

А. Н. Гребнев

AtLear - Java каркас с открытым исходным кодом для Web приложений

В работе рассматривается спроектированное автором и разработанное под руководством автора, веб приложение AtLear – Java каркас с открытым исходным кодом для быстрого построения на его основе веб приложений. Объясняются причины выбора основных библиотек используемых при построении. Особое внимание уделено рассмотрению основной функциональности AtLear – системе управления содержимым, на основе концепции разработанной автором.

Ключевые слова: веб приложение, открытый исходный код (open-source), система управления содержимым.

1 Введение

Кроссплатформенный объектно-ориентированный язык программирования Java™ от Sun Microsystems Inc. и огромное количество технологий построенных на базе этого языка уже прочно заняли свою нишу во многих направлениях индустрии разработки программного обеспечения (ПО). Настольные (desktop) приложения на платформе Java не во всех случаях могут соперничать с приложениями на платформе .NET от Microsoft Corp., что же касается Web приложений то здесь явно прослеживается преимущество Java, на нее уже давно сделали ставку такие гиганты ИТ индустрии как Sun, IBM, Bea, Oracle, NEC и др. Однако их решения на J2EE (Java 2 Enterprise Edition) платформе в большинстве случаев предназначены для «больших» корпоративных задач и неоправданно дороги, сложны и тяжеловесны для задач средних и малых (~ до 1000 чел./часов.) [1]. С самого начала появления, Java органично соединяется с идеологией программного обеспечения с открытым кодом (open-source), которая на примере операционных систем Linux или Apache HTTP сервера уже доказала свою конкурентоспособность с коммерческими решениями. На сегодняшний день

существует достаточное количество отдельных open-source технологий [2], которые в совокупности дают возможность создания продукта способного конкурировать с коммерческими J2EE решениями на средних и мелких задачах.

2 Обзор

В середине 2004 года был инициирован проект AtLeap – 100% Java каркас (framework) с открытым исходным кодом (под лицензией Apache 2.0) для быстрого создания Web-приложений на его основе. AtLeap включает в себя CMS (Систему Управления Содержимым СУС), позиционирован на мелкие и средние задачи, основан на лучших open-source технологиях, построен с учетом современных практик и паттернов проектирования. Проект AtLeap (<https://atleap.dev.java.net/>) разрабатывается на сайте java.net сообщества (в своем роде аналог freshmeat.net или sourceforge.net). Изначально в качестве базы для AtLeap был использован open-source проект AppFuse (<https://appfuse.dev.java.net/>) Мет Рейбла, автора многих публикаций, книги «Spring in Live», члена экспертной группы J2EE 1.5.

AtLeap это веб приложение работающее на основе некоторых стандартов J2EE 1.4 платформы включая: JSR 152: Java Server Pages™ 2.0 и JSR 154: Servlet™ 2.4. AtLeap имеет многослойную структуру:

- Персистентный слой (Persistence layer) – слой, обеспечивающий отображение (mapping) объектной Java модели в таблицы реляционной СУБД, так же обеспечивает поддержку транзакций.
- Слой бизнес логики (Business logic layer) – слой, реализующий основную бизнес логику приложения.
- Слой представления (Presentation layer) – слой, обеспечивающий обработку HTTP запросов, переходы между страницами и т.п.

- Слой генерации пользовательского интерфейса (ПИ) (UI generation layer) – слой, обеспечивающий генерацию HTML страниц.

AtLeap не реализует стандарт JSR 153: Enterprise JavaBeans™ 2.1, поэтому для его работы не требуется сервер приложений, например JBoss, Weblogic, WebSphere. Для запуска AtLeap достаточно лишь servlet-контейнера, например Apache Jakarta Tomcat (<http://jakarta.apache.org/tomcat/>). Таким образом, AtLeap обеспечивает независимость:

- от аппаратной платформы и операционной системы, за счет использования Java машины;
- от СУБД, за счет использования персистентного слоя;
- от servlet-контейнера, за счет использования J2EE стандартов.

