

РАЗВЕРТЫВАНИЕ И ИНТЕГРАЦИЯ ИННОВАЦИОННЫХ УЧЕБНЫХ СРЕД: БОРДКАСТИНГ, ОБЛАЧНЫЕ ХОСТИНГИ И EDX

Аннотация

Статья содержит анализ основных тенденций распространения виртуальных учебных сред, описание технологических основ их внедрения, рекомендации по развертыванию инновационной среды (edX) на облачном хостинге, описание интеграции e-Learning-решений на примере среды edX и редактора бордкастов.

Ключевые слова: веб-технологии, развитие и интеграция инновационных виртуальных сред обучения, бордкастинг, полнодуплексное взаимодействие в вебе.

Для современного этапа развития образования в мире и в России характерно широкое внедрение различных систем управления обучением, которые также можно называть виртуальными учебными средами: LMS, Learning Management Environment.

Цель данной статьи – проанализировать предпосылки формирования этого этапа и на основе анализа предложить один из вариантов развертывания и интеграции E-Learning-решений, включающий инновационную учебную среду и средство расширения ее функциональности. Рассматриваемый этап является следующим после этапа обеспечения образовательного процесса электронными образовательными ресурсами. Автор данной статьи принимал участие в качестве инженера и программиста в разработке первых каталогов ЭОР-ов, затем в качестве разработчика ЭОР-ов нового поколения (на основе системы OMS), которые впоследствии сформировали Федеральный центр информационно-образовательных ресурсов (ФЦИОР). Отметим, что это была одна из первых попыток построения централизованной системы управления ресурсами с организацией локальных хранилищ на компьютерах пользователей. Для создания ресурсов был разработан специальный язык разметки RMML, напоминающий предложенный ранее Консорциумом W3C язык SMIL. Оба языка содержат средства для декларативного описания сцен и анимации, вставки аудиовидео-роликов и других подключаемых ресурсов. Наиболее продвинутые ЭОР-ы неизбежно включали также Flash как самый быстрый способ создания интерактивных учебных анимаций. И для создания ресурсов, и для проигрывания (OMS Player) использовалось Windows-зависимое ПО, что впоследствии создало непреодолимые трудности при внедрении Linux-решений в образовательных учреждениях РФ.

В настоящее время оба указанных языка и Flash/Flex представляются аутсайдерами на фоне взрывообразного распространения решений на основе HTML5.

Замещение Flash на HTML5 в вебе связано с тенденцией к отказу от проприетарных стандартов в пользу свободных кодеков/форматов, таких как VP8/webm, а также с быстрым распространением мобильных платформ, для которых Flash не лицензирован. В результате развития HTML5 и сопутствующих стандартов в любом современном браузере всю функциональность, присущую богатому приложению (Rich Application) можно реализовать на основе базовых языков и связанных с ними API (табл. 1).

В совокупности с распространением облачных решений это привело к тому, что разработка мультимедийных интерактивных ресурсов неотвратимо перемещается в веб. При-

© Государев И.Б., 2014

Табл. 1. Реализация функциональности ЭОР в HTML5

Функциональность	Реализация
Встраивание мультимедиа	Теги HTML5 (audio, video)
Пространственный звук и эффекты	Web Audio API
Анимация	Свойства CSS3
Геометрические преобразования (аффинные, проективные)	Преобразования CSS3
Векторная декларативная графика	Язык SVG
Динамическая программируемая графика	Canvas
Трехмерная графика, виртуальные миры	WebGL
Сторонние приложения	Различные API (например, OpenStreetMap)
Интерактивность	Javascript и библиотеки типа jQuery

чем, если на данном этапе в вебе располагаются прежде всего платформы (а ресурсы для них частично изготавливаются с помощью десктоп-приложений, таких как видеоредакторы), то с развитием онлайн-редакторов и облачных хранилищ перспективным представляется перемещение всей деятельности по разработке образовательных ресурсов любой сложности в веб. Однако развитие возможностей учебных сред, как правило, не успевает за развитием отдельных инструментов. Поэтому для решения всего спектра задач образования необходима их интеграция с другими инструментами, расширяющими их функциональность.

