

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ Н.Э.БАУМАНА»
(МГТУ им. Н.Э. Баумана)

**Разработка web-ориентированного
программного обеспечения,
реализующего автоматизированное
построение динамических
графических пользовательских
интерфейсов**

Студент:
Научный руководитель:

Василян А.Р.
Соколов А.П.

Москва, 2023



Содержание доклада

1. Введение
2. Постановка задачи
3. Программная реализация
4. Тестирование
5. Заключение

Графический пользовательский интерфейс

Интерфейс¹ — это совокупность средств, методов и правил взаимодействия, управления, контроля и т.д. между элементами системы.

Пользовательский интерфейс¹ — это разновидность интерфейсов, в котором одна сторона представлена человеком-пользователем, другая — машиной-устройством.

Графический пользовательский интерфейс¹ (GUI) — это разновидность пользовательского интерфейса, в котором элементы интерфейса, представленные пользователю на дисплее, исполнены в виде графических изображений.

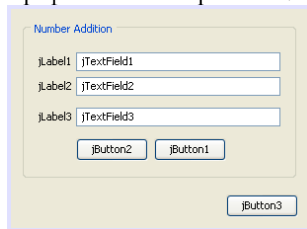


Рис. 1. Пример GUI

[1] Лукьянов Д. В. Разработка графического пользовательского интерфейса // Новые информационные технологии в автоматизированных системах. 2012

Цель и задачи работы

Цель: разработать web-ориентированное программное обеспечение, обеспечивающее автоматизацию построения динамических пользовательских интерфейсов.

Задачи:

- Рассмотреть существующие подходы разработки графического пользовательского интерфейса.
- Разработать тестовое web-приложение и запустить его на тестовом сервере.
- Разработать программное обеспечение для генерации интерфейса.
- Запустить разработанное web-приложение с использованием сгенерированного интерфейса.

Подходы к разработке пользовательского интерфейса

На основе пользовательских целей¹.

Методы взаимодействия человека и ЭВМ:

1. **Ограничительный.** Пользователь должен иметь необходимые знания для того, чтобы самому планировать ход выполнения своего задания, используя предоставляемые ему операции.
2. **Направляющий.** Каждая из целей соответствует определенному пользовательскому заданию, которое может выполнить ЭВМ, взаимодействуя с пользователем.

Методический подход², который разделен на 4 составных элемента.

- Мониторинг действий оператора.
- Применение типовой системы показателей качества.
- Программирование и документирование пользовательского интерфейса.
- Интерпретация сценария воздействия пользователя в стандартные программные процедуры.

Так же существует **метод построение пользовательского интерфейса с использованием интерактивного машинного обучения**³.

[1] Санковский Ю.Е. Метод построения оконного интерфейса пользователя на основе моделирования пользовательских целей. 1998

[2] Казаков Г.В., Корянов В.В., Чемирисов В.В., Уваров А.В. Методический подход к созданию универсального пользовательского интерфейса. 2020.

[3] Юркин В.А., Сараджишвили С.Э. Построение пользовательского интерфейса с использованием интерактивного машинного обучения. 2020.

Постановка задачи

Построение графического пользовательского интерфейса осуществляется в результате интерпретации описания элементов интерфейса на предметно-ориентированных языках (Domain-Specific Language, DSL).

В рамках проекта был выбран aINI из-за несложного синтаксиса.

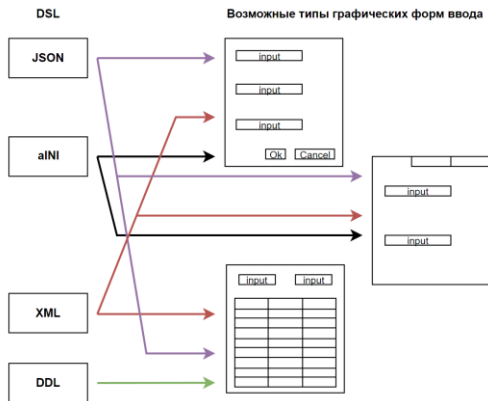


Рис. 2. Схема базового принципа генерации GUI на основе DSL

Особенности генерации GUI на основе aINI

В разработке программы для генерации HTML файла на основе aINI использовалась библиотека `ryparsing`¹.

Ниже рассмотрено распознавание строки aINI на примере элемента поля ввода. Строка поля ввода распознаётся программой согласно шаблону :

```
variable + '=' + word_num + '//' + rus_eng_word_num
```

где `variable` – слово, в котором допускаются буквы латинского алфавита, цифры и символ “_”,

`word_num` – слово, в котором допускаются цифры и буквы латинского алфавита,

`rus_eng_word_num` – слово или несколько слов, в которых допускаются буквы русского и латинского алфавита и цифры.