3 Слой представления

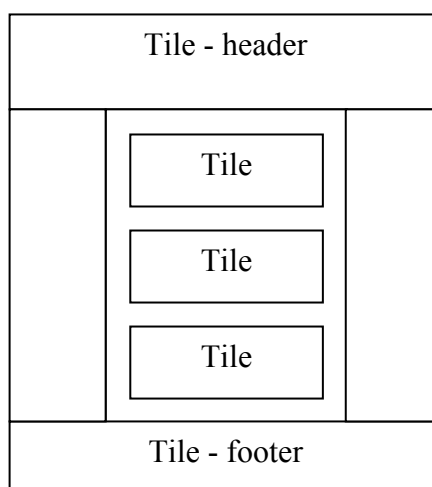
Ядром СУС в AtLeap является библиотека Apache Jakarta Struts (<http://struts.apache.org/>), реализующая вторую модель (Model 2) MVC (Model View Controller) паттерна проектирования. На данный момент на рынке open-source продукции представлено два основных конкурента Struts, реализующих MVC паттерн, это Spring (<http://www.springframework.org/>) и OpenSymphony WebWork (<http://www.opensymphony.com/webwork/>). Эти продукты появились на рынке не так давно, в сравнении со Struts разрабатываемой с 2001 года. Со Struts уже имеют интеграцию десятки зарекомендовавших себя на рынке продуктов, ее конкуренты подобным погордиться пока не могут. Второй не мало важной причиной выбора в пользу Struts было наличие гораздо большего количество программистов имеющих опыт разработки на ее основе.

За счет реализации MVC паттерна Struts обеспечивает разделение логики и представления. Основной сущностью Struts является понятие *Action* (*действие*). Action – это java метод, который инкапсулирует бизнес логику

приложения, обычно он подготавливает данные полученные из нижних слоев, для отображения на странице. Struts имеет богатую библиотеку тегов для генерации форм ввода данных и возможности для их автоматической валидации как на стороне сервера, так и на стороне клиента.

4 Генерация пользовательского интерфейса

Для генерации веб страниц используется библиотека шаблонов Tiles входящая в поставку Struts. Основной сущностью Tiles является Tile (плитка), «строительный» блок, из которых складывается результирующая страница. Плитки могут вкладываться друг в друга и наследоваться, переопределяя вложенные плитки. Плитки описываются в XML файле, каждой плитке может соответствовать шаблон (template) JSP файл, описывающий размещение вложенных плиток с помощью тегов. Для плитки можно объявить действие (Action) который будет выполняться перед ее отображением.



Альтернативой для генерации пользовательского интерфейса на основе MVC паттерна была библиотека OpenSymphony SiteMesh (<http://www.opensymphony.com/sitemesh/>). Идея SiteMesh основывается на паттерне проектирования Декоратор (Decorator). SiteMesh перехватывает запрос к любой странице (динамической или статической) запрошенной с веб сервера, анализирует HTML получает от туда содержимое тегов title, body, head, meta, подбирает в соответствии с параметрами запроса (URL, язык и

т.п.) соответствующий декоратор и генерирует результирующую страницу, т.е. «обворачивает» исходную страницу новыми элементами. Декоратором является JSP страница, в которой с помощью тегов указано размещение полученных данных. Таким образом, достигается значительное упрощение смены дизайна, и получение различных «обверток» в зависимости от свойств запроса. SiteMesh так же поддерживает оборачивание включений (includes) реализуемых с помощью своих тегов, т.е. композицию страницы. Однако SiteMesh не поддерживает наследование декораторов и у него отсутствует возможность определить java код предшествующий отображению отдельного блока страницы, поэтому выбор был сделан в пользу Tiles. Если бы к моменту разработки AtLeap существовала хорошая реализация стандарта JSR 168: Portlet Specification, то в комплексе с SiteMesh она составила бы серьезную конкуренцию Tiles.

5 Система управления содержимым

Задача управления содержимым сайта является неотъемлемой при построении любого Web-приложения и самой весомой при разработке информационных сайтов. Прежде чем перейти к обсуждению особенностей главного модуля AtLeap СУС нам понадобится более точно определить исследуемые понятия.

Содержимое (контент) – информация, размещенная на сайте, не зависимо от формы представления (графическая, текстовая, аудио, видео и т.п.) несущая явную смысловую информационную нагрузку. Для примера графическое изображение, примененное к заднему фону сайта, являющемуся частью замысла графического дизайна, к содержимому сайта относить не следует.