Характерной чертой конца первого десятилетия XXI века является активное использование в образовании разрозненных технологий Web 2.0, располагающихся на проприетарных серверах, например, блогов, вики, видеохостингов, Google-сайтов и т. д. Это более высокий уровень по сравнению с использованием отдельных ресурсов, загружаемых из репозитивов и каталогов типа fcior.edu.ru. Но во втором десятилетии прослеживается новая тенденция – это тенденция интеграции различных решений на какой-либо учебной платформе, или в виртуальной образовательной среде. Так, на самое начало второго десятилетия пришелся пик распространения среды MOODLE – модульной объектно-ориентированной динамической среды обучения, относящейся к классу LMS – систем управления обучением. В связи с некоммерческим характером этой среды она приобрела огромную популярность в России, несмотря на довольно нерациональное внутреннее устройство и относительно низкую производительность, связанную с выбором интерпретируемого языка PHP для реализации серверной бизнес-логики. Среда (автор анализировал текущую поддерживаемую версию 2.6.x) программно состоит из огромного количества PHP-файлов, решающих отдельные задачи, обращение к которым происходит с помощью гиперссылок в интерфейсе пользователя вида <http://servername/moodle/course/category.php?id=1> или [/moodle/admin/user.php](http://servername/moodle/admin/user.php) и ее установка представляет собой распаковку архива с этими файлами с последующим обращением к конфигурационному сценарию через веб. Однако преимуществом является возможность установки MOODLE на очень многие виртуальные хостинги, в том числе бесплатные, что обеспечивается сравнительно лояльными системными требованиями (PHP 5.3.3, MariaDB 5.3.5 или MySQL 5.1.33 или Postgres 8.3 или MSSQL 2005 или Oracle 10.2 для 2.6.x). Также MOODLE может быть установлена в локальную сеть и быть частью инфраструктуры, включающей, к примеру, репозиторий Alfresco, аутентификацию LDAP, систему портфолио Mahara и так далее. Как и многие современные LMS, MOODLE поддерживает такие стандарты, как LOM или SCORM, позволяя импортировать учебные модули. Благодаря невысоким системным требованиям и открытости кода MOODLE победила многих конкурентов (таких как WebCT). MOODLE установлена на серверах множества вузов и других образовательных учреждений. Например, в Санкт-Петербурге это сервер дистанционного обучения РЦОКОиИТ <http://do.rcokoit.ru>, сервер портала ЕГЭ <http://distant.ege.spb.ru>.

Но его пользовательский интерфейс является устаревшим. Он не поддерживает архитектуру One Page Application (одностраничного приложения), основанную на асинхронных запросах, и не содержит удобных средств нативной (изначальной) поддержки HTML5-мультимедиа. Функциональность системы может быть расширена с помощью множества надстроек, но базовая конфигурация, доступная большинству пользователей, довольно бедна. Она позволяет создавать весьма ограниченный набор простых ресурсов (ссылки, HTML-страницы), импортировать пакеты с ресурсами и создавать такие компоненты курсов, как тесты с ограниченным набором видов псевдотестовых заданий. Эти и другие обстоятельства указывают на предстоящий неизбежный спад популярности данной учебной среды.

Еще одной тенденцией является возрастание требований к системным ресурсам. Как уже было отмечено выше, перенос не только процесса просмотра и обучения, но и процесса разработки в веб, обуславливает необходимость наличия достаточно специального ПО на стороне сервера. Для функционирования такой системы требуется предоставление не готовой конфигурации для работы с PHP, Python, Ruby-on-Rails и т. д. в рамках виртуальных серверов виртуального хостинга, а как минимум виртуального выделенного сервера (VPS). В основе VPS находится виртуализация на уровне операционной системы. В настоящее время всё большее распространение приобретают *облачные хостинги* (Selectel, Cloud4Y, DigitalOcean и др.) [1], предоставляющие платформу как регулируемую услугу. Виртуализация осуществляется на аппаратном уровне, используется несколько физических серверов, и, по мере изменения нагрузки на ресурсы, облачный виртуальный сервер мигрирует на менее нагруженный гипервизор, а пользователь может гибко масштабировать ресурсы — динамически наращивать доступный объем оперативной памяти, дискового пространства, при этом выбирая удобную форму оплаты (в минуту, в час, в месяц, за реально использованные вычислительные ресурсы).

