



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____

КАФЕДРА _____

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

*Разработка динамических пользовательских
интерфейсов для распределенных систем
инженерного анализа*

Студент РК6-71
(Группа)

(Подпись, дата)

С.А. Неклюдов
(И.О.Фамилия)

Руководитель курсовой работы (проекта)

(Подпись, дата)

А.П. Соколов
(И.О.Фамилия)

Консультант

(Подпись, дата)

А.Ю. Першин
(И.О.Фамилия)

2018 г.

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой _____
(Индекс)
_____ А.П. Карпенко
(И.О.Фамилия)
« ____ » _____ 20__ г.

ЗАДАНИЕ
на выполнение курсовой работы

по дисциплине *«Методы оптимизации»*

Студент группы: *РК6-71*

Неклюдов Семен Александрович
(Фамилия, имя, отчество)

Тема курсовой работы: *Разработка динамических пользовательских интерфейсов для распределенных систем инженерного анализа*

Направленность КР :учебная

Источник тематики : *МГТУ им. Н.Э. Баумана*

График выполнения КР: 25% к ____ нед., 50% к ____ нед., 75% к ____ нед., 100% к ____ нед.

Техническое задание:

1. Провести аналитический обзор литературы по теме: «методы автоматической генерации графических пользовательских интерфейсов»
2. Ознакомление с форматом aINI
3. Доработка WEB-ориентированных GUI, основанных на формате aINI, в части поддержки специальных типов параметров;
4. доработка и разработка новых типов динамических WEB-ориентированных GUI, формируемых на основе DDL описаний реляционных моделей БД и файлов в формате aINI (древовидный, сгруппированный, сгруппированный с подытогами)

Оформление курсовой работы:

Расчетно-пояснительная записка на _____ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « 30 » сентября 2018 г.

Руководитель курсовой работы

Студент

(Подпись, дата)

(Подпись, дата)

А.П. Соколов
(И.О.Фамилия)

С.А. Неклюдов
(И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

АННОТАЦИЯ

Работа посвящена разработке динамических графических пользовательских интерфейсов в системах инженерного анализа. Разработка ПО инженерного анализа подразумевает создание модулей трех типов:

1. модули подготовки данных
2. модули обработки данных(вычисления)
3. модули представления данных

Наиболее рутинным процессом является разработка модулей подготовки данных. Системы инженерного анализа могут иметь большое количество разнородных входных данных различных типов. Возможность автоматизировать процесс построения GUI, в частности экранных форм, может существенно снизить трудозатраты по вводу входных данных.

Согласно исследованиям [3] время разработки модулей подготовки данных при помощи генерации динамических GUI можно сократить с 65% всего времени разработки до 5%, в связи с чем, можно повысить трудозатраты на более наукоемкие модули обработки данных.

Тип работы: курсовая работа

Тема работы (проект темы): «Разработка динамических пользовательских интерфейсов для распределенных систем инженерного анализа».

Объект исследований: *“методы генерации GUI ”*.

СОКРАЩЕНИЯ

PBC GCD — Распределенная вычислительная система GCD.
Comwps — веб клиент
GUI — графический пользовательский интерфейс
action item – тип действия в системе GCD. Например – тестовый генератор(wps_gui_ini_builde)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	7
1. ПОСТАНОВКА ЗАДАЧИ.....	7
1.1. Концептуальная постановка задачи.....	7
1.2. Математическая постановка задачи.....	7
2. ВЫЧИСЛИТЕЛЬНЫЙ МЕТОД.....	8
3. АРХИТЕКТУРА ПРОГРАММНОЙ РЕАЛИЗАЦИИ.....	8
4. ТЕСТИРОВАНИЕ И ОТЛАДКА.....	9
5. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ.....	9
6. АНАЛИЗ РЕЗУЛЬТАТОВ.....	9
ЗАКЛЮЧЕНИЕ.....	10
СПИСОК ЛИТЕРАТУРЫ.....	10

ВВЕДЕНИЕ

Изучением методов построения динамических пользовательских интерфейсов занимались многие исследователи и разработчики программного обеспечения.

