



Сафонов Владимир Олегович

ТЕХНОЛОГИИ SUN MICROSYSTEMS И ОТКРЫТОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ: ОПЫТ МАТ-МЕХА

Аннотация

В статье описывается опыт автора по преподаванию и развитию технологий Sun Microsystems – Java, компиляторов Sun Studio, ОС Solaris, архитектуры SPARC – и использованию открытого программного обеспечения в преподавании на математико-механическом факультете СПбГУ. Анализируются преимущества открытого программного обеспечения. Сформулированы рекомендуемые направления дальнейшей работы для преподавателей ИТ. Намечены перспективы новых технологий. Даны ссылки на полезные ресурсы.

Ключевые слова: Sun Microsystems, Java, компиляторы, Sun Studio, операционные системы, Solaris, SPARC, открытое программное обеспечение.

ВВЕДЕНИЕ

Материал данной статьи основан на двух докладах автора на прошедшей в апреле 2009 г. в Санкт-Петербурге конференции Sun Technology Days 2009 [1] и на идеях и впечатлениях автора, возникших при обсуждении на круглом столе для преподавателей (на конференции) об использовании свободного программного обеспечения. Считаю необходимым в рамках расширенной дискуссии на эту тему высказать свое мнение и приглашаю коллег продолжить это обсуждение, тем более что мы стоим на пороге важных административных решений в стране о переходе в бюджетных вузах к преподаванию на основе свободного программного обеспечения. Статья продолжает серию статей автора о преподавании ИТ, начатую статьями [2] и [3].

© В.О. Сафонов, 2009

1. ИСТОРИЯ ПРОЕКТОВ ПО РАЗВИТИЮ ТЕХНОЛОГИЙ SUN И ИХ ПРЕПОДАВАНИЯ НА МАТЕМАТИКО-МЕХАНИЧЕСКОМ ФАКУЛЬТЕТЕ СПбГУ

Чтобы было понятно, почему столь велик интерес автора к технологиям Sun и их преподаванию, приведу некоторые факты из истории мат-меха и нашего сотрудничества с Sun.

- 1992, июль. Начат первый проект для фирмы Sun на мат-мехе по развитию и сопровождению компилятора Sun Pascal для платформы SPARC/Solaris. Проект выполнялся тремя сотрудниками под моим руководством.

- 1992, сентябрь. Благодаря совместным проектам с Sun, наша группа получила первую SPARC-станцию. Это была, по нынешним понятиям, не очень мощная машина: SPARC 1, память 32 МВ, операционная система – Solaris 1, развитие ОС

UNIX Berkeley Software Distribution – BSD. Но для нас эта станция была необыкновенно важна – мы начали на практике осваивать Solaris и применять наш к тому времени уже достаточно большой опыт разработки компиляторов для МК «Эльбрус» [4] к сопровождению компилятора SPARCCompiler Pascal (нынешнее название компиляторов Sun – Sun Studio).

- 1992–1996. Было исправлено более 500 ошибок в компиляторе и выпущено четыре его новых версии. Была выполнена уникальная работа по применению разработанных мной эффективных методов семантического анализа к компилятору Pascal фирмы Sun, благодаря чему удалось ускорить front-end компилятора в 4 раза без коренных переделок остального кода front-end'a. Методы, примененные в данном проекте, защищены четырьмя опубликованными патентами США и России по новым методам компиляции. Очень важным стал для нас также опыт удаленной работы в команде и опыт взаимодействия с сотнями пользователями компилятора из разных стран и фирм мира. Это были бесценные уроки на всю жизнь – этот опыт я передаю своим ученикам.