Веб приложение – серверное приложение с веб интерфейсом. Часто можно видеть использование понятий веб сайта и веб приложения, как слов синонимов. Однако выделяют и отличия, *веб сайт* является частным случаем веб приложения [3] и обычно подразумевается, что он контентно-

базирующийся (content-based). Веб приложение же обладает двумя отличительными свойствами:

- Связь один к одному между уникальной сессией приложения и каждым посетителем.
- Возможность создавать, манипулировать и перманентно сохранять данные.

Веб приложения обладают большей интерактивностью при взаимодействии с пользователем, они обладают поведением, состоянием, логикой, большей функциональностью, могут обладать аутентификацией, авторизацией, персонализацией пользователей.

Система управления содержимым СУС (CMS Content Management System) – это приложение, позволяющее пользователям, не обладающим специализированными знаниями в области сайтостроения, создавать и модифицировать содержимое сайта. Иногда в понятие СУС включают функцию управления графическим дизайном. Это не совсем правильно, поскольку нет необходимости разрешать пользователям, не обладающим дизайнерским «вкусом» менять отдельные детали графического дизайна (за исключением случаев выбора различных дизайнов целиком). СУС может быть настольным приложением (с online или offline режимом работы), однако, в большинстве случаев (в AtLeap в том числе), это подсистема разрабатываемого веб приложения, поскольку существующие каналы связи вполне могут обеспечить необходимую скорость взаимодействия, а возможности современных браузеров – пользовательский интерфейс. Заметим, что в последнем случае сама СУС является веб приложением и достаточно сложным.

Графический (веб) дизайн – оформление веб страниц сайта, определяющее внешний вид целиком. В большинстве случаев веб дизайн, в конечном итоге, определяется CSS (Cascading Style Sheets) стилями (шрифты, цвета, отступы отдельных элементов на странице и т.п.) и набором HTML

(Hyper Text Markup Language) макетов/шаблонов (layout/template) определяющих размещение элементов (блоков) на странице, по одному макету для каждого типа страницы. К графическому дизайну так же относят изображения (реже текстовую, аудио, видео информацию) являющиеся частью замысла дизайна и не несущие явную смысловую информационную нагрузку или изображения, замена которых потребует изменения остальных элементов дизайна.

Главная задача СУС – это переложить рутинную работу по созданию и модификации содержимого сайта с технических специалистов (*разработчиков*) на более квалифицированных для данной задачи работников (*редакторов*): менеджер содержимого, менеджер по рекламе, лингвист, переводчик или даже представитель заказчика, ведь именно он обладает наиболее точными и оперативными данными в предметной области разрабатываемого сайта. Таким образом, с точки зрения СУС, выделяется две роли: разработчики и редакторы.

На данный момент на рынке систем управления содержимым существует множество продуктов. Сравнение российских СУС можно найти на сайте http://business-site.ru/articles/wsms_strat.htm, сравнение СУС с открытым кодом – <http://www.opensourcecms.com/>. Для начинающих разработчиков может быть интересно исследование [4] в области CMS от Elashkin Research.

6 СУС в AtLeap

Для AtLeap была разработана усовершенствованная концепция СУС, которая аккумулировала лучшие стороны многих подходов построения подобных систем. СУС в AtLeap базируется на библиотеках Struts и Tiles.

Каждая страница СУС генерируется на основе *макета* (Layout), разрабатываемого дизайнером и верстальщиком. Макет соответствует одному определению (definition) плитки (tile), таким образом, макет может собираться из нескольких шаблонов (JSP страниц) путем композиции и

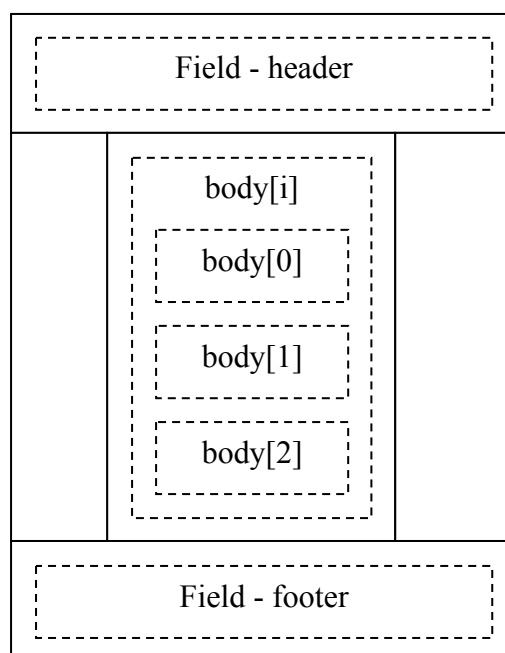
наследования плиток. Редакторам сайта вмешиваться в дизайн не позволительно, разработчики же, по причинам гибкости и скорости работы, предпочтут WYSIWYG редактору непосредственное редактирование кода (HTML или CSS). Следовательно, через HTML редактор необходимо позволять редактировать только отдельные части страницы. Эти места в макете называются *полями* (Field), определяются разработчиками и помечаются уникальными, в рамках макета, идентификаторами. Редакторам создавать новые макеты, а значит и новые типы страниц не разрешено.