По оценкам специалистов на пользовательский интерфейс уходит не менее половины времени, требуемого на разработку программного средства [4].

Для исследователей и разработчиков систем инженерного анализа методы построения GUI имеют важное практическое значение: они позволяют снизить время разработки программного обеспечения.

В системах инженерного анализа, не использующих методы динамического построения графического интерфейса пользователя время, затраченное на разработку модулей подготовки данных значительно выше, чем в системах, использующих такие методы, что наглядно показано в статье [3].

Были выявлены основные тенденции развития пользовательских интерфейсов, такие как событийная архитектура[11] , использование MVC фреймворков и JavaScript технологий.

На основе проведенного анализа литературы были изучены несколько подходов к автоматическому построению динамических пользовательских интерфейсов[6-10], в том числе подход, используемый в распределенной системе инженерного анализа GCD[3], изучены требования, предъявляемые к программному инструментарию PBC GCD:

1. Программный инструментарий должен включать в свой состав GUI-генератор, обеспечивающий автоматическое построение GUI на основе заранее определенных входных параметров
2. Для составления списка входных параметров должен быть разработан специализированный текстовый формат данных.
3. Формат файла должен поддерживать хранение скалярных типов данных.
4. Должна иметься возможность расширения списка поддерживаемых типов.

5. Программа-генератор GUI должна зависеть не от конкретных метаданных, а лишь от грамматики специализированного текстового формата, определяющей произвольные метаданные.
6. Подготовка списка входных параметров должна быть доступной для неподготовленного специалиста, не владеющего навыками программирования, в том числе для специалистов, представляющих заказчика разрабатываемой прикладной программы.
7. При изменении исходного списка входных параметров соответствующий GUI должен быть перестроен автоматически.
8. Процесс построения GUI должен осуществляться во время выполнения разрабатываемой прикладной программы, использующей программный инструментарий.
9. Программный инструментарий должен обеспечивать возможность автоматического определения параметров различных типов.
10. Специализированный текстовый формат данных, обеспечивающий возможность определения метаданных, должен предоставлять возможность группировки входных параметров по их назначению.
11. Должна быть обеспечена возможность редактирования списка входных параметров независимо от программы-генератора GUI.
12. Программный инструментарий должен иметь возможность встраивания в web-ориентированные приложения, приложения для мобильных платформ и приложения для операционных систем семейств Windows, Linux, macOS. Для поддержки соответствующих платформ должен быть разработан свой GUI-генератор.

Основной задачей проведения аналитического обзора литературы было ознакомление с методами формирования графических пользовательских интерфейсов, в частности, подхода, используемого в PBC GCD.

1. ПОСТАНОВКА ЗАДАЧИ

1.1. Концептуальная постановка задачи

Объект исследования: методы автоматического формирования GUI

Целью разработки является: доработка подсистемы ввода — вывода PDC GCD,

1. Доработка WEB- ориентированных GUI, основанных на формате Aini в части поддержки специальных типов параметров.
2. доработка и разработка новых типов динамических WEB-ориентированных GUI, формируемых на основе DDL описаний реляционных моделей БД и файлов в формате aINI (древовидный, сгруппированный, сгруппированный с подытогами)

2. АРХИТЕКТУРА ПРОГРАММНОЙ РЕАЛИЗАЦИИ

Стек технологий:

1. Frontend: HTML5+CSS+JavaScript(JQueru)
2. Backend: Python3, Django

Разработка ведется на базе проекта comwps системы PBC GCD, а точнее, в рамках модулей `wps_gui_ini_builder` и `iniparser`, представляющих собой обработчик ajax-запросов и парсер файла входных параметров в формате aINI, а также была выполнена доработка Django шаблонов для отображения извлеченной информации.

Фреймворк Django навязывает некоторую структуру для архитектуры приложения. Выделяют 3 главные составляющие Django: модели — сущности, ассоциирующиеся с данными, реагирующие на команды представлений; представления - интерпретируют действия пользователя; шаблоны отвечают за отображения данных для пользователя.

