

В. С. Масис¹✉, Е. Ю. Воронкин¹

Разработка методики проектирования интерактивного интерфейса интеллектуальных систем

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: masis.vova@bk.com

Аннотация. В статье рассматривается разработка методики проектирования интерактивного интерфейса интеллектуальных систем. Актуальность исследования обусловлена необходимостью создания методики проектирования интерактивного интерфейса, представляющей комплексный подход к разработке интерактивного интерфейса интеллектуальных систем. Традиционные подходы фокусируются на отдельных аспектах разработки, игнорируя необходимость целостного подхода. Это приводит к разрозненным решениям, которые не обеспечивают полноценного взаимодействия пользователя с системой. В ходе выполнения были реализованы следующие задачи: анализ предметной области, разработка методики проектирования интерактивного интерфейса интеллектуальных систем, программная реализация методики проектирования интерактивного интерфейса интеллектуальных систем. Для решения поставленных задач были использованы методы анализа предметной области и существующих решений, применение паттернов проектирования, применение современных технологий.

Ключевые слова: веб-ресурс, проектирование, верстка, программирование

V. S. Masis¹✉, E. Yu. Voronkin¹

Development of a methodology for designing an interactive interface for intelligent systems

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: masis.vova@bk.com

Abstract. The article discusses the development of a methodology for designing an interactive interface for intelligent systems. The relevance of the research is due to the need to create an interactive interface design methodology that represents an integrated approach to the development of an interactive interface for intelligent systems. Traditional approaches focus on individual aspects of development, ignoring the need for a holistic approach. This leads to disparate solutions that do not ensure full user interaction with the system. In the course of the implementation, the following tasks were implemented: domain analysis, development of a methodology for designing an interactive interface for intelligent systems, software implementation of a methodology for designing an interactive interface for intelligent systems. To solve the tasks set, methods of domain analysis and existing solutions, the use of design patterns, and the use of modern technologies were used.

Keywords: web resource, design, layout, programming

Введение

Современные интеллектуальные системы активно внедряются в различные сферы деятельности человека, начиная от бытовых устройств и заканчивая сложными промышленными решениями.

Для эффективного взаимодействия с интеллектуальными системами пользователям необходимы интуитивно понятные и удобные интерфейсы, который обеспечивает доступ к функционалу и возможности системы.

Однако существующие подходы к проектированию интерфейсов зачастую не учитывают специфики интеллектуальных систем, что приводит к снижению их эффективности и неудовлетворенности пользователей, это подчеркивает необходимость разработки новых методик, ориентированных на создание интерактивного интерфейса, которые учитывают современные технологии и особенности взаимодействия с интеллектуальными системами.

Современные интеллектуальные системы требуют интерактивных интерфейсов, способных адаптироваться к потребностям пользователей и контексту их использования, проектирования таких интерфейсов представляет собой сложную задачу, объединяющую технические, когнитивные и дизайн-ориентированные подходы.

Такие интерфейсы способны адаптироваться под индивидуальные предпочтения и контекст использования, что особенно актуально для интеллектуальных систем.

Основным недостатком существующих методик в контексте интеллектуальных систем, является фокусировка на отдельные аспекты разработки, что не дает полной картины того, как данная методология должна применяться в системе целиком.

Фокусировка на отдельных аспектах разработки представляет собой подход, при котором внимание концентрируется на определенных частях процесса проектирования, таких как визуальный дизайн, пользовательский опыт или техническая реализация, при этом остальные аспекты могут оставаться недостаточно проработанными.

Такой подход часто применяется в методологиях, стремящихся к оптимизации отдельных этапов, например, в Design Thinking, где акцент делается на генерацию идей и креативность [1]. Несмотря на преимущества, этот подход может привести к игнорированию важных элементов, таких как интеграция или аспекты архитектуры, что существенно ограничивает общую эффективность процесса.

Узкая фокусировка на отдельных аспектах разработки существенно влияет на качество конечного продукта. Согласно исследованию компании McKinsey, 70 % проектов терпят неудачу из-за отсутствия целостного подхода [2].

