

# ВЕБ-СИСТЕМА РАЗРАБОТКИ И ВНЕДРЕНИЯ РАСПРЕДЕЛЕННЫХ ВИРТУАЛЬНЫХ ИСПЫТАТЕЛЬНЫХ СТЕНДОВ

*Е.С. Худякова, Г.И. Радченко*

*Южно-Уральский госуниверситет, Челябинск*

Для суперкомпьютерного моделирования сложных технологических процессов требуется организация потока задач, объединяющего несколько систем инженерного моделирования. Непосредственное решение такой задачи требует от инженера знаний по организации расчетов в распределенной вычислительной системе (РВС). Технология распределенных виртуальных испытательных стендов (РаВИС) позволяет скрыть от пользователя особенности реализации РВС, обеспечивая прозрачное моделирование инженерных задач. Предлагается веб-приложение для разработки РаВИС. С его помощью разработчик может описать параметры РаВИС, определить потоки задач в виде визуальных диаграмм, осуществить шаблонизацию файлов постановки инженерной задачи.

## **Введение**

На сегодняшний день для решения задач, связанных с разработкой сложных технологических процессов, широкое применение нашли системы компьютерного моделирования (САЕ - Computer Aided Engineering). Благодаря таким системам стало возможным проводить виртуально те эксперименты, которые нельзя или трудно осуществить реально (высокая стоимость, небезопасность и др.) [7]. Однако использование систем САЕ также сопровождается рядом трудностей. Как правило, технологический цикл решения задачи включает в себя несколько этапов, которые соединены в нетривиальный нелинейный поток задач, выполняемый в рамках распределенной вычислительной системы. Процесс решения инженерной задачи можно разбить на следующие типичные этапы: формирование геометрии задачи, генерация вычислительной сетки, определение граничных условий, проведение компьютерного моделирования, визуализация, анализ результатов решения. Таким образом, для рядового инженера описание и запуск процесса инженерного моделирования с использованием одного или нескольких инженерных пакетов в распределенной вычислительной системе (РВС) может потребовать глубоких знаний в области системного программирования [5, 6].

В грид-сообществе было разработано два основных метода описания потоков задач: скриптовые языки и графы [8]. Например, в системе CLAVIRE [1] для задания потока задач используется предметно-ориентированный язык EasyFlow. Интерфейс системы представляет поле для ввода скрипта на данном языке, в другой части автоматически генерируется диаграмма потока задач. Такой подход является не очень удобным, поскольку для его использования необходимо изучение нового языка описания процессов. В системе MathCloud [2] пользователю предоставляется возможность непосредственного изображения потока задач в виде графа. И хотя граф изначально является чисто математическим понятием, он обладает очень интуитивным способом визуализации, с которым может работать каждый. Недостатком этой системы можно назвать неудачное решение представления данных в виде отдельных вершин графа, что перегружает диаграмму. Существуют и другие подходы к заданию потоков задач. В проекте Taverna [9] поток задач представляется для пользователя в трех видах (граф, xml-код,

иерархическая структура), однако редактирование потока задач возможно только в блоке представления потока задач в виде иерархической структуры. Во всех описанных системах разработчики ограничивают пользователей данными изначально сервисами, отсутствует возможность модификации и параметризации исходных файлов постановки задач, что позволило бы существенно увеличить гибкость систем.

Технология внедрения ресурсов САЕ-пакетов в распределенные вычислительные среды, предлагаемая в работе [3], основывается на концепции распределенного виртуального испытательного стенда. Распределенный виртуальный испытательный стенд (РаВИС) – это программное решение, обеспечивающее проведение работ инженерного моделирования задачи в распределенной вычислительной среде с грид-инфраструктурой. РаВИС регламентирует процесс декомпозиции задачи в иерархию подзадач; поиск вычислительных ресурсов; сопоставление задачам соответствующих базовых компонентов САЕ-систем; мониторинг хода решения задач; передачу результатов решения задач пользователю.

В данной статье описывается веб-приложение DiVTB Developer, обеспечивающее разработку распределенных виртуальных испытательных стендов: визуальное проектирование потока задач для проведения моделирования, шаблонизацию файлов постановки задачи, формирование набора параметров моделируемой задачи. Обеспечивается экспорт разработанного РаВИС в систему DiVTB Server для предоставления его конечным пользователям.