Поля бывают 3 типов:

- однострочный текст,
- многострочный текст,
- HTML редактируемый через визуальный WYSIWYG (What You See Is What You Get) редактор.

Поля содержат *значения* (Value) в кодировке UTF-8 для каждого языка зарегистрированного в системе.

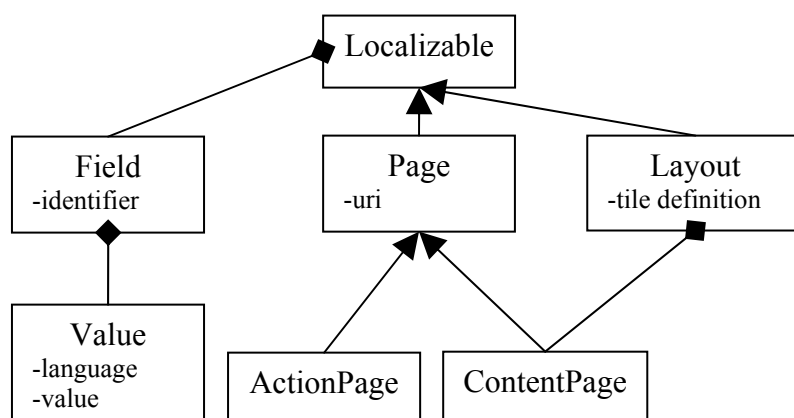
Отдельно выделяется особый вид поля, индексированное. Оно позволяет динамически повторять блоки, достаточно сложные для редактирования в визуальном редакторе.



Как мы отметили, плитки могут наследоваться, а значит, могут наследоваться и макеты, переопределяя значения полей с одинаковыми идентификаторами или добавляя новые поля.

В AtLeap существуют два понятия страниц: *статическая страница* (Content Page) и *динамическая страница* (Action Page).

Статическая страница основывается на макете, она полностью виртуальная, хранящаяся в БД. Создатель страницы может самостоятельно определить URI страницы. Статическая страница может переопределять значения полей своего макета. Создатель страницы может динамически определить роли для ограничения доступа к странице.



Динамическая страница это результат выполнения Struts действия (action). Это позволяет редактировать содержимое даже на динамически генерируемых (например, на основе данных из БД) страницах. Результатом может быть одна из нескольких страниц, поэтому макет, на котором основывается динамическая страница, определяется только во время выполнения. Динамическая страница может переопределять значения полей макета или/и добавлять новые поля. Создатель страницы может определить роли для ограничения доступа к странице в конфигурационном XML файле Struts.

AtLeap позволяет зачислять ресурсы (изображения и вообще файлы) в систему с сохранением их в БД. Создатель ресурса может определить роли для ограничения доступа к ресурсу.

AtLeap имеет возможности построения меню, независимого от структуры страниц. Система меню основывается на библиотеке Struts-Menu (<http://www.sf.net/projects/struts-menu>). Начальная структура меню определяется в конфигурационном XML файле. Элементы меню могут быть переопределены (или добавлены) на одном из макетов или странице. Создатель элемента меню может определить ссылку, выбрав из списка существующую страницу (или ресурс) и задать иконку, выбрав из списка существующее изображение.

AtLeap использует самый новый HTML редактор FCKEditor (<http://www.fckeditor.net>). Это первый чисто DHTML редактор совместимый с Internet Explorer 5.5+ и Gecko браузерами (Mozilla, FireFox, Netscape). Он имеет богатые возможности форматирования текста (таблицы, списки), буфер обмена, функция отмены, вставка текста из Word. AtLeap позволяет вставить ссылку, выбрав из списка, существующую страницу (или ресурс) или вставить изображение, выбрав из списка существующее изображение.

