотехнического проектирования // Информационные технологии и вычислительные системы. – 2012. – № 4. – С. 17-23.
13. *Гридин В.Н., Анисимов В.И., Алмаасали С.А.* Повышение эффективности процессов моделирования нелинейных систем // Информационные технологии в проектировании и производстве. – 2013. – № 4. – С. 10-13.
14. *Гридин В.Н., Анисимов В.И., Шабани М.А.* Методы построения высокопроизводительных распределенных систем автоматизации схемотехнического проектирования // Информационные технологии. – 2014. – № 8. – С. 59-63.
15. *Гридин В.Н., Анисимов В.И., Алмаасали С.А.* Применение метода диакоптики для моделирования и расчета больших систем // Проблемы управления. – 2014. – № 4. – С. 9-13.
16. *Robert Heineman.* PSPICE Einführung in die Elektroniksimulation, Carl Hanser Verlag München, 2011, 400 Seiten.
17. *Matthew MacDonald, Adam Freeman, Mario Szpuszta.* Pro ASP.NET 4 in C# 2010. Apress, Berkeley, 2010. – 1413 p.

18. *Morgan S., Ryan B., Blomsa M.* Microsoft Self-Placed Training Kit. Microsoft Press, 2008. – 749 p.
19. *Alex Ferrara, Matthew MacDonald.* Programming .NET Web Services. O'Reily, Beijing, Cambridge, 2002. – 422 p.
20. *Dey N., Mandel L., Ryman A.* Eclipse Web Tools Platform. Developing Java Web Applications. Pearson Education, Inc. 2007. – 752 p.

REFERENCES

1. *Gridin V.N., Anisimov V.I.* Metody postroeniya sistem avtomatizirovannogo proektirovaniya na osnove Internet-tekhnologiy i kompaktnoy obrabotki razrezhennykh matrits [Methods of construction of systems of the automated designing on the basis of Internet-technologies and compact processing sparse matrices], *Informatsionnye tekhnologii v proektirovanii i proizvodstve* [Informacionnye Tehnologii v Proektirovanii i Proizvodstve], 2009, No. 1, pp. 3-7.
2. *Kovalenko O.S., Kureychik V.M.* Obzor problem i sostoyaniy oblachnykh vychisleniy i serverov [Review of problems and aspects about cloud computing and services], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 146-153.
3. *Gridin V.N., Dmitrevich G.D., Anisimov D.A.* Postroenie sistem avtomatizirovannogo proektirovaniya na osnove Web-servisov [Construction of systems of the automated designing on the basis of Web services], *Avtomatizatsiya v promyshlennosti* [Automation and Remote Control], 2011, No. 1, pp. 9-11.
4. *Gridin V.N., Dmitrevich G.D., Anisimov D.A.* Postroenie sistem avtomatizirovannogo proektirovaniya na osnove Web-tekhnologiy [Construction of systems of the automated designing on the basis of Web-technologies], *Informatsionnye tekhnologii* [Information Technology], 2011, No. 5, pp. 23-27.
5. *Gridin V.N., Dmitrevich G.D., Anisimov D.A.* Arkhitektura raspredelennykh servis-orientirovannykh sistem avtomatizirovannogo proektirovaniya [Architecture of distributed service-oriented system computer-aided design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 7 (156), pp. 51-58.
6. *Gridin V.N., Dmitrevich G.D., Anisimov D.A.* Metodika postroeniya veb-servisov raspredelennykh platformenno-nezavisimykh sistem avtomatizirovannogo proektirovaniya [Methodology of construction of web services of distributed platform-independent computer-aided design], *Sistemy i sredstva informatiki* [Systems and Means of Informatics], 2014, No. 1, pp. 219-229.
7. *Gridin V.N., Anisimov V.I., Abukhazim M.M.* Metody postroeniya vysokoproizvoditel'nykh sistem na osnove szhatiya dannykh [Methods for constructing high-performance systems based on data compression], *Informatsionnye tekhnologii i vychislitel'nye sistemy* [Information Technology and Computer Systems], 2015, No. 2, pp. 16-21.
8. *Gridin V.N., Anisimov V.I., Bashkatov A.S.* Tekhnologiya postroeniya sistemy informatsionnoy podderzhki raspredelennykh protsessov proektirovaniya [Technology of construction of system of information support of distributed design processes], *Avtomatizatsiya v promyshlennosti* [Automation and Remote Control], 2010, No. 2, pp. 41-43.
9. *Dmitrevich G.D., Anisimov D.A.* Platform-Independent Computer-Aided Circuit Design Systems Based on Web Services Technology, *Proceedings of the IEEE, North West Section*, 2014, Vol. 6, pp. 49-53.
10. *Anisimov V.I., Almaasali S.A.* Methods of Organization of Computations During the Simulation of Grand Systems Based on Diacoptics, *Proceedings of the IEEE, North West Section*, 2014, Vol. 6, pp. 44-48.
11. *Almaasali S.A., Anisimov V.I.* Postroenie raspredelennykh sistem avtomatizirovannogo proektirovaniya na osnove metodov diakoptiki [The construction of distributed systems of the automated designing on the basis of methods of diakoptic], *Izvestiya SPbGETU «LETI»*, 2014, No. 1, pp. 15-19.
12. *Gridin V.N., Dmitrevich G.D., Anisimov D.A.* Postroenie veb-servisov sistem avtomatizatsii skhemotekhnicheskogo proektirovaniya [Building web services the automation of circuit design], *Informatsionnye tekhnologii i vychislitel'nye sistemy* [Information Technology and Computer Systems], 2012, No. 4, pp. 17-23.