Пример строки, соответствующей этому шаблону, представлен на листинге 1, а элемент интерфейса, созданный на её основе, на рис. 3.

```
x=12//Параметр X
```

Листинг 1. Пример описания поля ввода в формате aINI

Параметр X

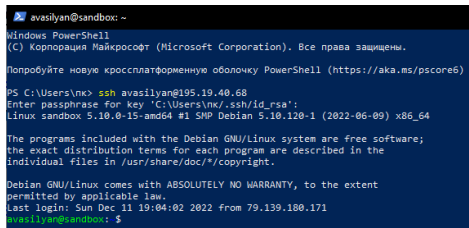
12

Рис. 3. Пример сгенерированного поля ввода

[1] Ryparsing — модуль синтаксического анализа для языка Python.

Разработка тестового web-приложения

Во время разработки использовались: Django¹, Docker², Nginx³.



```
avasilyan@sandbox: ~  
Windows PowerShell  
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.  
  
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)  
  
PS C:\Users\nk> ssh avasilyan@195.19.40.68  
Enter passphrase for key 'C:\Users\nk\.ssh\id_rsa':  
Linux sandbox 5.10.0-15-amd64 #1 SMP Debian 5.10.120-1 (2022-06-09) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Dec 11 19:04:02 2022 from 79.139.180.171  
avasilyan@sandbox: $
```

Рис. 4. Вход на сервер

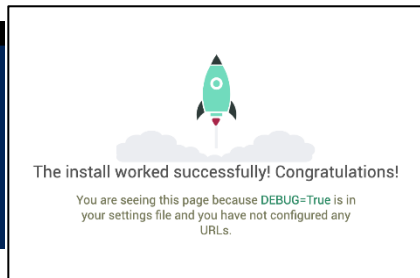


Рис. 5. Приветственное окно Django

[1] Django — это Python web-фреймворк, с помощью которого можно вести разработку web-приложения.

[2] Docker — программное обеспечение, применяемое для разработки web-приложений в средах с поддержкой контейнеризации.

[3] Nginx — веб-сервер и почтовый прокси-сервер.

Особенности генерации интерфейса с применением Django

`urlpatterns` (листинг 1) – список всех URL, которые обрабатываются web-приложением.

`menu` (листинг 2) – функция-представления, которая сопоставлена с адресом URL с помощью функции `path()` и которая обрабатывает запрос по этому адресу.

```
...
urlpatterns = [
    path("", menu, name="menu"),
    path("F1/", input1, name="input1"),
    path("F2/", input2, name="input2"),
    path("F3/", input3, name="input3")
]
...
```

Листинг 2. Содержимое файла `urls.py`

В функции `menu` используется HTML файл (`menu.html`), который был сгенерирован с помощью преобразователя данных в формате aINI в HTML-код.

```
def menu(request):
    return render(request, "menu.html")
```

Листинг 3. Функция-представления `gui` в `views.py`

Примеры генерации GUI на основе aINI

```
F1 = [input1]//Test1  
F2 = [input2]//Test2  
F3 = [input3]//Test3
```

Листинг 4. config файл

Test1

Test2

Test3

Рис. 6. Результат генерации (Меню)

Примеры генерации GUI на основе aINI

```
[sec1]//Вкладка 1
x=25//Параметр X
y=@y@//Параметр Y
box1=[0]{0|1}//Флажок 1
box2=[1]{0|1}//Флажок 2
[sec2]//Вкладка 2
q=ABC//Параметр Q
box3=[0]{0|1}//Флажок 3
ParametersFile=[file]//Выберите требуемый файл
//МГТУ им. Н. Э. Баумана
[https://bmstu.ru]//Дополнительная информация
```

Листинг. 5. Входные данные в формате aINI

Вкладка 1	Вкладка 2
<p>Параметр X <input type="text" value="25"/></p> <p>Параметр Y <input type="text"/></p> <p><input type="checkbox"/> Флажок 1 <input checked="" type="checkbox"/> Флажок 2</p> <p><input type="button" value="Отправить"/> <input type="button" value="Назад"/></p>	<p>Параметр Q <input type="text" value="ABC"/></p> <p><input type="checkbox"/> Флажок 3</p> <p>Выберите требуемый файл <input type="button" value="Выберите файл"/> <input type="button" value="Файл не выбран"/></p> <p>МГТУ им. Н. Э. Баумана Дополнительная информация</p> <p><input type="button" value="Отправить"/> <input type="button" value="Назад"/></p>

Рис. 7. Результат генерации (страница с полями ввода)

Выводы

1. Разработано программное обеспечение для преобразования файлов в формате aINI в файлы формата HTML и генерации дополнительного Python-кода.
2. В рамках Django было разработано web-приложение, использующее сгенерированный интерфейс.
3. Автоматизированное построение GUI на основе данных в формате с простым синтаксисом (например, aINI) позволяет быстро создавать графические формы ввода.
4. Разработанное программное обеспечение, в том числе удобно тем, что формат ввода и вывода стандартизированы, что обеспечивает доступность результата генерации автору исходного файла в формате aINI.

Дальнейшие перспективы развития

1. Разработанное программное обеспечение имеет смысл внедрить в инструментарий некоторой системы для упрощения разработки интерфейса.
2. Добавление большего числа элементов, возможных для генерации.
3. Добавление возможности редактировать стили (внешний вид) элементов через описание на aINI