В случае использования виртуального выделенного сервера (в том числе облачного) пользователю предоставляется копия системы с правами доступа уровня root (UID: 0) для Unix или Administrator для Windows, что позволяет производить компиляцию, а также установку собственного программного обеспечения с изменённой конфигурацией. Это резко сужает круг лиц, которым доступна установка подобного ПО, поскольку для решения задач даже первоначальной установки, развертывания и настройки требуются навыки системного администратора. И это еще одна важная тенденция: учебные среды требуют новых технических компетенций и становятся менее доступны. В связи с этим становится актуальным исследование процесса развертывания современных учебных сред.

Таким образом, инновационная учебная среда включает компоненту, ответственную за предоставление аудитории интерфейса к курсам (LMS), и компоненту, ответственную за создание контента (CMS). Отметим, что в MOODLE эти две части, по сути, слиты вместе, переключение осуществляется в одной среде от роли к роли кнопкой «Режим редактирования». Примером платформы, в которой эти две компоненты разделены, может служить платформа edX, являющаяся разработкой Массачусетского технологического института и первоначально существовавшая в виде сервера MITx (mitx.org). 1 июня 2013 года исходные коды сервера были предоставлены в публичный доступ (<http://code.edx.org>, <https://github.com/edx>), после чего стали доступны стандартные процедуры загрузки исходных кодов из репозитория github. Платформа edX является типичной средой разработки и размещения курсов глобальной открытой модели (MOOC, Massive Online Open Courses). Эта модель, среди других моделей электронного обучения [2] в настоящее время активно внедряется на межвузовском уровне, примером чего являются проекты Coursera, Udacity, edX (и с недавнего времени российский проект lektorium.tv). Для MOOC в существующих формах реализации характерно использование видеолекций с личным участием преподавателя, вокруг которых выстраиваются все остальные компоненты курса (тестовые вопросы, задания для переключе-

стной оценки (peer assessment), консультационные форумы), в качестве центрального компонента. Этим обусловлена повышенная нагрузка на аппаратные возможности сервера, на котором функционирует среда – как правило, она не просто является процессом, запускаемым на хосте, а требует для своей работы выделенного компьютера.

Итак, с точки зрения развертывания коренное отличие edX от MOODLE состоит в том, что для установки MOODLE на виртуальный хостинг (просто в папку на виртуальном сервере) требуется распаковать загруженный архив и запустить точку входа (конфигурации) системы из веб-интерфейса, а для развертывания edX требуется:

- загрузить и установить требуемые сторонние компоненты,
- загрузить и настроить коды самой системы,
- создать суперпользователя (админа) для управления системой через интерфейс админа,
- настроить систему для конкретного выделенного виртуального сервера, и только затем
- обратиться через веб-интерфейс к LMS/CMS.

Система edX состоит из подсистем LMS (основной продукт) и CMS “edX Studio” (создание курсов). Обе подсистемы написаны на языке Python с использованием фреймворка Django. Первая использует в качестве СУБД mysql, вторая – mongo. Другие компоненты решают различные внутренние задачи (форумы, написанные с помощью языка ruby, ORA – open response assessor – компонент для оценки эссе, реализующий некоторые идеи машинного обучения). Для обработки HTTP-запросов используются веб-сервера nginx и gunicorn. Подсистемы существуют в виртуальных средах Python (virtualenv).

Для локализации используется проект Transifex (<https://www.transifex.com/projects/p/edx-platform/>) – из его репозитория копируются файлы с переводами сообщений, элементов интерфейса и так далее.

Возможна установка двух реализаций edX:

- developer stack – для исследований и разработки (<https://github.com/edx/configuration/wiki/edX-Developer-Stack>),
- production stack – для реальной работы (<https://github.com/edx/configuration/wiki/edX-Ubuntu-12.04-Installation>).

Developer stack устанавливается на локальный компьютер в виде виртуальной машины с помощью Vagrant и VirtualBox.

Установка production stack кратко описана в виде алгоритма выше.

Автор данной статьи выбрал образ Ubuntu 12.04.4 LTS x86_64 с 2-мя гигабайтами оперативной памяти и 40 гигабайтами дискового пространства на облачном хостинге Digital Ocean. Виртуальный выделенный сервер (VPS) в терминологии DigitalOcean называется дроплетом (droplet). Весьма удобной возможностью является создание промежуточных образов дроплета в процессе постепенной настройки, позволяющих осуществлять откат к предыдущему стабильному состоянию (рис. 1).