Недостаточная проработка интеграции различных элементов системы приводит к несоответствию между требованиями пользователей и реализованным функциями. Акцент на одном аспекте приводит к пропуску критически важных этапов, значительно снижая общую эффективность и результативность процесса разработки.

Для минимизации недостатков, связанных с узкой фокусировкой, важно применять комплексный подход, учитывающий все аспекты проектирования.

Одной из основных проблем в разработке интеллектуальных систем является недостаточная адаптивность к изменениям требований. В условиях дина-

мично меняющихся рыночных условий и потребностей пользователей способность системы к быстрой и эффективной адаптации становится критически важной [3]. Исследование Standish Group показывает, что 66 % проектов в сфере разработки программного обеспечения сталкиваются с необходимостью изменения требований в процессе реализации, что подчеркивает ограничения применяемых методологий.

Причины ограниченной адаптивности часто связаны с недостаточной гибкостью используемых методологий и инструментов. Методология Agile, хотя и ориентирована на адаптацию, испытывает трудности при интеграции изменений на поздних этапах разработки.

Кроме того, отсутствие четко определенных процессов управления изменениями и недостаточная коммуникация между участниками проекта усугубляют проблему. Согласно исследованию McKinsey & Company, 17 % проектов проваливаются именно из-за недостаточной адаптивности, что подчеркивает важность комплексного подхода к управлению проектами, результаты этого исследования, представлены на рисунке 1.

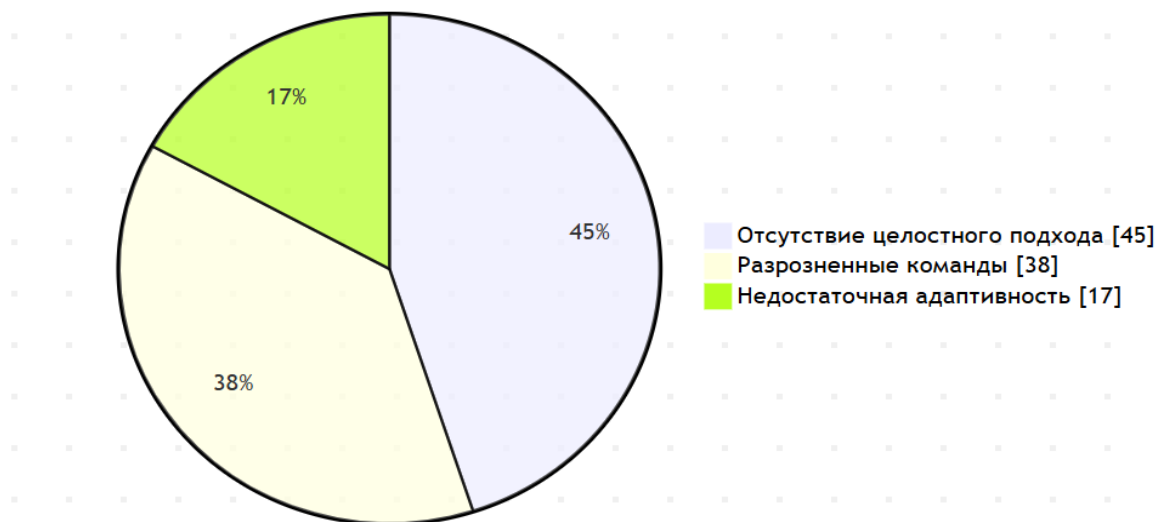


Рис. 1. Исследование компании McKinsey & Company

Методы и материалы

Первым этапом при разработке, является определение требований к интерфейсу. На этом этапе необходимо собрать и проанализировать информацию о потребностях пользователей и целях системы [11]. Основной задачей данного этапа, является формирование четкого представления о том, какие функции должен выполнять интерфейс и каким образом он будет взаимодействовать с пользователем, данный процесс включает в себя опросы, интервью и анализ существующих решений.