- 1993. Начато преподавание курса по компиляторам с использованием материалов по SPARC, компиляторам Sun и новым методам компиляции. В своем курсе я даю много интересной информации о компиляторах Sun Studio, которые выполняют уникальный набор оптимизаций. У истоков компиляторов Sun стоит такой известный специалист по компиляторам и оптимизации программ, как Steven Muchnick, автор широко известной монографии по оптимизации в компиляторах [5]. Это человек, с которым мне посчастливилось общаться и работать; он интервьюировал меня при приеме в команду для совместных работ с Sun в 1992 г. Преподавание раздела по генерации кода в моем курсе построено на основе изучения основ архитектуры и системы команд SPARC, и все примеры генерации кода даны для данной архитектуры. Это существенно дополняет и расширяет знания студентов, которые из языков ассембле-

ра, главным образом, изучают ассемблер платформы x86.

- 1996, июль. Начаты первые проекты для Sun по Java-технологии:

- участие в разработке Java Compatibility Kit (JCK) – крупного комплекса тестов для сертификации сторонних реализаций Java, поставляемого фирмой Sun вместе с лицензией на реализацию;

- участие в развитии технологии Java Beans, которая в тот момент только что появилась. Трудно передать мои ощущения, когда я уже осенью 1996 г. держал в руках спецификацию Java Beans (ее автор – Graham Hamilton, Sun distinguished engineer).

- В 1996 г., через два месяца после того как мы начали заниматься Java-проектами, я начал преподавание Java-технологии на мат-мехе – сначала в рамках спецкурса по технологии программирования, затем стал читать отдельный спецкурс «Язык программирования Java» для 3 курса, а позднее – также вести спецсеминар для 3 курса «Java-технология». Интерес студентов к этим занятиям очень велик. Курс по Java ежегодно посещают и сдают более 100 студентов, семинар по Java – 30–40 студентов. Курс ежегодно обновляется, так что подчас студенты слушают некоторые лекции на следующий год снова, настолько интересны для них новые возможности Java.

- 1996–2002. Группа Sun на мат-мехе под моим руководством выросла до 75 человек. Стартовало много новых проектов: развитие компиляторов C и C++; инструментальные комплексы тестирования для «микроизданий» Java – J2ME TCKs; развитие и тестирование среды разработки NetBeans (сама среда появилась в 1997 г., а мы начали проекты по ее развитию в 1998 г.); сопровождение Abstract Window Toolkit (AWT) и Swing – Java-инструментариев для поддержки графических пользовательских интерфейсов; сопровождение и тестирование KVM (виртуальной машины Java с малым потреблением памяти для микропроцессоров) и JavaCard (библиотек на Java для программирования smart-

карт); сопровождение и развитие браузера HotJava (написанного полностью на Java) и его аналога для Web-телефонов – WebPhone browser – и других. Таким образом, в течение нескольких лет группа получила уникальный в России опыт по освоению, использованию и развитию самых разнообразных инструментов Java-технологии.

- 1999. На мат-мехе создан один из первых в России учебных классов SPARC-станций, подаренных университету фирмой Sun. Класс с интересом посещали не только студенты, но и многие преподаватели. В настоящее время две современных рабочих станции Sun – SPARC Ultra 20 и Ultra 25 – установлены в моей лаборатории Java-технологии и используются для проведения совместных работ с Sun.

- 2002. Вышла моя книга «Введение в Java-технологию» [6]. Книга является базовым учебником по технологии, рекомендованным Минобразования для студентов ИТ-специальностей. Подчеркну *научный* характер книги (что, к сожалению, в настоящее время достаточно редко в литературе по программированию, где преобладают, по сути дела, публикации инженерно-технического и рекламного характера, что не вполне соответствует духу университетского образования). В книге проанализированы основы Java-технологии, даны наглядные примеры, проведен сравнительный анализ с другими технологиями, намечены перспективы развития. Объем книги – около 190 страниц, что значительно меньше, чем у большинства других книг по программированию (более традиционен очередной американский «кирпич» объемом 800–1000 страниц, уместный, главным образом, как справочное издание ввиду его огромного объема). Поэтому студентам или другим новичкам в области Java-технологии вполне реально (и настоятельно рекомендуется) *полностью* прочитать мою книгу, изучить, откомпилировать и выполнить приведенные в ней примеры программ на языке Java, а затем перейти к дальнейшему практическому освоению Java-технологии на

основе учебных проектов и изучения документации. Книга до сих пор популярна среди студентов крупных городов России, ее материал нисколько не устарел.