На чистой системе начнем с обновления репозитория, установки Midnight Commander и перезагрузки (соединившись с дроплетом через SSH или консоль на сайте):

```
sudo apt-get update -y
sudo apt-get upgrade -y
sudo apt-get install mc
reboot
```

Для удобства все команды, реализующие перечисленные выше шаги установки, сосредоточены в одном пакетном файле для оболочки bash, располагающемся в репозитории проекта. Для его загрузки и выполнения следует использовать команду bash:

```
wget https://raw2.github.com/edx/configuration/master/util/install/vagrant.sh -O - | bash
```

В ходе установки устанавливается в числе прочего ansible - платформа для конфигурирования и управления компьютерами, использующая Python, JSON и YAML. С помощью

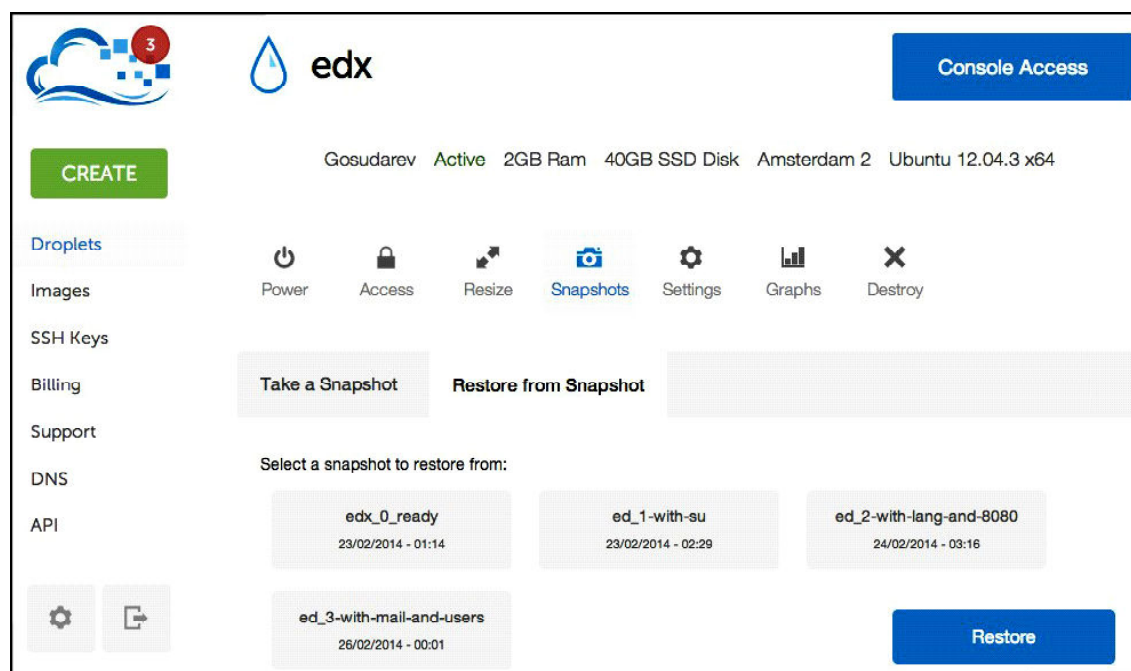


Рис. 1. Созданные промежуточные образы

формата YAML оформляются playbooks – подобие сценариев установки. В нашем случае они хранятся в `/var/tmp/configuration/playbooks`.

Последняя из команд установки – это как раз запуск основного такого сценария:

```
sudo ansible-playbook -c local ./edx_sandbox.yml -i "localhost,"
```

Надо помнить, что когда мы устанавливаем edX на Linux-сервер, мы уже имеем одну платформу – это Linux с ее пакетами, зависимостями и прочим. edX – это платформа поверх платформы. Идеология построения системы такова, что нежелательно вносить правки в конфигурационные файлы. Изменения должны вноситься через API самой платформы – то есть через более высокий уровень.

Один из этих уровней – ansible, другой – Django. Это соответствует идеологии MVC, где все делается через одну точку входа, или работу с объектами, когда мы не устанавливаем вручную переменную, а вызываем для нее сеттер.

Пример. Чтобы редактировать курсы через веб-интерфейс, используется Studio – для загрузки веб-интерфейса нужно перейти по адресу `myedxserver:18010` – таковы настройки по умолчанию. Порт обслуживается сервером nginx. Однако в корпоративных сетях необычные порты часто бывают закрыты. Как изменить порт доступа к CMS?