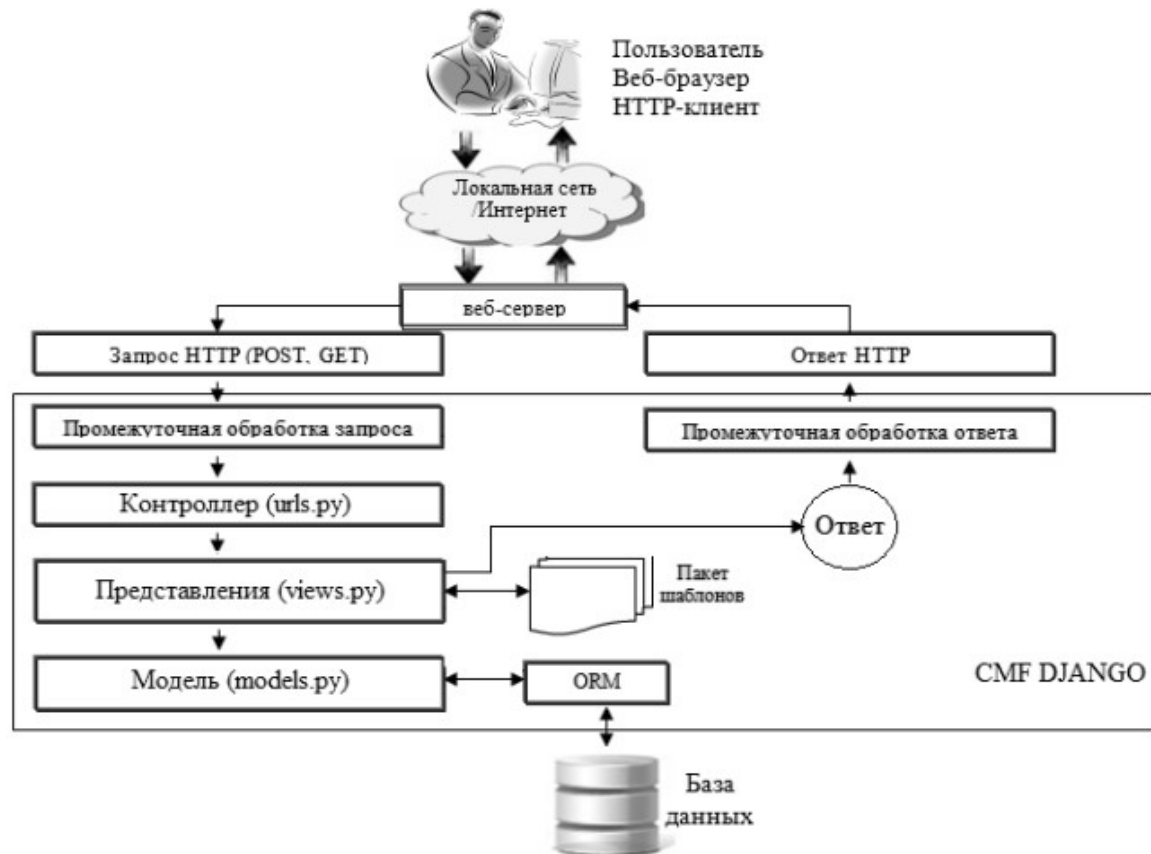


Рис. 1 . Структура приложения Django

Для понимания роли разработки в системе следует представить общую архитектуру подсистемы web — клиента comwpc и последовательность выполнения ajax запроса на выполнение action item wpc_gui_ini_builder:

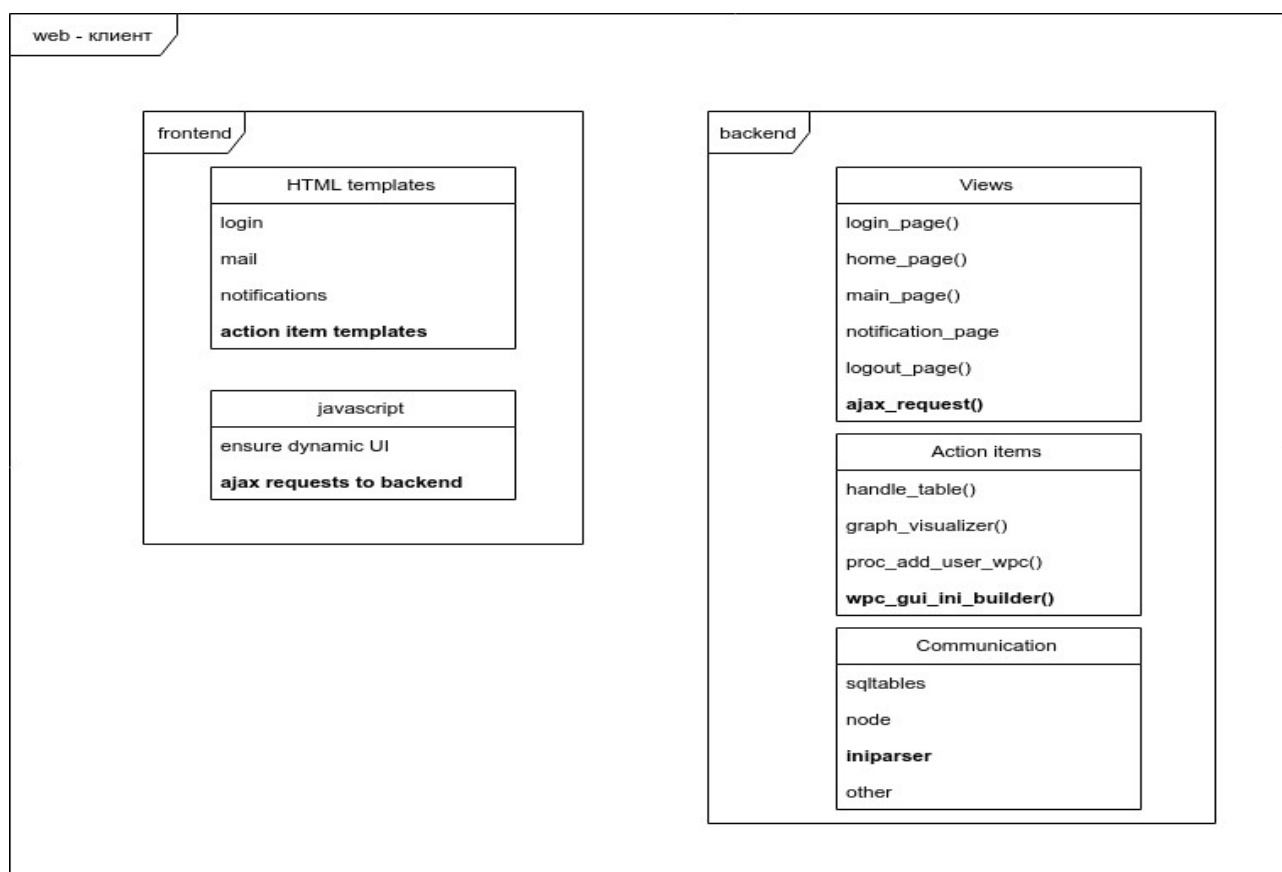


Рис. 2 . Архитектура web — клиента.

На серверной части система состоит из нескольких представлений, реализующих основной функционал web — клиента (авторизация, отправка запросов), `action_item` (действия), а также парсер, классы для взаимодействия с сетью и базой данных.

Клиентская часть состоит из шаблонов и сценариев на языке JavaScript.

Процедура генерации `gui` включает в себя следующие стадии:

1. Пользователь, находясь на главной странице нажимает на поле меню, после чего отправляется ajax — запрос



Рис. 3 Меню. Каждый элемент соответствует action item

2. Поступивший запрос обрабатывается представлением `ajax_request`, где на основе данных из поступившего запроса запускается определенный обработчик, а также обрабатываются исключительные ситуации.
3. Если тип соответствует `wpc_gui_ini_test`, то запускается построитель GUI. В одной директории с обработчиком располагается файл формата `aINI`, представляющий собой несколько различных секций с определениями переменных. Переменные могут быть различных типов, могут быть обязательны для заполнения (атрибут «*» перед объявлением), а также объявляться, но не отображаться (атрибут «-»). Обработчик представляет собой функцию, способную возвращать HTML код `ini` файла (рендеринг) и отправлять на обработку полученные данные по нажатию кнопки «Обработать»

Рис. 4 пример сгенерированного GUI.