- 2004. *Создан Центр Высоких технологий Sun в Санкт-Петербурге.* Я очень горд тем, что на основе группы Sun на мат-мехе, успешно работавшей более 10 лет, фирма Sun в 2004 году создала Центр высоких технологий в Петербурге – отделение фирмы Sun. Среди менеджеров и инженеров этого Центра много моих учеников. Лаборатория Java-технологии поддерживает сотрудничество с Центром высоких технологий Sun и занимается совместными с ним проектами. Среди них – разработка утилиты *Make2NetBeans* для конвертирования традиционных проектов на C / C++ (на основе утилиты *make*) в проекты NetBeans C / C++ Development Pack; разработка поддержки языка UML для C++ в NetBeans C / C++ Development Pack.

Материалы моих учебных курсов «Язык программирования Java», «Разработка компиляторов» и спецсеминара «Java-технология» опубликованы на русском академическом сайте фирмы Sun [7].

Операционная система Solaris преподается в рамках курса «ОС и сети» для студентов 2 курса. Об этом я уже подробно рассказывал в работе [3].

2. ОПЫТ И АНАЛИЗ ИСПОЛЬЗОВАНИЯ ОТКРЫТОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПРЕПОДАВАНИИ

Для преподавания и исследовательских проектов в области Java-технологии мы используем продукт Sun с открытым исходным кодом – среду разработки NetBeans, а также открытую среду Eclipse для разработки приложений на Java. Студенты разрабатывают расширения (plugins) к этим средам, в том числе – утилиты и игры.

В 2005 г. фирма Sun начала проект по созданию версии JDK с открытым исходным кодом – *openJDK*. Данный проект предоставляет студентам возможность участвовать в развитии Java-технологии – реализовать какую-либо новую черту языка Java,

новые интересные библиотеки и т. д. Однако студенты, несмотря на их очень большой интерес к проекту, пока в нем не участвуют, по-видимому считая его слишком сложным. Надеюсь, что в ближайшее время это положение исправится.

Другой open source – проект Sun OpenSolaris [3, 8] – также находится в области интересов наших студентов, но пока ощутимых результатов нет – видимо, также ввиду сложности архитектуры проекта. В этом отношении следует отметить работу Политехнического университета по переносу в среду OpenSolaris утилиты управления BluRay – дисками, удостоенную гранта Sun.

Компиляторы Sun Studio изучаются в моих спецкурсе и спецсеминаре по компиляторам для студентов 5 курса. Хотелось бы пожелать фирме Sun изменить статус продукта Sun Studio на open source (по аналогии с известными компиляторами GNU). Насколько мне известно, работа в этом отношении в фирме Sun ведется. Считаю компиляторы Sun Studio и примененные в них методы классикой в данной области, которую следует изучать с активным использованием исходных кодов.

Что касается open source – продукта Sun OpenOffice, то его используют студенты при подготовке презентаций. Более широкое использование данного продукта в учебном процессе и исследованиях вполне возможно, но для этого требуется соблюдение разработчиками OpenOffice стандартов совместимости с файлами Microsoft Office: например, при открытии в OpenOffice презентации PowerPoint 2003 (.ppt) подчас имеют место дефекты воспроизведения – например, обрезаются символы разбиения слайда на пункты (bullets), что недопустимо. Надеемся, что эти недостатки будут исправлены в следующих версиях OpenOffice.

Преимущества открытого программного обеспечения

Прежде всего, благодаря открытому программному обеспечению, студенты приобретают практические знания и навыки на базе *реальных*, а не «игрушеч-

ных» проектов, а также навыки удаленной работы в распределенной команде и навыки профессионального общения. На мой взгляд, студентов необходимо учить «в боевой обстановке», то есть на реальных проектах.

