

Отчет о проведенном аналитическом обзоре литературы

Студент	Неклюдов Семен Александрович
Группа	РК6-71
Тип задания	аналитический обзор литературы
Тема исследования:	Методы автоматической генерации пользовательских интерфейсов

Студент	<hr/>	Неклюдов С.А. <hr/>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Преподаватель	<hr/>	А.П. Соколов <hr/>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Москва, 2018 г.

Оглавление

Задание.....	3
Введение.....	4
1. Результаты поиска источников литературы.....	5
2. История развития объекта поиска.....	5
Заключение.....	5
Список найденных источников.....	5

Задание

Аналитический обзор литературы проводился в рамках выполнения работ по разработке динамических пользовательских интерфейсов для распределенных систем инженерного анализа.

Объект исследования: динамические пользовательские интерфейсы.

Объект поиска: существующие решения в области методов автоматической генерации пользовательских интерфейсов

Ключевые слова: методы построения пользовательских интерфейсов, gui generation, user-interface generation, генерация GUI, Django, automated GUI generation, UI, MVC, модель-представление-контроллер,

Основная задача аналитического обзора литературы: изучить современные подходы к разработке динамических пользовательских интерфейсов и ознакомиться с их реализациями, доработать функционал GUI генератора системы инженерного анализа GCD.

Задачи аналитического обзора литературы (детально):

1. Провести поиск источников литературы (преимущественно научных публикаций) согласно определенным ключевым словам.
2. Определить историю развития объекта поиска и выявить основные тенденции развития.
3. Определить наиболее перспективные современные разработки и направления развития объекта поиска.
4. Определить перечень УДК, к которым относится объект поиска¹.
5. Составить список найденных источников согласно ГОСТ Р 7.0.5-2008.

Начало поиска 01.10.18. Окончание поиска 07.10.18.

Введение

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

По оценкам различных 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.

Так, в системах инженерного анализа выделяют три составляющих их программных модулей:

- 1) модули подготовки данных
- 2) модули обработки данных
- 3) модули представления результатов

В системах, 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].

Таким образом, методы формирования динамических графических интерфейсов являются важной частью программной системы.

1. Результаты поиска источников литературы

В результате поиска было найдено несколько методов построения динамических пользовательских интерфейсов, описанных в приведенных статьях[6-10].

Также был изучен подход, используемый в системе инженерного анализа GCD, изучен текстовый формат, разработанный для представления списка входных данных программной системы.[3]

2. История развития объекта поиска. Основные тенденции развития

История разработки пользовательских интерфейсов берет свое начало в конце 1970-х годов, в момент выхода на рынок вычислительной техники персональных компьютеров. Тогда получил распространение один из первых методов разработки программ — Waterfall model. Процесс разработки не был итеративным, что означало, что каждый этап выполняется лишь один раз. К концу 80-х годов назрело понимание отличий процесса разработки программного обеспечения от инженерной деятельности, заключавшихся в необходимости частого внесения модификаций в объект разработки. В результате для создания информационных систем распространилась модель быстрой разработки приложений (RAD), представляющей собой по сути визуальное проектирование. Широкое распространение получили интегрированные среды, поддерживающие языки 3 и 4 поколения, такие как Visual Basic, Delphi, Oracle Forms. Они сокращают время разработки, облегчая разработку графического интерфейса пользователя и связывая доступ к данным с графическими компонентами. Однако развитие этих приложений было затруднено в долгосрочной перспективе, поскольку бизнес-логика имела тенденцию смешиваться с логикой графического интерфейса. Это побудило большое количество предприятий перевести свои унаследованные системы RAD на новые платформы (обычно веб-платформы), которые лучше удовлетворяют их потребности в расширяемости, поддержке или распределении.[10]

Сегодня сформировались две основные парадигмы автоматизации разработки пользовательского интерфейса: дизайнерская и моделиориентированная. В рамках дизайнерской парадигмы разработаны

методы высокоуровневого проектирования визуального представления пользовательских WIMP — интерфейсов и автоматической генерации кода на некоторый язык программирования (Pascal, C#, C++, Java и др.).

Проектирование интерфейса, основанного на дизайнерской парадигме, осуществляется с помощью строителей интерфейсов, входящих в состав интегрированных пакетов и CASE- средств. В рамках модели ориентированного подхода разработаны методы раздельного проектирования и реализации интерфейса и прикладной программы, а также методы автоматической генерации кода пользовательского интерфейса по его модели и связывания этого кода с прикладной программой. [4]

3. Существующие перспективные современные разработки

В результате поиска было найдено несколько методов построения динамических пользовательских интерфейсов, описанных в приведенных статьях [6-9].

- Онтологический [6]
- Контекстно-зависимый [7]
- Адаптивный [8]
- Сборочный [9]
- Модели ориентированный [10]

Заключение

На основе проведенного обзора литературы и с учетом особенностей разработки ПО инженерного анализа были выявлены требования для программного инструментария, позволяющие сократить затраты на ввод и подготовку исходных данных для вычислительных задач [3]:

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

Список найденных источников 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. Грибова В.В. Автоматизация проектирования, реализации и сопровождения пользовательского интерфейса на основе онтологического подхода : Автореферат диссертации на соискание ученой степени доктора технических наук: 05.13.11/ Грибова Валерия Викторовна – Институт автоматики и процессов управления Дальневосточного отделения РАН; 2007 - 41
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.