Если проанализировать содержимое каталога `/edx/app/nginx/sites-available`, то мы увидим файлы `cms`, `lms` и так далее. В файле `cms` находится в числе прочего строка:

```
# CMS configuration file for nginx, templated by ansible
listen 18010 ;
```

Логично, если мы хотим поменять порт, написать здесь вместо 18010 свое число (допустим, 8080). Далее следует перезапустить сервер nginx.

```
sudo service nginx stop
sudo service nginx start
```

Однако официальный форум предлагает сделать смену через запуск ansible с «перекомпилированием» системы:


```
ansible-playbook -c local -limit "localhost:127.0.0.1" /var/tmp/configuration/playbooks/
edx_sandbox.yml -i "localhost," -e 'cms_nginx_port=8080'
```

Многие действия можно совершать через веб-интерфейс после создания суперпользователя. Для этого используется файл `manage.py` (консоль Django для администрирования). Чтобы вывести список всех команд, нужно выполнить переход в папку с виртуальной средой:

```
cd /edx/app/edxapp/venvs/edxapp/bin/
```

и запустить консоль с рядом специальных параметров (см. рис. 3):

```
sudo -u www-data ./python /edx/app/edxapp/edx-platform/manage.py lms --settings=aws --service-variant lms help
```

Отметим, что `/edx/app/edxapp/edx-platform/` – это папка с ядром платформы

После создания суперпользователя

```
sudo -u www-data ./python /edx/app/edxapp/edx-platform/manage.py lms --settings=aws --service-variant lms createsuperuser --username creator
```

можно обратиться к его «кабинету» по адресу `.../admin` (рис. 2).

В некоторых случаях проще внести нужные изменения в ряд конфигурационных файлов, таких как файлы `common.py` в папке платформы в папках `lms/envs` и `cms/envs`, и некоторые другие. Рассмотрим следующий пример: когда пользователь осуществляет саморегистрацию на сайте LMS, на указанный им электронный адрес отсылается автоматически созданное письмо от адреса `registration@example.com` и предлагается перейти на сервер `localhost`. Судя по всему, на данном этапе edX не содержит удобного способа настроить реальные значения, и для этого приходится вносить правки в конфигурационный файл `/edx/app/edxapp/lms.env.json`

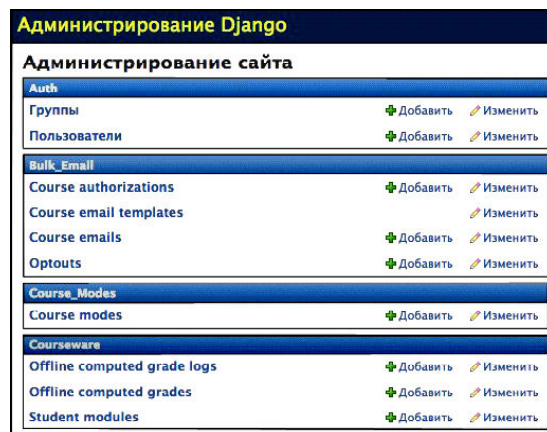


Рис. 2. Веб-интерфейс суперпользователя

```
root@edx:/#
root@edx:/edx/app/edxapp/venvs/edxapp/bin# sudo -u www-data ./python /edx/app/edxapp/edx-platform/manage.py lms --settings=aws --service-variant lms help
2014-02-27 21:01:47.794 INFO 5504 [dd.dogapil dog_stats_api.py:66 - Initializing dog api to use statsd: localhost, 8125
Usage: manage.py subcommand [options] [args]

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level: 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS
                        The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback
                        Print traceback on exception
  --version
                        show program's version number and exit
  -h, --help
                        show this help message and exit

Type 'manage.py help <subcommand>' for help on a specific subcommand.

Available subcommands:

[auth]
  changepassword
  createsuperuser

[certificates]
  cert_whitelist
  fix_ungraded_certs
  gen_cert_report
  regenerate_user
  ungenerated_certs
```

Рис. 3. Вывод доступных команд консоли Django

После внесения изменений необходимо перезапускать LMS и Studio (см. рис. 4):

```
sudo /edx/bin/supervisorctl -c /edx/etc/supervisord.conf restart edxapp:lms
sudo /edx/bin/supervisorctl -c /edx/etc/supervisord.conf restart edxapp:cms
```

Платформа edX, конечно, поддерживает большинство современных e-Learning-инструментов. Однако некоторые задачи все же пока приходится решать за пределами учебных сред, находя способы интеграции с ними. Интеграция может осуществляться на основе различных решений и их комбинаций, например:

- SCORM – стандарт, определяющий взаимодействие между LMS и загружаемым модулем через API, реализованное на языке Javascript (как правило, модули загружаются на сервер с LMS и вставляются в HTML-код с помощью тега `iframe`);
- REST – архитектура взаимодействия между веб-приложениями, использующая логическое URL API и форматы взаимодействия – XML, JSON;
- взаимодействие через промежуточный слой (например, БД);
- взаимодействие через специальные API (например, с помощью обмена кроссдоменными запросами на основе XMLHttpRequest + HTTP access control (CORS), – Dropbox API или WebSockets).

В самом простом случае интеграция реализуется на логико-семантическом / содержательном уровне путем вставки гиперссылок. Так, в курсы Moodle часто встраиваются гиперссылки на материалы, опубликованные в Dropbox или других облачных хранилищах. Взаимодействие может предполагать внесение изменений в связанный ресурс, то есть интерактивное, двустороннее (полнодуплексное) взаимодействие. На уровне пользователя это может быть реализовано с помощью Google-документов, предоставленных в режиме редактирования в общий доступ. Для обозначения такого типа взаимодействия можно использовать термин «бордкастинг».

Бордкастинг [3] (англ. board – доска и англ. broadcasting – передача, вещание) позволяет передавать для широкой аудитории содержимое электронного документа в процессе его изменения. Отличается от скринкастинга интерактивностью, ориентированностью на он-

The screenshot shows a web browser window with the URL `smartedu.mobi:8080/course`. The page title is "My Courses" and the user is logged in as "gossoudarev". A green button labeled "+ New Course" is in the top right. The main content area is titled "Welcome, gossoudarev!" and lists "Here are all of the courses you currently have access to in Studio:". Below this is a "Create a New Course" section with two input fields: "Course Name *" (containing "Основы веб-разработки") and "Organization *" (containing "HerzenUniv"). A note below the organization field states: "The name of the organization sponsoring the course. Note: This is part of your course URL, so no spaces or". To the right of the form is a "Need help?" section with the text: "If you are new to Studio and having trouble getting started, there are a few things that may be of help:". It includes two links: "Get started by reading Studio's Documentation" and "Request help with Studio".

Рис. 4. Окно создания курсов в Studio с измененным портом

лайнное взаимодействие с аудиторией. Бордкастинг может быть реализован на основе существующих облачных технологий на примере Google Drive: при предоставлении документа или электронной таблицы (презентации, рисунка и т. д.) в публичный доступ с правом просмотра или редактирования все подключенные участники могут наблюдать «в прямом эфире» изменения в документе, вносимые ведущим. Наилучших результатов позволяет достигать сочетание бордкастов, скринкастов и облачных лабораторных работ, образующих единый облачный УМК – мобильную информационную образовательную среду [2].

В качестве действующего примера интеграции учебной среды и веб-приложения, расширяющего ее функциональность, автором данной статьи разработано, зарегистрировано в Роспатенте и введено в эксплуатацию приложение «Кодактор», позволяющее осуществлять вещание борда (доски), состоящего из кода на языках HTML, CSS и Javascript, а также результата отображения этого кода в браузере. Приложение интегрирует Dropbox и валидатор W3C, а также библиотеку бордов (заранее созданных досок) по работе с веб-фреймворками, мультимедиа и анимацией, геоинформационными сервисами и т. д. Приложение, как и описанная выше платформа edX, развернуто на том же облачном виртуальном сервере, что обеспечивает возможность взаимодействия с основной средой сразу на нескольких уровнях (общая база данных, взаимодействие через REST API).

Основной алгоритм работы с бордкастингом в данном приложении состоит в том, что ведущий (преподаватель) создает исходный борд и передает его ключ (URL-адрес) вида <http://kodaktor.info/6c1da88> аудитории (студентам). После этого любые производимые им действия одновременно отображаются в браузерах студентов. Обычно борд содержит также задания для выполнения, на основе которых студенты создают свои борды и передают их адреса преподавателю для проверки (путем публикации в блогах или социальных сетях). На основе ряда бордкастов может быть организовано обучение решению задач возрастающей сложности в формате компьютерного практикума, активной лекции или дистанционного семинара (вебинара).

