

8. *Vestnik Banka Rossii*. 2010, no. 36–37 (1205–1206).  
9. GOST R ISO/MEK 13335-3-2007 “Metody i sredstva obespecheniya bezopasnosti”: *Metody menedzhmenta bezopasnosti informatsionnykh tekhnologiy* (GOST R ISO / IEC 13335-3-2007 “Methods and tools to ensure security”: *Methods for Management of Information Technology Security*). Moscow, ИПК, 2007, 49 p.

10. *Rukovodstvo po upravleniyu riskami dlya sistem informatsionnykh tekhnologiy: Rekomendatsii Natsional'nogo instituta standartov i tekhnologiy* (Risk Management Guide for Information Technology Systems: Recommendations of the National Institute of Standards and Technology). Moscow, 2003, available at: <http://egov-center.ru/sites/default/files/000077.pdf>. (14.05.2013)

© Антамошкин О. А., Пузанова Г. А., Онтужев В. В., 2013

УДК 621.31:681.5

## СТРУКТУРА WEB-СЕРВИСА ДЛЯ МОДЕЛИРОВАНИЯ ЭЛЕКТРИЧЕСКИХ СХЕМ С СУЩЕСТВЕННЫМИ НЕЛИНЕЙНОСТЯМИ

К. В. Богданов, А. Н. Ловчиков

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева  
Россия, Красноярск, просп. им. газ. «Красноярский рабочий», 31  
E-mail: 2519869@gmail.com, ivt\_anlovch@sibsau.ru

*Цель работы заключается в создании программного обеспечения, позволяющего проводить анализ и моделирование в системах с несколькими блоками, обладающими существенной нелинейностью, такими как электронные ключи, например. Сложность создания такой системы заключается в том, что существующие подходы к моделированию предполагают решение системы дифференциальных уравнений, при составлении которой вносятся искусственные ограничения, влияющие на поведение системы непосредственно в моменты переключений. Предлагаемый подход опирается на представление модели не как единой, описываемой одним набором ДУ, а как набора взаимодействующих между собой простых систем. Так как такое моделирование накладывает специфические требования на оборудование, было выдвинуто предложение разработать данное ПО как систему, работающую в облаке.*

*Ключевые слова: нелинейные системы, моделирование, функциональные языки программирования, облачные сервисы.*

## STRUCTURE OF SIGNIFICANT NONLINEARITY ELECTRIC CIRCUITS OF THE WEB-SERVICE MODELING

K. V. Bogdanov, A. N. Lovchikov

Siberian State Aerospace University named after Academician M. F. Reshetnev  
31 “Krasnoyarskiy Rabochiy” prosp., Krasnoyarsk, 660014, Russia  
E-mail: 2519869@gmail.com, ivt\_anlovch@sibsau.ru

*The objective of the work is the creation of such a software system which will help to analyze and model the systems with multiple blocks bearing the significant nonlinearity (as transistor keys, for example). The complicity of this work is in need of creation of the main system of differential equations for the system as a whole. Such systems of differential equation has some important restrictions that may affect the results of modeling, especially in bifurcation points. We offer a method that represents the system as a number of connected simple subsystems. Each of them has its own math model. This approach impose specific requirements on hardware, so we offer to use the cloud computing to solve this inconvenience.*

*Keywords: nonlinear systems, modeling, functional programming, cloud computing*

Моделирование и анализ работы электронного оборудования – весьма сложная задача, для решения которой активно используется специализированное программное обеспечение – EDA-системы. Развитие таких систем идёт, по большей части, экстенсивным

путём, начиная от систем, созданных в 70-ых годах прошлого столетия. Как правило, улучшаются пользовательские интерфейсы, расширяются базы данных электронных компонент и т. п., в то время как основные вычислительные алгоритмы остаются прежними.

Чаще всего всё сводится к решению получившейся системы линейаризованных дифференциальных уравнений, являющейся математической моделью моделируемого устройства. Численные методы решения при этом позволяют получать весьма точные результаты, но проблемы возникают при моделировании систем с большим количеством электронных компонентов, что приводит к усложнению математической модели, и при моделировании систем, в которых имеются существенные нелинейности, которые значительно огрубляют результаты моделирования, либо вообще приводят к тому, что вычислительный процесс расходится.

В работах [1; 2] авторами был предложен альтернативный подход для моделирования подобных систем. Вместе с тем, стоит признать, что схемы, для которых требуется подобный подход в моделировании, весьма специфичны (как правило, это схемы с ШИМ). Следовательно, EDA-система для моделирования работы схем с существенными нелинейностями должна либо производить моделирование также всех остальных электронных схем на уровне аналогичного ПО, являющегося отраслевым стандартом де-факто, либо интегрироваться в процесс разработки. Так как первый путь является практически не реализуемым, то имеется смысл в создании узкоспециализированного ПО. Реализация данного ПО в виде Web-сервиса в сети Интернет позволит сделать доступ к нему наиболее лёгким для всех потенциальных пользователей.