AtLeap обеспечивает полнотекстовый поиск по значениям полей страниц и макетов на нескольких языках (Английский, Испанский, Русский, Немецкий и т.д.) с учетом окончаний и стоп слов. Для поиска используется библиотека Lucene (<http://jakarta.apache.org/lucene>).

В связи с тем, что почти все данные хранятся в БД, с целью уменьшения количества запросов к БД, реализована эффективная система кэширования, основанная на библиотеке OSCache (<http://www.opensymphony.com/oscache/>).

Таким образом, полученная концепция СУС реализует, среди прочих требований (гибкость, платформонезависимость, расширяемость и т.п.), 3 главных требования к СУС:

- Простоту – обеспечивается наличием мощного WYSIWYG (What You See Is What You Get) редактора для создания и модификации содержимого с возможностью импорта текста из Microsoft Word.
- Многоязычность – AtLear обеспечивает работу с несколькими естественными языками одновременно, с полнотекстовым поиском.
- Индивидуализацию – AtLear никоим образом не ограничивает графический дизайн сайта. Заказчики не хотят видеть шаблонный дизайн, они хотят видеть уникальное индивидуальное решение.

Спроектированная и реализованная концепция позволяет составить конкуренцию существующим решениям на рынке СУС, например: самой распространенной Java open-source СУС OpenCMS (<http://www.opencms.org/>) или даже коммерческой Jahia (<http://www.jahia.org/>).

7 База

Возможностей многих сегодняшних open-source СУС бывает вполне достаточно для простых информационных сайтов, однако с ростом бизнеса компании растут и потребности в наличии более сложной бизнес логики в веб приложении. Часто фундамент, на которых построена используемая СУС, бывает неспособен к расширению и построению на его основе качественного, стабильного и безопасного веб приложения со сложной бизнес логикой.

AtLear имеет прочную базу, что обеспечивает ему гибкость и расширяемость для построения достаточно сложных веб приложений.

7.1 Хранение данных

AtLear использует библиотеку Hibernate (<http://www.hibernate.org/>), которая обеспечивает:

- Прозрачную способность создавать и поддерживать перманентные объекты.

- Отображение (mapping) объектной модели (включая наследование) в таблицы реляционной СУБД.
- Поддержка до 17 различных реляционных СУБД.
- Объектно-ориентированный язык запросов, независимый от СУБД.
- Поддержка транзакций.

При выборе персистентного слоя, помимо Hibernate, было проанализировано несколько решений: OJB (<http://db.apache.org/ojb/>), JPOX (<http://www.jpox.org/>), iBatis (<http://www.ibatis.com/>). Однако Hibernate более распространена, большее количество open-source продуктов имеет интеграцию с ней, на главное в основе разрабатываемого стандарта JSR 220: Enterprise JavaBeans™ 3.0, будет лежать большое количество идей Hibernate. Существующий же стандарт JSR 153: Enterprise JavaBeans™ 2.1 не обладает достаточной функциональностью (отсутствие гибкого языка запросов, отсутствие наследования и т.п.), поэтому его реализации даже не рассматривалась в качестве альтернативы.

С целью обеспечения большей гибкости при работе с данными используется библиотека Spring (<http://www.springframework.org/>), она обеспечивает:

- Управление конфигурацией основанное на JavaBeans.
- Абстрактный слой для управления транзакциями.
- Поддержку паттернов проектирования DAO (Data Access Object) и Business Delegate, для инкапсуляции части бизнес логики.

С использованием библиотеки DBUnit (<http://www.dbunit.org/>) реализуется возможность резервного копирования БД, в независимый от СУБД, XML формат.

7.2 Библиотека тегов

AtLeap использует библиотеку JSP тегов из Struts в дополнении к Apache реализации JSTL (<http://jakarta.apache.org/taglibs/>). Однако AtLeap имеет свою богатый набор тегов для построения пользовательского интерфейса, в частности теги для представления списочных данных в виде таблицы. Разработка позволяет:

- Возможность сортировки столбцов,
- Разделение на страницы длинных списков,
- Фильтрация данных следующих типов: строка, дата, число, множество.
- Представление гетерогенных данных в одной таблице (например: папки и элементы).

