

МГТУ им. Н. Э. Баумана  
Факультет “Робототехника и комплексная автоматизация”  
Кафедра “Системы автоматизированного проектирования”

Разработка динамических пользовательских интерфейсов для  
распределенных систем инженерного анализа

Выполнил: Неклюдов С.А.  
Руководитель: Соколов А.П.

Москва, 2018

# Введение Актуальность

- В условиях конкуренции на рынке программного обеспечения компании стараются предоставить пользователям удобные средства для использования их продуктов. Зачастую конкуренцию выигрывает продукт, имеющий более эргономичный, понятный, гибкий интерфейс.
- В системах инженерного анализа редко уделяется внимание построению пользовательских интерфейсов.

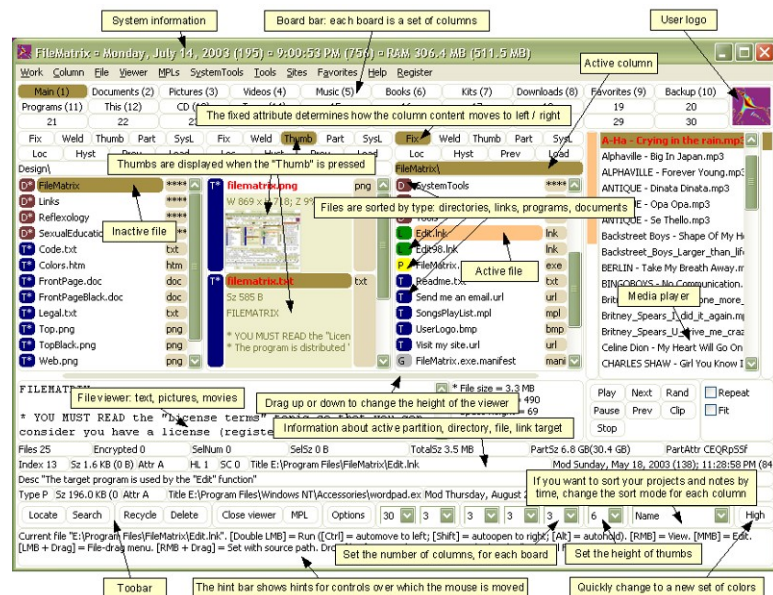


Рис 1. Плохой пример  
пользовательского интерфейса

# Введение

## Динамические пользовательские интерфейсы

- Создаются на основе переопределения существующей страницы
- Применяются для приложений, где пользователю могут понадобиться несколько экземпляров страницы.

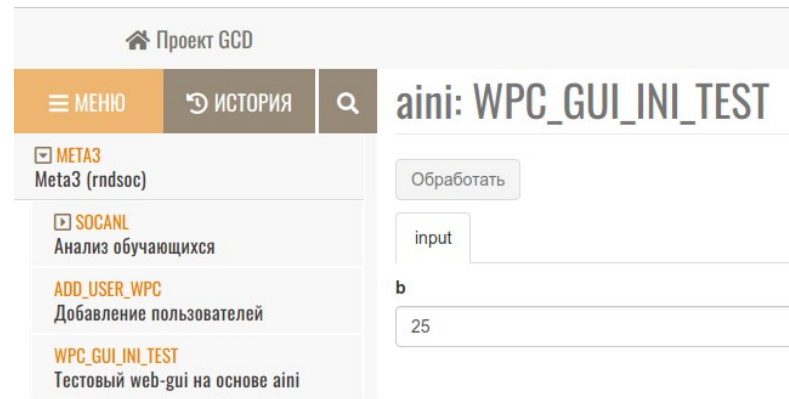


Рис. 2 .При переходе между функциями системы GCD не происходит перезагрузки страницы

Объект исследования: динамические пользовательские интерфейсы

Цель работы: доработка интерпретатора alNI и парсера iniparser для распределенной системы инженерного анализа GCD

Задачи работы:

- Ознакомление с системой comwrc
- Ознакомиться с форматом alNI
- Доработка парсера формата alNI
- 1. Обеспечить синтаксический разбор строк файлов, диапазонов, ссылок на записи базы данных, целых и вещественных числа, функций одной переменной.  
2. Обеспечить вызов функций по ссылкам на строки базы данных (RNDSOC.ANLEDT, RNDSOC.ANLSTU, RNDSOC.soc.prcmp, test\_graph\_pred6, ADD\_USER\_WPC)
- Доработка Django шаблонов для интерпретации входных параметров

# Концептуальная постановка задачи

## Входные данные

- Входными данными будут являться файлы в формате alni. Структура файла подразделяется на секции, которые содержат поля. Поля могут быть недоступны для отображения, обязательны для заполнения или произвольны.

```
[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 репозитория
-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.tpls//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)
```

Рис. 3 Пример файла в формате alni

# Концептуальная постановка задачи

## Выходные данные

- Выходными данными будет являться сгенерированная страница с отдельными блоками для секций и полями ввода для переменных различных типов, поддерживаемых форматом alNI.