Как было отмечено в [2], ядром данного сервиса становится моделирующий модуль, реализованный в виде виртуальной машины Erlang. Для обеспечения работы ядра, ему необходимо передать информацию об объекте моделирования в виде структуры взаимосвязей входов и выходов элементов схемы и сведений о типах самих элементов и их номиналах (рис. 1). В качестве формата, позволяющего описать данную информацию было принято решение использовать формат SPICE Netlist [3] из-за его широкой распространённости и простоты. Кроме того, пользователю представляется возможность в графическом виде изо-

бразить необходимую схему, которую также надо будет преобразовать в данный формат.

Учитывая многопользовательскую природу web-сервисов, системе необходимо иметь возможность проводить одновременно независимое моделирование нескольких десятков схем, т. е. запускать по одной виртуальной машине для каждого процесса моделирования. При этом вычислительная нагрузка может существенно варьироваться. Наиболее адекватным инфраструктурным решением для поддержания такой схемы работы, является размещение ядра в «облаке», т. к. ресурсы в этом случае будут выделяться по необходимости.

С учетом вышесказанного, архитектура web-сервиса состоит из следующих уровней:

1. Клиентский уровень – обеспечивает работу пользовательского интерфейса, отвечает за работу редактора схем и загрузку пользовательской информации в виде SPICE Netlist, осуществляет вывод результатов моделирования.

2. Уровень преобразования и контроля – осуществляет синтаксический и, отчасти, семантический контроль введенной информации, проверяет наличие моделей элементов, используемых пользователем в схеме, в базе ядра, осуществляет преобразование графического представления схемы в единый формат.

3. Библиотеки элементов – коллекции функций, реализующих отдельные электронные компоненты

4. Подсистема хранения – СУБД, реализующая хранение вводимой пользовательской информации и результатов расчета

5. Подсистема работы с облачным сервисом [4] – осуществляет запуск/останов счетных ядер, передаёт им исходные данные и получает результаты расчетов.

6. Виртуальные машины в облачном сервисе – осуществляют собственно моделирование.

Клиентский уровень реализован в виде т.н. «одностраничного» web-приложения, представляющего собой визуальный редактор принципиальных схем, работающий в во всех современных браузерах, с возможностью импорта схем в формате SPICE Netlist.