## 1. Архитектура DiVTB Developer

На рис. 1 изображена диаграмма вариантов использования системы DiVTB Developer. Пользователь системы, разработчик РаВИС, может создать РаВИС с нуля на базе имеющихся в системе сервисов, описав необходимый поток задач. Разработчик может импортировать РаВИС в систему, загрузив XML-файлы его описания. Сохранив РаВИС, пользователь может в любой момент открыть его для редактирования. Кроме того, инженер может экспортировать проект в систему DiVTB Server [4], откуда возможен запуск расчета задачи в распределенной вычислительной среде.

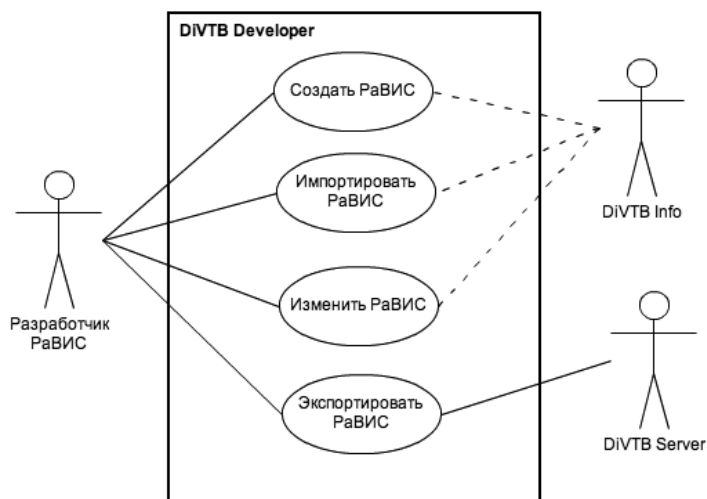


Рис. 1. Диаграмма вариантов использования системы DiVTB Developer

Для создания, импорта и изменения РаВИС необходима информация о существующих сервисах, их параметрах. Данную информацию предоставляет служебная система DiVTB Info.

Рассмотрим архитектуру системы DiVTB Developer (рис. 2). Разработчик РаВИС взаимодействует с *модулем пользовательского интерфейса*, при помощи которого со-

здает диаграмму потоков задач РаВИС. В *модуле управления проектом* поток задач обрабатывается и сохраняется в *хранилище проектов*. Файлы, необходимые для запуска расчета в конкретном инженерном пакете, загружаемые пользователем во время формирования потока задач, сохраняются в *хранилище файлов постановки задач*. *Модуль взаимодействия с DiVTB Server* позволяет инженеру загрузить проект на сервер, с которого возможен запуск проекта. Система *DiVTB Info* предоставляет системе DiVTB Developer информацию об имеющихся в РВС доступных сервисах (установленных на узлах РВС пакетах инженерного моделирования), имеющихся лицензиях на САЕ-пакеты, а также о параметрах каждого сервиса.

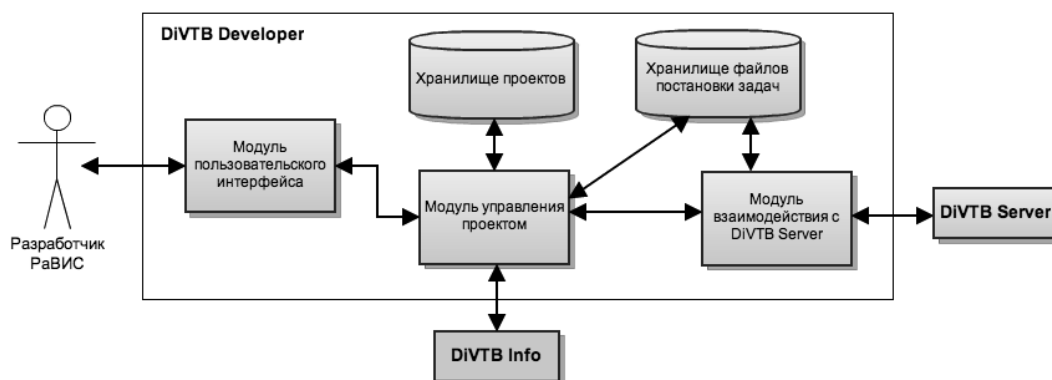


Рис. 2. Архитектура системы DiVTB Developer

Пользовательский интерфейс системы состоит из трех логических блоков (рис. 3): блок определения потока задач, блок управления процессом, блок редактирования файлов постановки задач.