На данном этапе также происходит процесс проектирования концептуальной модели, которое включает в себя создание общей структуры и логики взаимодействия системы с пользователем.

Разработка прототипа, позволяет визуализировать и протестировать основные элементы системы. Прототипирование помогает выявить проблемы и улучшить функциональность до начала программирования.

Следующим этапом, является функционального проектирования, которое включает детализацию функциональности интеллектуальной системы, определение возможностей, которые система должна обеспечивать.

Реализация программного обеспечения интерфейса представляет собой заключительный этап, на котором концепции и прототипы превращаются в работающую систему.

При реализации программного обеспечения, важно учитывать инструменты разработки интерфейсов, которые предоставляют разработчикам спектр возможностей для создания удобного интерактивного интерфейса.

Паттерны проектирования интерфейсов представляют собой проверенные решения для типовых задач, возникающих при создании пользовательских интерфейсов.

Тестирование интерфейсов также является важным этапом при разработке интерфейсов интеллектуальных систем, данный этап позволяет выявить недостатки в процессе работы системы.

Unit-тесты, важная часть функционального тестирования, которая включает проверку отдельных компонентов или модулей кода (юнитов) в изоляции. Целью является убедиться, что каждый юнит работает правильно, прежде чем интегрировать его с другими частями системы.

Интеграционное тестирование, важный этап разработки программного обеспечения, который фокусируется на проверке взаимодействия между различными модулями или компонентами системы.

Mock-тесты используются, когда юнит зависит от других компонентов, которые еще не разработаны или сложны в настройке для тестирования. Общая схема Mock тестирования, представлена на рисунке 2.



Рис. 2. Представление тестирования при помощи Mock-инструментов

Сравнение различных методов тестирования позволяет выбрать наиболее подходящий подход в зависимости от целей и ресурсов проекта.

Анализ собранной информации представляет собой первый шаг в процессе доработки интерфейса. Этот этап включает систематизацию и интерпретацию данных, полученных в ходе тестирования и сбора обратной связи [12].

На основе анализа собранных данных необходимо определить приоритетные изменения, которые окажут наибольшее влияние на пользовательский опыт.

Результаты

Процесс проектирования начинается с глубокого анализа требований, где определяются как функциональные характеристики системы, так и потребности пользователей системы. На этапе проектирования архитектуры выбираются оптимальные паттерны и технологии.

Создание интерактивных прототипов, позволяет визуализировать интерфейс и провести раннее тестирование с пользователями. При реализации бизнес-логики строго соблюдаются выбранные принципы проектирования, что обеспечивает высокое качество кода. Интеграция с базой данных включает настройку репликации и кластеризации для отказоустойчивости, реализацию миграций и тщательную оптимизацию производительности.

Заключительный этап, это всестороннее тестирование, включающее проверку соответствия требованиям, ручного, функционального и нагрузочного тестирования [16]. Такой комплексный подход гарантирует создание интерактивного интерфейса, который не только отвечает современным стандартам качества, но и обеспечивает исключительный пользовательский опыт.

При проектировании архитектуры современного веб-приложения особое внимание уделяется выбору инструментов разработки, основанный на основных потребностях разрабатываемой системы. Для клиентской части были выбраны инструменты HTML, CSS, JavaScript с фреймворком Bootstrap, для серверная часть выбраны инструменты Java с использованием фреймворка Spring Framework.

Такой выбор обусловлен рядом важных преимуществ выбранного набора технологий и особенностью проектируемой системы.

В качестве системы сборки проекта был выбран инструмент Gradle за особенностью производительность и гибкость в сравнении с другим сборщиком проекта Maven [17].

Для тестирования серверной части используется комбинация использования технологий JUnit и Mockito, а также проведение интеграционных тестов. JUnit является стандартом модульного тестирования, предоставляя простой и эффективный способ проверки отдельных компонентов.

Mockito дополняет его возможностью создания mock-объектов, что важно при тестировании компонентов с внешними зависимостями.