4. В обработчике происходит разбор файла входных параметров, расположенного в соответствующей директории при помощи `iniparser` и формируется контекст для рендеринга по HTML шаблону. Парсинг осуществляется построчно, в контекст записывается информация о секциях, атрибутах, переменных, комментариях. На основе регулярных выражений определяется тип значения. Поддерживаются несколько типов данных: целые, вещественные числа, диапазоны, массивы, пары х-у для представления функций одной переменной, ссылки на записи таблиц баз данных и др. Разбор этих строк основан на механизме регулярных выражений и реализован с помощью стандартного модуля `re` языка Python. Итоговый контекст представляет собой список, содержащий необходимые для генерации формы параметры, такие как тип переменной, информация о секции и атрибутах. Этот список передается в шаблонизатор, где каждому типу данных ставится в соответствие некоторая часть шаблона. Так, на основе переменной логического параметра генерируется чекбокс, а на основе переменной типа целого числа — обычное поле ввода со значением по умолчанию. Для каждого типа указывается соответствующий placeholder — строка подсказка.

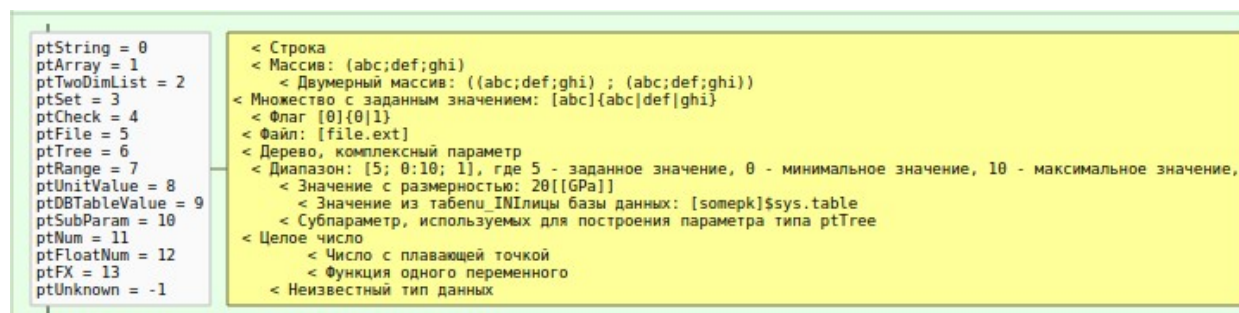


Рис. 5 Поддерживаемые параметры

Пример сформированного контекста:

```
{
    'action_item': 'wpc_gui_ini_builder',
    'action_class': 'PYWEBS',
    'aini_name': 'WPC_GUI_INI_TEST',
```

```

'processing': 'PYTHON_CALLER',
'proc_ai_class': 'PLUGIN',
'all_data':
    OrderedDict(
        [('section',
            OrderedDict(
                [
                    ('variable1', {'value': '1', 'type': <enu_INIPParamType.ptNum:
11>}),
                    ('variable2', {'value': '2', 'type': <enu_INIPParamType.ptNum:
11>}))]
            )
        ])
    )

```

В процессе парсинга были выделены значения 2 переменных, а также секция

5. На основе содержимого контекста происходит рендеринг HTML страницы в соответствии с HTML шаблоном.