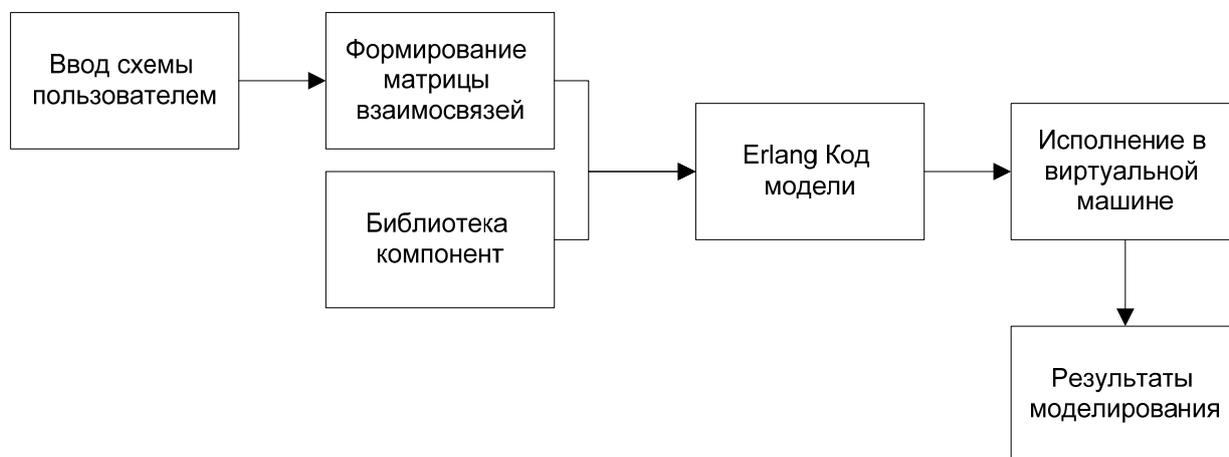


Рис. 1. Этапы процесса моделирования

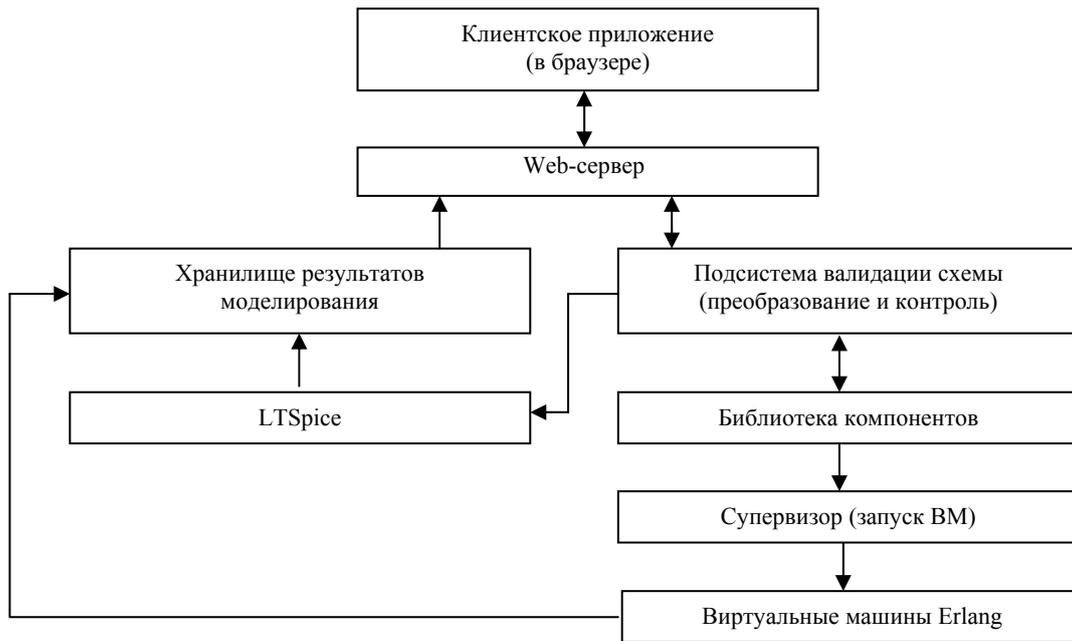


Рис. 2. Структура сервиса

Важным условием является необходимость наличия описанных элементов в библиотеке сервиса, иначе схема не сможет быть обработана. Если это условие выполняется, то приложение анализирует предоставленный файл, работая в тесной связи со вторым уровнем, и строит на его основе графическое представление принципиальной схемы, которое в дальнейшем может быть отредактировано стандартными визуальными методами. Обратный экспорт в формат SPICE Netlist не осуществляется (хотя принципиально возможен).

Получаемый файл описания схемы передаётся для моделирования серверной части, которая также анализирует параметры схемы, но уже на предмет наличия нелинейных компонентов. После чего, опираясь на результаты анализа, запускает либо обычный процесс моделирования путём вызова LTSpice (это происходит в том случае, если нелинейных компонентов не более 1), либо начинает создание отдельных процессов для каждого из компонентов схемы в виртуальной машине Erlang.

Для функционирования системы необходимо иметь несколько различных типов элементов:

1. Обычный процесс (передаточная функция, пример реализации приведён выше). Обеспечивает преобразование входного потока данных в выходной. В каждый момент синхронизации обязан принять и передать одну порцию (кортеж) данных.

2. Коммутатор. Обеспечивает перераспределение сигналов по нескольким выходным каналам в зависимости от соотношения количества входов и выходов. По сути является аналогом узла в электротехнике, но во избежание путаницы в терминологии, будем называть его коммутатором.

3. Источник. Обеспечивает выдачу сигналов. Входов не имеет.

4. Выход. Псевдоблок, необходимый для получения текущих значений параметров для анализа. Является адаптером между моделью и пользовательским интерфейсом. Имеет один вход, выходов нет.

Одной из важнейших процедур при создании подобной модели является построение таблицы маршрутизации по пользовательской модели (к примеру, по принципиальной электрической схеме). На основе этой информации также должны создаваться процессы-коммутаторы.

После запуска, все элементы начнут передавать сообщения маршрутизирующему процессу (серверу), который будет их распределять, опираясь на заданную топологию моделируемой системы.

Важнейшей функцией клиентского уровня также является визуализация результатов моделирования, которые возвращаются клиентскому браузеру в виде AJAX-ответа в формате JSON, а после могут быть визуализированы в виде двумерного графика, либо выгружены в формате CSV.