Интеграционные тесты проверяют взаимодействие между модулями и с внешними системами.

Для контейнеризации приложения был выбран инструмент Docker, обеспечивающий идентичность окружений на всех стадиях разработки и развертывания [18].

Предложенный технологический стек формирует сбалансированное решение, где каждая технология дополняет другие, создавая целостность разрабатываемой системы. HTML, CSS и JavaScript с Bootstrap обеспечивают быструю разработку адаптивных интерфейсов, Java, Spring, Gradle и Docker создают надежную, безопасную разработку серверной логики системы.

Предложенная методика, сочетающая проверенные паттерны проектирования, современные принципы разработки и четкий процесс реализации, представляет собой оптимальный инструмент для создания интерактивного интерфейса систем. Ее применение позволяет разрабатывать системы, которые отличаются высокой гибкостью, масштабируемостью, что подтверждает ее актуальность и практическую ценность в условиях быстро развивающихся цифровых технологий.

Заключение

Разработанная методика обладает высокой практической ценностью и может быть успешно внедрена в корпоративные системы, обеспечивая эффективное взаимодействие пользователей с интеллектуальными системами. Реализованная система бронирования наглядно демонстрирует преимущества предложенного подхода, включая адаптивность интерактивного интерфейса. Применение современных паттернов проектирования, а также передовых технологий разработки, гарантирует гибкость и масштабируемость решения, что делает его применимым в различных предметных областях.

Перспективы дальнейшего развития связаны с интеграцией мультиагентного подхода, включает расширение функциональности системы, включая поддержку новых сценариев использования и интеграцию с дополнительными сервисами. Важным направлением остается оптимизация производительности и улучшение пользовательского опыта, что будет способствовать дальнейшему повышению эффективности внедряемых решений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Браун, Т. Design Thinking в бизнесе: От разработки новых продуктов до проектирования бизнес-моделей : [пер. с англ.] / Т. Браун. – Москва : Альпина Паблишер, – 2020. – 256 с. ISBN 978-5-00100-829-3. – Текст : непосредственный.
2. Шалфеева, Е. А. Методы, модели и технология обеспечения жизнеспособности интеллектуальных систем с декларативными базами знаний : автореф. дис. ... д-ра техн. наук : 05.13.01 / Е. А. Шалфеева. Владивосток, – 2021. – 44 с. Библиогр.: – с. 40–44. – Текст : непосредственный.

3. Голенков, В. В. Адаптивные интеллектуальные системы: проблемы и решения / В. В. Голенков, Е. А. Смирнова // Информационные технологии. – 2022. № 5. – С. 34–45. – Текст : непосредственный.
4. Мартин, Р. Чистая архитектура: искусство разработки программного обеспечения : [пер. с англ.] / Р. Мартин. Санкт-Петербург : Питер, – 2021. – 352 с. ISBN 978-5-4461-0772-8. – Текст : непосредственный.
5. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования : [пер. с англ.] / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. Санкт-Петербург : Питер, – 2023. – 512 с. ISBN 978-5-496-00389-6. – Текст : непосредственный.
6. Рочев, К. В. Информационные технологии. Анализ и проектирование информационных систем : учеб. пособие для вузов / К. В. Рочев. 4-е изд., стер. Санкт-Петербург : Лань, – 2025. – 128 с. ISBN 978-5-507-50803-7. – URL: <https://e.lanbook.com/book/465164> (дата обращения: 11.03.2025). – Режим доступа: для авториз. пользователей. – Текст : электронный.
7. Смирнов, М. В. Проектирование и разработка информационных систем и бизнес-приложений : методические указания / М. В. Смирнов, Р. А. Исаев, Р. С. Толмасов. Москва : РТУ МИРЭА, – 2020. – 44 с. – URL: <https://e.lanbook.com/book/163878> (дата обращения: 13.03.2024). – Режим доступа: для авториз. пользователей. – Текст : электронный.

© В. С. Масис, Е. Ю. Воронкин, 2025