Рис. 6. Схема обработки запроса

3. ТЕСТИРОВАНИЕ И ОТЛАДКА

Для тестирования необходимо подготовить aINI файл входных параметров:

```
[Author]//Идентификация автора разработки
AuthorName=[alsokolo]$sys.users//Автор разработки
*AuthorSID=sa//SID Автора
G=4.5 // Вещественное число
-OutputFilename=@AuthorSID@_@CodeObjectName@.res//

[Generator parameters]//Параметры генерации
CopyObjectToRep=[0]{0|1}//Перенести объект генерации в репозиторий (ONLINE-
MODE)
// следующие
-RepPath=/var/www/group/repos //Путь к локальной рабочей копии репозиторию
-TemplatesPath=/var/www/Templates //Путь к каталогу с шаблонами
-TemporaryPath=/var/www/${APS}/_tmp //Путь к каталогу временного хранения

[Object parameters]//Параметры генерируемого объекта
CodeObjectName=AdvancedINI//*Наименование объекта(varchar(25))
Description=Формат данных Advanced INI (aINI)//*Описание
ParametersFile=[UGD_COMFRMINI_aINI.imp]//Имя файла дополнительных
параметров
TemplateSID=[UGD]$gen.tmpls//Тип генерируемого объекта из БД

[Project data]//Идентификация объекта
ComplexSID=[com]$sys.cmplx//Идентификатор комплекса (группа GitLab)
SolutionSID=[frm]$sys.solun//Идентификатор решения (проект GitLab)
ProjectSID=[ini]$sys.prjct//Идентификатор проекта (модуль в GitLab)
```

Обработать

Author Generator parameters Object parameters Project data

Автор разработки

[alsokolo] \$

SID Автора

sa

Вещественное число

4.5

OutputFilename

@AuthorSID@_@CodeObjectName@.res

Рис. 3. Секция Author, сформированная на основе aini файла

Для запуска приложения необходимо выполнить миграции базы при первом запуске и запустить сервер при помощи команд:

```
$python3 manage.py makemigrations
```

```
$python3 manage.py migrate
```

```
$python3 manage.py runserver
```


ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы был доработан интерпретатор. Добавлен следующий функционал:

1. Обработка парсером различных типов данных, поддерживающихся в формате aINI
2. Отредактированы шаблоны для отображения типов

СПИСОК ЛИТЕРАТУРЫ

1. Федоров А.В. Современный подход в проектировании грамотного пользовательского интерфейса. //Научный вестник Воронежского Государственного Архитектурно-Строительного Университета. Сер. информационные технологии в строительных, социальных и экономических системах. - 2015. - №2. - С. 97-100.
2. Бубарева О.А. Методы проектирования эффективных экранных интерфейсов. //Информация и образование. - 2018 - №10. - С. 91-94.
3. Соколов А.П., Першин А.Ю. Программный инструментарий для создания подсистем ввода данных при разработке систем инженерного анализа. //Программная инженерия – 2017 - №8 – С. 543 – 552.
4. Литвинов В.Л., Онтологическое проектирование пользовательских интерфейсов//Информационные системы и технологии в моделировании и управлении. Сборник материалов III Всероссийской научно-практической конференции с международным участием, посвященной 100-летию Крымского федерального университета имени В.И. Вернадского. - 2018 – стр. 33-37
5. Чернов В.В. К проблеме разработки Веб – интерфейсов. //Фундаментальные исследования. - 2012 - №11-2. - С. 463 - 465.
6. Грибова В.В., Черкезишвили Н.Н., Развитие онтологического подхода для автоматизации разработки пользовательских интерфейсов с динамическими данными. //Информационные технологии, 2010, №10, -С. 54–58.
7. Глазков С.В., Ронжин А.Л., Контекстно-зависимые методы автоматической генерации многомодальных пользовательских веб-интерфейсов. //Тр. СПИИРАН. - 2012. - 21(2012) – С.170–183.
8. Zhizhimov O. L. Explain Services on ZooSPACE Platform and Adaptive User Interfaces // CEUR Workshop Proceedings. 2015. Vol. 1536. P. 30—36.
9. Пискунов С. В., Кратов С. В., Остапкевич М. Б., Веселов А. В. Использование сборочной технологии для построения пользовательских

интерфейсов сетевой информационно-вычислительной системы //

Проблемы информатики. 2010. № 4. С. 41—48

10. Ramon O. S., Cuadrado J. S., Molina J. G. Model-driven reverse engineering of legacy graphical user interfaces // Automated Software Engineering. 2014. Vol. 21, Issue 2. P. 147—186. DOI: 10.1007/s10515-013-0130-2.
11. Нешляева А.В. Актуальность инструмента построения графического интерфейса пользователя в облачных средах разработки // Альманах современной науки и образования. - 2013 - №5 – С. 134-136.