# Архитектура программной реализации

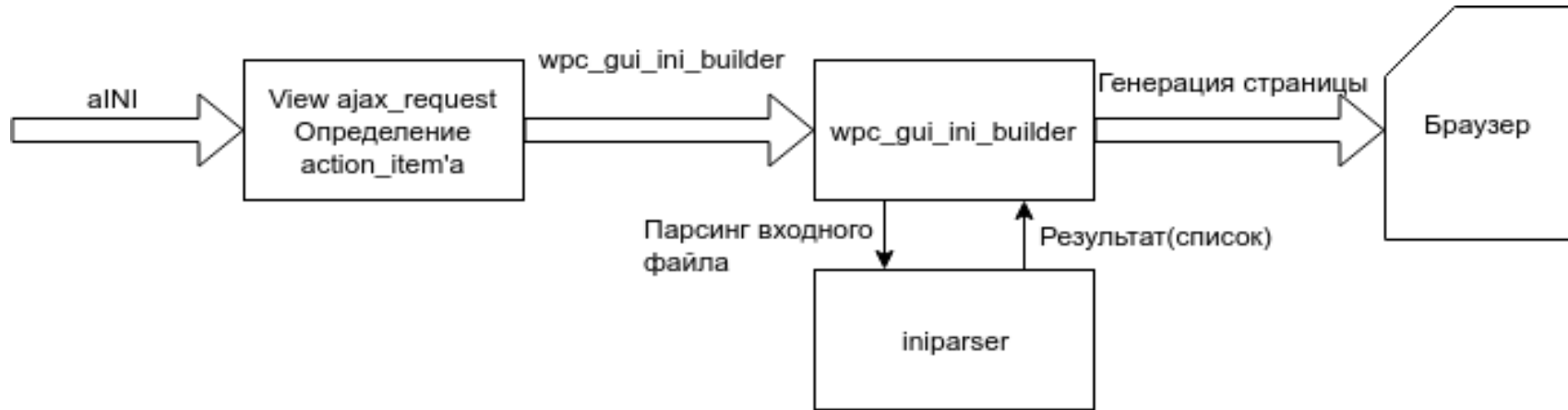
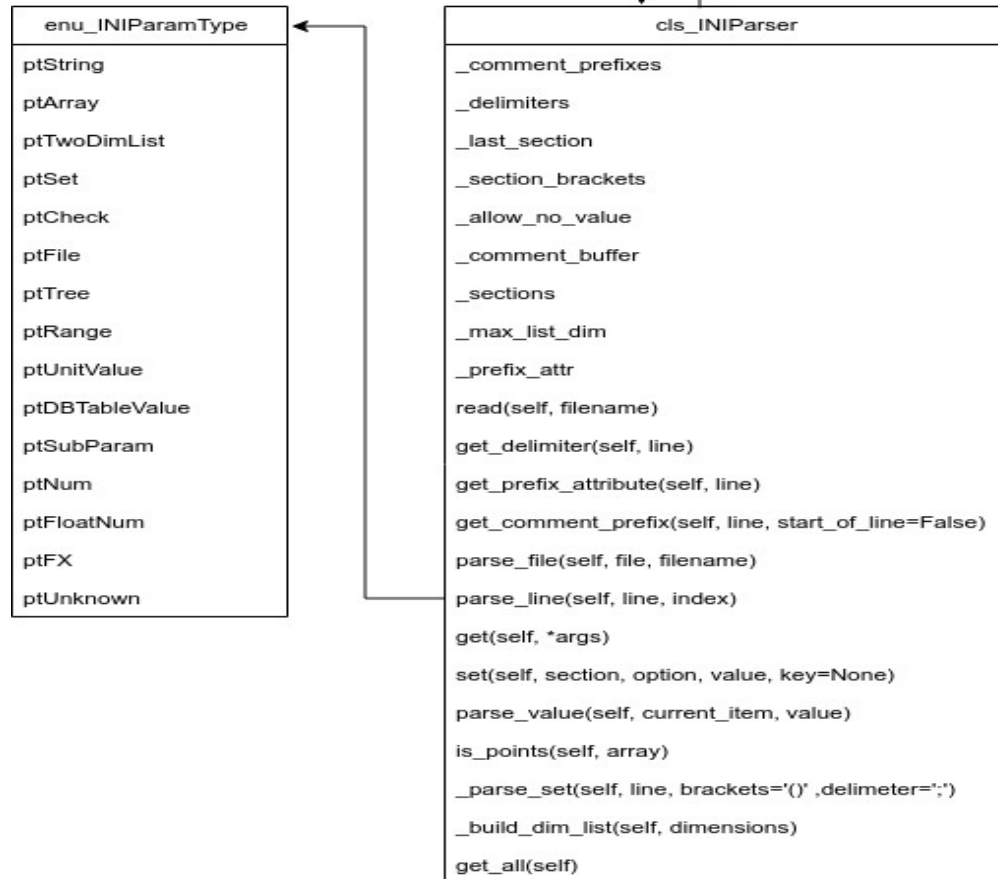


Рис. 4 Схема обработки запроса

# Архитектура программной реализации

Рис. 5 Класс `cls_INIParser`, в котором реализуется разбор входного файла





# Архитектура программной реализации

Последовательность действий:

1. Отправляется Ajax запрос.
2. Запрос обрабатывается представлением, где определяется, к какому обработчику следует его передать.
3. При передаче в генератор строка парсится и формируется список параметров, при помощи которого в шаблоне `aini_template` происходит рендеринг страницы.

# Анализ полученных результатов

aini: WPC\_GUI\_INI\_TEST

Обработать

Author

Generator parameters

Object parameters

Project data

**Автор разработки**

Ссылка на таблицу базы данных

**SID Автора**

sa

**Вещественное число**

4.5

Рис. 2. Результат генерации  
формы из файла входных данных

## Заключение

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

1. Обработка парсером различных типов данных, поддерживающихся в формате a1Ni(файлы, диапазоны, ссылка на запись базы данных, целые и вещественные числа, функции одного переменного)
2. Отредактированы шаблоны для отображения типов.

Спасибо за внимание!