*Блок определения потока задач* (рис. 4а) необходим для построения диаграммы потока задач – набора процессов, основанных на существующих сервисах, и дуг управления, их соединяющих. В *блоке управления процессом* (рис. 4б) происходит определение процесса, а именно:

- задание имени, словесное описание семантики процесса;
- выбор вычислительного сервиса (как системного, так и пользовательского);
- загрузка файлов постановки задачи для конкретного сервиса (САЕ-пакета);
- задание значений входных параметров процесса.

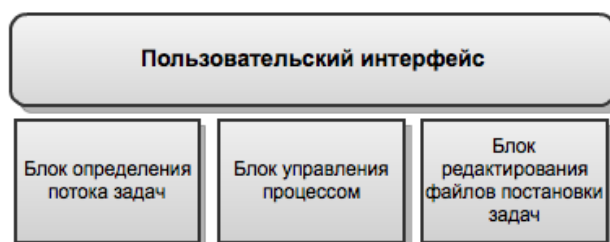


Рис. 3. Логические блоки пользовательского интерфейса системы

Под пользовательским сервисом подразумевается РаВИС, созданный инженером ранее. Он, наряду с системным сервисом (добавленным в систему администратором), может выступать в качестве блока потока задач, таким образом, реализуя идею вложенного процесса.

*Блок редактирования файлов* постановки задач позволяет изменять *проблемно-ориентированные файлы*. Данные файлы описывают задачу и необходимы для запуска расчета в определенном САЕ-пакете. Пользователь может динамически менять файл, вставляя в текст специальный код с использованием входных параметров текущего

процесса в качестве переменных. Такой процесс назовем *шаблонизацией* проблемно-ориентированного файла. Таким образом, создав проблемно-ориентированный файл один раз для определенной задачи, пользователь может описать РаВИС, решающий целый класс задач.

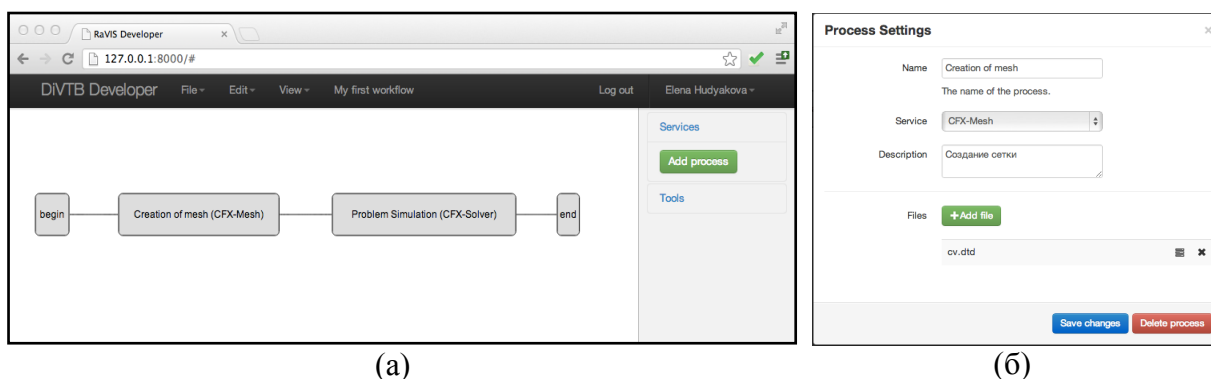


Рис. 4. Интерфейс системы: (а) основное окно редактирования потока задач, (б) окно редактирования процесса

## 2. Реализация системы

Система реализована в виде веб-приложения. Серверная часть приложения реализована на языке программирования Python с использованием фреймворка Django [10]. Данный фреймворк обеспечивает объектно-реляционное отображение данных, а его система шаблонов, разделяющая логику и представление информации, позволяет быстро и эффективно реализовывать необходимый функционал.