Наличие двух вариантов моделирования обусловлена тем фактом, что для схем, в которых не встречаются существенные нелинейности, моделирование с помощью виртуальной машины Erlang будет проходить медленнее, а, кроме того, будет использован относительно дорогостоящий облачный вычислительный ресурс. Кроме того, на данном этапе схема формирования набора вычислительных процессов находится на экспериментальном этапе, а возможность анализа пока ограничена только анализом переходных процессов.

Таким образом, данный web-сервис после расширения библиотеки доступных компонентов позволит проводить анализ схем с большим количеством нелинейных компонентов (таких, как, например ключи) любому пользователю, имеющему доступ к сети Интернет.

Основными преимуществами данного подхода в реализации этой системы являются:

1. Доступность – отсутствует необходимость в приобретении ПО и аппаратного обеспечения соответствующей мощности.
2. Масштабируемость – вне зависимости от сложности моделируемой схемы, будет выделено необходимое количество ресурсов (при условии размещения в облачном хостинге).
3. Оперативность – возможность выполнения моделирования в любой момент.

#### Библиографический список

1. Богданов К. В., Ловчиков А. Н. Построение EDA-системы на основе синхронизированных параллельных процессов // Вестник СибГАУ. 2009. Вып. 4 (25). Красноярск. С. 58–61.
2. Богданов К. В., Ловчиков А. Н. Архитектура EDA-системы на основе конкурирующих параллельных процессов // Известия вузов: Приборостроение. 2011. № 4. С. 63–67.
3. Хайнеман Р. PSPICE. Моделирование работы электронных схем. М.: ДМК Пресс, 2005.
4. Риз Д., Облачные вычисления. СПб.: БХВ-Петербург, 2011.
5. Чезарини Ф., Томпсон С. Программирование в Erlang. 2012.

6. Норенков И. П. Основы автоматизированного проектирования. М.: Изд-во МГТУ им. Н. Э. Баумана, 2002.

7. Численные методы, параллельные вычисления и информационные технологии: сб. научн. тр. / под ред. В. В. Воеводина, Е. Е. Тыртышников. М.: Изд-во Москов. ун-та, 2008.

#### References

1. Bogdanov K. V., Lovchikov A. N. Vestnik SibGAU. 2009, № 4(25), p. 58–61.
2. Bogdanov K. V., Lovchikov A. N. Izvestiya vuzov: Priborostroyeniye. 2011, № 4, p. 63–67.
3. Hineman R. PSPICE. Electronically schemes modeling. DMK. Moscow, 2005, 336p.
4. Riz D. Cloud computing. BHV, St-Petersburg, 2011, 288 p.
5. Chezarini F., Thompson S. Erlang programming. 2012, 488 p.
6. Norenkov I. P. CAD basics // MGTU, Moscow, 2002, 336 p.
7. Numerical methods, parallel computing and information technologies: scientific works compilation. Under redaction V. Voevodin, E. Tyrtysnikov, MGU, Moscow, 2008, 320 p.

© Богданов К. В., Ловчиков А. Н., 2013

УДК 681.5:517.551

### К ВОПРОСУ ФУНКЦИОНАЛЬНОЙ АППРОКСИМАЦИИ НЕЛИНЕЙНЫХ ХАРАКТЕРИСТИК ОБЪЕКТОВ СИСТЕМ АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ НЕПРЕРЫВНЫХ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ\*

А. В. Большевичус<sup>1</sup>, И. В. Ковалев<sup>2</sup>, В. В. Лосев<sup>1</sup>

<sup>1</sup>Сибирский государственный технологический университет  
Россия, 660049, г. Красноярск, просп. Мира, 82. E-mail: bolsavichus@gmail.com

<sup>2</sup>Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева  
Россия, 660014, г. Красноярск, просп. имени газеты «Красноярский рабочий», 31  
E-mail: rector@sibsau.ru

*Предложен подход к построению динамической модели объекта системы автоматического регулирования путем идентификации функции объекта и анализа его нелинейных характеристик, с последующим изучением численных характеристик полученных случайных величин. Применены прикладные программные средства, обеспечивающие автоматизацию задач исследования на основе достоверных методов (Гаусса-Ньютона). В итоге обеспечена возможность оценки степени достоверности используемой передаточной функции в задаче функциональной аппроксимации нелинейных характеристик объекта, а также получены количественные значения показателя проявления статической характеристики из временного ряда полной продолжительности.*

*Ключевые слова: технологический процесс, объект, передаточная функция, случайная величина, математическое ожидание, аппроксимация.*

\*При поддержке Министерства образования и науки Российской Федерации в рамках государственного задания, проект № 8.5534.2011 «Модели, методы и алгоритмы синтеза и управления развитием отказоустойчивых программных архитектур распределенных информационно-телекоммуникационных систем».