Работающее приложение доступно по адресам: h1.dist-learn.spb.ru, ict.herzen.spb.ru/h1 или kodaktor.info. Использование «Кодактора» в связке с учебной средой позволяет решать две задачи: интеграция в среду интерактивных учебных материалов, связанных с языками разметки и сценариев (для курсов предметных областей «Интернет-технологии», «Веб-программирование» и т. п.) и подготовка кода на веб-языках для самой среды (то есть онлайн-редактирование клиентских расширений среды). В последнем случае с помощью онлайн-редактора, установленного в одном облаке со средой, можно создавать и редактировать плагины для среды. В связи с этим актуальна задача создания библиотеки распространенных решений, образцов кода и готовых для встраивания фрагментов.

Некоторые борды с предустановленными ключами содержат заранее отобранные образцы кода, применяемые для конструирования приложений. Например, борд 5 (<http://ict.herzen.spb.ru/h1/?v=5> или <http://kodaktor.info/5>) содержит код и встроенную инструкцию-скринкаст по внедрению видео в форматах HTML5 (о чем шла речь в начале статьи) (рис. 5).

Техническая реализация бордкастинга зависит от платформы, на которой предполагается развертывать приложение. Технически бордкастинг – это просто чат, поэтому программно он может быть реализован двумя основными способами:

- 1) участники (одноранговые клиенты или аудитория с вещателем) обмениваются данными через общий промежуточный слой, в роли которого выступает база данных на сервере и обслуживающий ее сценарий; сервер только отвечает на запросы клиентов;
- 2) участники представляют собой равноправные хосты, вступающие в полнодуплексное взаимодействие на базе соответствующего протокола типа WebSocket; сервер инициирует обновления на клиентах, выступая одним из них.

Чтобы обеспечить настоящее полнодуплексное взаимодействие, нужна платформа наподобие node.js с такими библиотеками как sockjs или Socket.IO. Поскольку node.js пред-

ставляет собой серверную платформу, кардинально отличающуюся по концепции и реализации от таких традиционных веб-серверов, как Apache, единственным реалистичным способом ее установки, изучения и использования является облачный хостинг. Это же касается и решений на базе Java и других необычных серверных решений.

В приложении «Кодактор» взаимодействие организовано по первой схеме, то есть происходит имитация полнодуплексного взаимодействия с помощью асинхронных AJAX-запросов. В основе приложения – библиотека jQuery, позволяющая построить код программы в виде последовательности вызовов безымянных callback-функций вида

```
$.get("codes/index.php", function(r){
    //загрузка шаблонов кодов для борда (с) И.Б.Государев
    $("#codes").append(r).val("codes/HTML5.html").change(function(){
        $.get($(this).val(), function(data) {
            $("#onboard").attr("checked", "");
            clearInterval(bTimer1); //после загрузки образца кода нужно перестать
                                //считывать борд, если он считывался
            editor.setValue(data); //загрузить в редактор считанный код
            editor.setCursor(1,1); // установить позицию курсора
            play(); //отобразить код в дисплейной панели
        });
    });
});
```

«Кодактор» интегрирован с облачным хранилищем DropBox и позволяет считывать данные из авторизованного аккаунта и записывать в него. Построение по модульно-плагиноному принципу позволяет достаточно легко расширять функциональность приложения. В частности, возможно создание плагина на основе стороннего API для взаимодействия с плат-