Клиентская часть реализована на стеке технологий HTML/CSS/JavaScript. Поскольку DiVTB Developer обеспечивает быструю работу с РаВИС, пользовательский интерфейс играет важную роль в данной системе. Поэтому при разработке интерфейса особое внимание уделялось его интуитивности и наглядности. Для этого использовался ряд библиотек. Для создания элементов пользовательского интерфейса используется JavaScript-библиотека Twitter Bootstrap [11]. Для рисования диаграммы потока задач используется JavaScript-библиотека Raphael [12].

В качестве шаблонизатора используется Java Minimal Template Engine (JMTE) [13]. JMTE – шаблонизатор, позволяющий вставлять значения переменных в текст шаблона при помощи простого синтаксиса, подобного String.format. JMTE поддерживает базовые операции с переменными (циклы, условия).

### Заключение

В данной статье рассмотрена архитектура веб-приложения DiVTB Developer, обеспечивающего разработку распределенных виртуальных испытательных стендов. Разработчику предоставляется возможность визуального проектирования потока задач для проведения моделирования, шаблонизации файлов постановки задачи, формирования набора параметров моделируемой задачи. Также система предоставляет функцию экспорта РаВИС в систему DiVTB Server, откуда возможен запуск расчета задачи.

Работа выполнялась при поддержке Совета по грантам Президента РФ (проект МК-1987.2011.9), Российского фонда фундаментальных исследований (проект № 11-07-00478-а) и Министерства образования и науки РФ (государственное задание 8.3786.2011).

## Литература

1. Князьков К.В., Ларченко А.В. Предметно-ориентированные технологии разработки приложений в распределенных средах // Известия вузов. Приборостроение. 2011. Т. 54, № 10. С.36-43.
2. Лазарев И.В., Сухорослов О.В. Реализация распределенных вычислительных сценариев в среде MathCloud // Проблемы вычислений в распределенной среде. Труды ИСА РАН, Т. 46. М.: КРАСАНД, 2009. С. 6-23.
3. Радченко Г.И. Распределенные виртуальные испытательные стенды: использование систем инженерного проектирования и анализа в распределенных вычислительных средах // Вестник ЮУрГУ. Серия "Математическое моделирование и программирование". 2011. № 37(254). В. 10. С. 108-121.
4. Федянина Р.С. CAEBeans Server: среда выполнения проблемно-ориентированных оболочек над инженерными пакетами // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 марта - 2 апреля 2010 г.). Челябинск: Издательский центр ЮУрГУ, 2010. С. 621-628.
5. Buriola T.M., Scheer S. CAD and CAE Integration Through Scientific Visualization Techniques for Illumination Design // Tsinghua Science & Technology. V. 13, N. 1. 2008. P. 26-33.
6. Fox G.C., Gannon D. Workflow in Grid Systems // Concurrency and Computation: Practice & Experience. V. 18. N. 10. 2006. P. 1009-1019.
7. Palloni F. A New Tool for Multidisciplinary CAE Simulations // NAFEMS 2005 World Congress, May 17-20, 2005, Malta.
8. Hoheisel A. User Tools and Languages for Graph-based Grid Workflows // Workflow in Grid Systems Workshop in GGF10, at Berlin, Germany, March 2004.
9. Oinn T., Addis M., Ferris J., Marvin D., Senger M., Greenwood M., Carver T., Glover K., Pocock M.R., Wipat A., Li P. Taverna: a tool for the composition and enactment of bioinformatics workflows // Bioinformatics, 2004. Vol. 20, No. 17. P. 3045-3054.
10. Фреймворк Django – [<https://www.djangoproject.com/>].
11. Библиотека Twitter Bootstrap – [<http://twitter.github.com/bootstrap/>].
12. Библиотека Raphael – [<http://dmitrybaranovskiy.github.io/raphael/>].
13. Шаблоноизатор Java Minimal Template Engine (JMTE) – [<http://code.google.com/p/jmte/>].