Но наиболее важна роль открытого ПО в том, что оно обеспечивает личную сопричастность студентов к развитию технологий. Об этом я уже писал в работе [2]. Студент, разработавший какой-либо новый модуль к продукту open source и опубликовавший его на сайте проекта, чувствует себя участником интересного процесса развития новой технологии и видит свои идеи воплощенными в серьезном проекте. Он может, благодаря этому, приобрести интересные и полезные контакты – например, получить предложение о приеме на работу или стажировку. С этой точки зрения, открытое ПО – широкое поле деятельности для обмена интересными идеями на уровне исходных кодов и реализаций.

Таким образом, *открытое ПО – уникальная основа для профессионального роста студентов, аспирантов и преподавателей вузов.*

Необходимость использования не только открытого и свободного ПО

Инициатива по переходу вузов на использование свободного программного обеспечения таит в себе определенную опасность при неправильной трактовке (например, не исключено запрещение использования в вузах программного обеспечения, не являющегося открытым или свободным, что было бы, на мой взгляд, совершенно неправильно). Я твердо убежден, что студентов необходимо учить принципам, архитектуре и корректному использованию *любого* ПО (а не только открытого или свободного), в том числе, например, столь широко распространенного ПО фирмы Microsoft – в первую очередь, Windows, Visual Studio, Microsoft Office. Другой важный пример ПО, не являющегося свободным и не имеющего статуса open source, – ПО фирмы Adobe (Illustrator, Acrobat, Photoshop и др.). Без

фундаментальных знаний о таком программном обеспечении и без опыта его использования современного специалиста по ИТ представить себе невозможно. Возможно, со временем Microsoft и Adobe примут решение об изменении статуса своего программного обеспечения. Определенные тенденции к этому налицо: например, как я уже писал, Microsoft организовала программы Windows Academic Shared Source и Shared Source Common Language Infrastructure (Rotor), позволяющие студентам и преподавателям изучить и даже развивать в учебных и исследовательских целях исходный код академических версий Windows и .NET. Однако коммерческое ПО, не являющееся свободным или open source, существует и широко используется, и наша задача – научить студентов правильно его использовать, в том числе – соблюдать лицензии, не покупать и не использовать пиратских версий и т. д. Это повысит культуру программирования и использования компьютерных технологий в нашей стране.

Кроме того, для фундаментального обучения ИТ, по моему глубокому убеждению, весьма важен *сравнительный анализ* различных современных технологий – как open source, так и иных.

Наконец, коммерческое ПО весьма важно для образования и потому, что на положительных примерах организации, распространения, маркетинга и рекламы коммерческого ПО ведущими фирмами мира студенты могут учиться *корректному ведению бизнеса в области ИТ*.

Студенческие и аспирантские проекты по технологиям Sun

Считаю важной целью своей деятельности организацию учебных проектов на основе технологий Sun. Вот лишь некоторые примеры тем аспирантских и студенческих работ, заинтересованность молодых программистов в которых очень велика:

- «Балансировка загрузки серверов приложений JEE на основе обучения с подкреплением – reinforcement learning».

- «Aspect.Java – разработка инструментария АОП для платформы Java по принципам Aspect.NET».

- «Разработка Java API для создания эффективных приложений для многоядерных процессоров фирмы Sun» (результаты данного проекта удаленно тестировались на мощном 16-ядерном сервере UltraSPARC-T1 – Niagara в Центре высоких технологий Sun).

- «Расширения языка Java для представления знаний».

- «Сборник задач по Java» (задачи для этого сборника составляются моими аспирантами под моим руководством и предлагаются для решения студентам при сдаче экзамена по моему курсу «Язык программирования Java»).

В результате выполнения подобных работ некоторые мои студенты и аспиранты становятся стажерами, а затем и постоянными сотрудниками Sun. Считаю такое сотрудничество очень перспективным, как и сотрудничество по совместным исследовательским проектам.