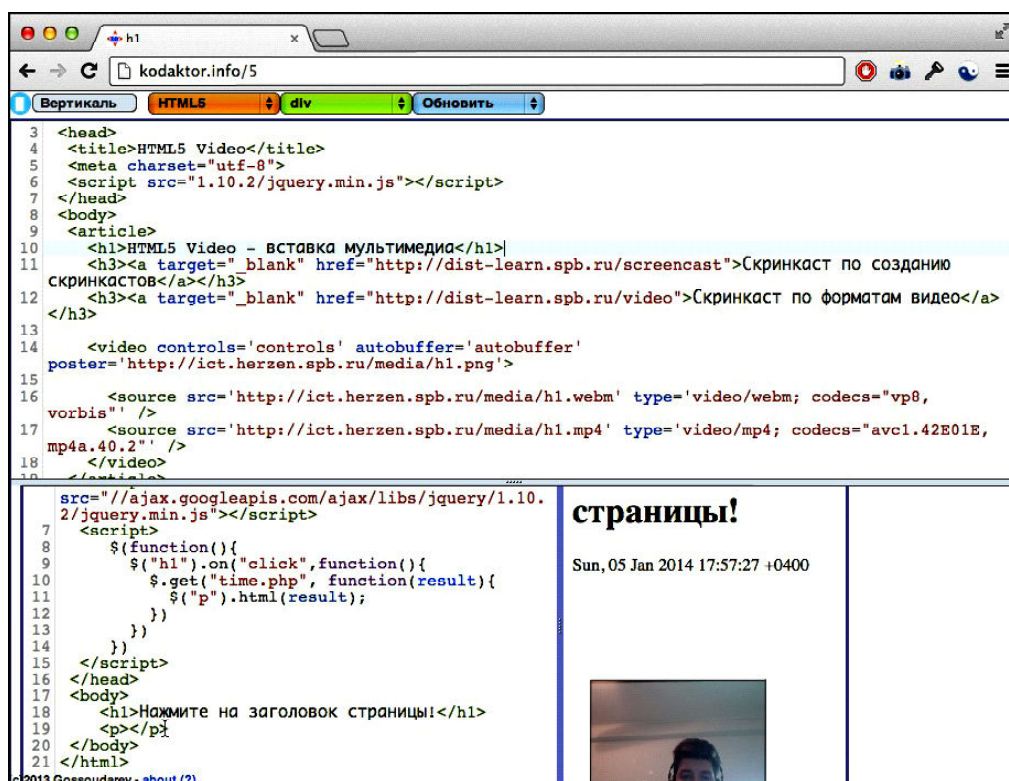


Рис. 5. Пример бордкастинга в приложении «Кодактор»

формой edX (не говоря о более простых, таких как MOODLE API). Поскольку приложение достаточно небольшое по объему кода, пока не полностью обосновано использование «серьезных» MVC-фреймворков для серверной части, таких как Zend или Yii. Поэтому на данном этапе существования серверная часть выполнена в функциональном стиле на основе PHP/MySQL или, по выбору, sqlite.

Развитие различных e-Learning-решений происходит крайне динамично. Возникают и исчезают стандарты, протоколы, библиотеки и паттерны проектирования. Различные решения интегрируются и становятся основами для создания новых платформ. В данной статье предложен вариант развертывания виртуальной учебной среды и ее интеграции с расширяющим ее функциональность приложением на основе облачного хостинга. В период тестирования проекта (<http://smartedu.mobi>) материалы курсов и приложение «Кодактор» (<http://kodaktor.info>) открыты для бесплатного использования и совместной разработки.

Литература

1. Amazon EC2 Beta [Электронный ресурс] URL: http://aws.typepad.com/aws/2006/08/amazon_ec2_beta.html (дата обращения: 01.05.13).
2. Гусударев И.Б. Электронное обучение : тенденции развития моделей и опыт применения // Известия Российского государственного педагогического университета имени А. И. Герцена, 2013. № 162. С. 162–166. [Электронный ресурс]. URL: http://lib.herzen.spb.ru/text/gosudarev_162_162_166.pdf (дата обращения: 01.02.14).
3. Гусударев И.Б. Электронное обучение веб-разработке на основе бордкастинга и платформы edX // Современная педагогика: теория, методика, практика. Сборник материалов международной научной конференции. Россия, г. Москва, 27–28 февраля 2014 г. [Электронный ресурс] / под ред. Д.Ю. Ануфриевой. Электрон. текст. дан. (1 файл 1,3 Мб). Киров: МЦНИП, 2014. 150 с. 1 электрон. опт. диск (CD-ROM). (дата обращения: 28.02.14).

DEPLOYING AND INTEGRATING INNOVATIVE LEARNING ENVIRONMENTS: BOARDCASTING, CLOUD HOSTING AND EDX

Abstract

The article provides analysis of the main trends in the virtual learning environments development and distribution; description of the technologies behind the implementation of these environments; recommendations on the deploying of the innovative learning environments (e.g. edX) on cloud hosting (virtual private servers); description of the e-Learning solutions integration, e.g. edX and the boardcasting online editor.

Keywords: web-technology, innovative learning environment development, boardcasting, full-duplex web interaction.

*Гусударев Илья Борисович,
кандидат педагогических наук,
доцент, доцент кафедры
информационных и
коммуникационных технологий
РГПУ им. А.И.Герцена,
gossoudarev@herzen.spb.ru*



Наши авторы, 2014.
Our authors, 2014.