Разработанное решение обладает значительной гибкостью в сравнении со всеми известными решениями, в том числе с библиотекой Display (<http://displaytag.sf.net/>).

7.3 Оповещения и сборка

AtLeap в своем коде имеет множество log вставок на нескольких уровнях: ошибка, предупреждение, информация, отладка, что позволяет отследить использование системы и возможные неполадки. С этой целью используется библиотека log4j (<http://jakarta.apache.org/log4j/>). Она позволяет перенаправлять информационные сообщения: на консоль, в файл, удаленный log4j сервер, удаленный Unix syslog демон, Windows NT систему оповещений или послать по эл. почте.

AtLeap использует платформонезависимую Java библиотеку Ant (<http://ant.apache.org/>) для компиляции и сборки проекта. Используется XDoclet (<http://xdoclet.sourceforge.net>) для генерации кода, что позволяет реализовать атрибутно-ориентированное программирование на Java (Java 5 уже имеет встроенные возможности), таким образом, автоматизировав часть

рутинной работы по написанию шаблонного кода и конфигурационных файлов.

8 Будущее

СУС в AtLear сейчас обеспечивает изменения страниц (частей страниц) сайта, однако функциональность не достаточно для написания больших статей, публикаций. Для этих целей необходима система управления документами, которая бы обеспечивала групповую работу над документом, контроль версий, историю изменения, документопотоки и т.п. Для реализации этой функциональности планируется использовать библиотеку Slide (<http://jakarta.apache.org/slide/>), имплементация стандарта WebDAV.

С выходом стандарта EJB3.0 и его open-source реализации JBoss (<http://www.jboss.org/>) планируется переход на EJB3.0, который позволит распределять нагрузку на несколько серверов.

Для построения качественных веб приложений требуется богатая среда автоматизированного тестирования, ее реализация планируется на основе библиотек: JUnit, DBUnit, JUnitDoclet, StrutsTestCase, WebTest.

А так же планируется разработка дополнительных возможностей:

- Модули: статистика посещения страниц сайта, рассылки, форумы, голосования и т.п.;
- Полнотекстовый поиск по документам: RTF, Word, Excel, PowerPoint, PDF;
- Контроль правильности ссылок и т.д.

9 Заключение

Таким образом, каркас AtLear является достаточной мощной базой для быстрого построения на нем веб приложений при решении задач средней и малой сложности. AtLear позволяет:

- обеспечить независимость от платформы, операционной системы, СУБД;
- используя СУС, создавать индивидуальные, гибкие веб решения, сохраняя при этом простоту использования;
- обеспечить расширяемость веб приложения за счет использования мощной базы;
- бесплатно использовать AtLeap (в том числе для коммерческих целей), за счет лицензии Apache 2.0.

Еще до выхода первой бета версии AtLeap появились иностранные заказчики для реализации веб приложений на его основе.

10 Литература

1. Bruce Tate, Justin Gehtlatd. Better, Faster, Lighter Java. O'Reilly, 2004.
2. Joseph Walnes, Ara Abrahamian, Mike Cannon-Brookes, Patrick A. Lightbody. Java Open Source Programming: with XDoclet, JUnit, WebWork, Hibernate. Wiley, 2003.
3. Bob Baxley. What is a Web Application? January 20, 2003. (http://www.boxesandarrows.com/archives/what_is_a_web_application.php)
4. Заостровцев Н.В. Выбираем систему управления контентом для небольшого предприятия. 2004. (<http://business-site.ru/articles/cmsnp.pdf>)

A.N. Grebnev

This paper covers AtLeap web application designed by author and developed under author management. AtLeap is the open source Java framework to construct new web application based on it rapidly. Author explains the reasons

of main libraries choice, which are used during construction. The big emphasis was placed to main AtLeap functionality – Content Management System based on ideology designed by author.

А. Н. Гребнев AtLear - Java каркас с открытым исходным кодом для Web приложений. // Вестник ИжГТУ: период. науч. – теор. журн. – Ижевск: 2006. Изд-во ИжГТУ, 2006. – №1. – 116 с. – С.64-68

Гребнев Андрей Николаевич

Удмуртский Государственный Университет,

Кафедра Математического Обеспечения ЭВМ,

Ижевск, ул. Университетская, 1 (корп. 4)

426034, Россия

E-mail: ag@blandware.com

<http://fox.blandware.com>