ЗАКЛЮЧЕНИЕ

Java-технология – замечательный пример *интенсивного развития ИТ*. Наиболее существенные нововведения в Java за недавние годы: *параметризованные типы (generics); перечисления (enumerations); аннотации; язык JavaFX* для разработки скриптов и поддержки создания GUI и полнофункциональных Интернет-приложений (*Rich Internet Applications – RIA*). Все эти новинки находят отражения в моих учебных курсах и семинарах, что требует их постоянного изучения, а иногда и непосредственного общения с разработчиками.

Кроме того, развиваются *стандарты ИТ-образования*, причем не только отечественные, но и зарубежные, например ACM Curricula 2001/2004; GSwERC (Graduate Software Engineering Reference Curriculum) – в обсуждении последнего я принял активное участие. Зарубежные стандарты подчас отличаются от наших, с одной стороны, большей ориентацией на

бизнес в области ИТ, с другой – не столь глубокими математическими основами. Например, в стандарте GSwERC в качестве необходимого математического фундамента отсутствуют такие важнейшие дисциплины, как алгебра и анализ, а теория вероятностей представлена только в дискретном варианте. Опять-таки наша задача – постоянно быть в курсе развития образовательных стандартов, но подходить к их внедрению разумно и осмотрительно.

Весьма важную роль для образования играет *развитие новых подходов и парадигм архитектуры и разработки программ: «облачные» вычисления (cloud computing), виртуализация, 3D Web, надежные и безопасные вычисления (trustworthy computing), параллельное программирование на базе многоядерных процессоров, аспектно-ори-*

ентрированное программирование [9], интеграция методов инженерии знаний, данных и программ (например Semantic Web) и др.

В связи с рассмотренными важнейшими особенностями динамичного развития нашей области, необходимо *постоянное обновление учебных курсов и семинаров, а для этого – постоянное совершенствование знаний преподавателей ИТ и их участие в разработках, в том числе с открытым кодом.*

Приглашаю уважаемых коллег к дискуссии и прошу присылать Ваши отзывы по электронной почте: v_o_safonov@mail.ru. Многие интересные учебные материалы, имеющие отношение к тематике данной статьи, доступны через сайт лаборатории Java-технологии математико-механического факультета СПбГУ [10].

Литература

1. Web-страницы международной конференции Sun Technology Days 2009: <http://ru.sun.com/techdays/>
2. Сафонов В.О. Актуальные проблемы преподавания технологий программирования в России // Компьютерные инструменты в образовании, 2008, № 5.
3. Сафонов В.О. «Золотой век» операционных систем и преподавание ОС в университетах России // Компьютерные инструменты в образовании, 2009, № 1.
4. Сафонов В.О. Языки и методы программирования в системе «Эльбрус». М.: Наука, 1989.
5. Michnick S. Advanced compiler design and implementation. Morgan Kaufmann, 1997.
6. Сафонов В.О. Введение в Java-технологии. СПб.: Наука, 2002.
7. Сафонов В.О. Материалы учебных курсов «Язык программирования Java» и «Разработка компиляторов» и спецсеминара «Java-технология» на сайте Sun: <http://ru.sun.com/teachingmaterials/>
8. Web-сайт проекта OpenSolaris: <http://opensolaris.org/os/>
9. Safonov V.O. Using aspect-oriented programming for trustworthy software development. Wiley Interscience. John Wiley & Sons, 2008.
10. Web-сайт лаборатории Java-технологии мат-мех. факультета СПбГУ: <http://polyhimnic.math.spbu.ru/jtl/>

Abstract

The article describes the author's experience of teaching Sun Microsystems' technologies – Java, Sun Studio compilers, Solaris OS, SPARC architecture – and of using open software in teaching at the Faculty of Mathematics and Mechanics of St. Petersburg University. Advantages of open software are analyzed. Directions for future work of IT teachers formulated; perspectives of new technologies outlined; references to useful resources provided.

Сафонов Владимир Олегович,
доктор технических наук,
профессор кафедры информатики
СПбГУ, руководитель лаборатории
Java-технологии,
v_o_safonov@mail.ru



Наши авторы, 2009.
Our authors, 2009.