13. Gridin V.N., Anisimov V.I., Almaasali S.A. Povyshenie effektivnosti protsessov modelirovaniya nelineynykh sistem [Improving the efficiency of processes modeling nonlinear systems], *Informatsionnye tekhnologii v proektirovanii i proizvodstve* [Informacionnye Tehnologii v Proektirovanii i Proizvodstve], 2013, No. 4, pp. 10-13.
14. Gridin V.N., Anisimov V.I., Shabani M.A. Metody postroeniya vysokoproizvoditel'nykh raspredelennykh sistem avtomatizatsii skhemotekhnicheskogo proektirovaniya [Methods for constructing high-performance distributed systems automation circuit design], *Informatsionnye tekhnologii* [Information Technology], 2014, No. 8, pp. 59-63.
15. Gridin V.N., Anisimov V.I., Almaasali S.A. Primenenie metoda diakoptiki dlya modelirovaniya i rascheta bol'shikh sistem [Application of the method of diakoptic for modeling and calculation of large systems], *Problemy upravleniya* [Problemy Upravleniya], 2014, No. 4, pp. 9-13.
16. Robert Heineman. PSPICE Einführung in die Elektroniksimulation, Carl Hanser Verlag München, 2011, 400 Seiten.
17. Matthew MacDonald, Adam Freeman, Mario Szpuszta. Pro ASP.NET 4 in C# 2010. Apress, Berkeley, 2010, 1413 p.
18. Morgan S., Ryan B., Blomsa M. Microsoft Self-Placed Training Kit. Microsoft Press, 2008, 749 p.
19. Alex Ferrara, Matthew MacDonald. Programming .NET Web Services. O'Reily, Beijing, Cambridge, 2002, 422 p.
20. Dey N., Mandel L., Ryman A. Eclipse Web Tools Platform. Developing Java Web Applications. Pearson Education, Inc. 2007, 752 p.

Статью рекомендовал к опубликованию д.т.н., профессор Г.Д. Дмитриевич.

Гридин Владимир Николаевич – Центр информационных технологий в проектировании РАН; e-mail: info@ditc.ras.ru, г. Одинцово, ул. Маршала Бирюзова, 7а; тел.: 84955960219; д.т.н.; профессор; научный руководитель ЦИТП РАН.

Анисимов Владимир Иванович – д.т.н.; профессор.

Осман Анур Абубекер – Санкт-Петербургский государственный электротехнический университет; г. Санкт-Петербург, ул. проф. Попова, 5; тел.: 89531529762; аспирант.

Gridin Vladimir Nikolaevich – Center for Information Technologies in designing RAS; e-mail: info@ditc.ras.ru, 7a, Marshal Biryuzova street, Odintsovo, Russia; phone: +74955960219; dr. of eng. sc.; professor; scientific director of the Russian Academy of Sciences TSITP.

Anisimov Vladimir Ivanovich – dr. of eng. sc.; professor.

Osman Anur Abubeker – St. Petersburg State Electrotechnical University; 5, prof. Popova street, St.-Petersburg, Russia; phone: +79052208714; postgraduate student.