

Методический подход к созданию универсального пользовательского интерфейса

© Г.В. Казаков¹, В.В. Корянов², В.В. Чемирисов¹, А.В. Уваров¹

¹ФГБУ «4 ЦНИИ» Минобороны России,
г. Королёв, Московская обл., 141091, Россия
²МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

При проведении исследований в научных организациях Министерства обороны Российской Федерации решаются задачи, связанные с применением информационных технологий в военной сфере, а полученные результаты (программное, математическое и методическое обеспечение, модели и т. д.) ориентированы на военных специалистов. Часто в ходе исследований сопутствующими научными результатами являются технологии (методики, модели), которые ориентированы на гражданскую направленность и могут применяться предприятиями промышленности (в военных разработках). Представлена в настоящей статье перспективная технология автоматического программирования программных средств подготовки данных. В информатике термин «автоматическое программирование» идентифицирует тип компьютерного программирования, в котором механизм генерирует компьютерную программу, чтобы позволить человеку-программисту писать код на более высоком уровне абстракции. Новая абстракция, которую применяют при программировании в рамках предлагаемой технологии, — это «сценарий воздействий пользователя на элементы управления пользовательского интерфейса программы». Сведениями, необходимыми для построения программной архитектуры, основанной на сценариях воздействий пользователя, являются перечень функций программы и описывающая данные структура, типы элементов управления, типы манипуляторов ввода и типы программных событий, генерируемых элементами управлений в результате воздействий манипуляторов ввода.

Ключевые слова: архитектура, ввод данных, качество, моделирование, оператор, программное средство, проектирование, пользовательский интерфейс, разработка

Введение. При разработке программных средств систем вооружения (ПССВ) внесение в них изменений после проведения Государственных испытаний и ввода в эксплуатацию возможно только по решению главного конструктора системы вооружения. У должностных лиц эксплуатирующих организаций часто возникают потребности в доработках элементов взаимодействия, в частности пользовательского интерфейса программных средств, актуальность которого возрастает в ходе эксплуатации. При этом мероприятия, необходимые для удовлетворения потребностей в таких улучшениях, заведомо не могут финансироваться в рамках сервисного обслуживания, а внесение изменений в систему вооружения, введенную в эксплуатацию, повлечет корректировку конструкторской документации и необходимость очередных испытаний. Этим обусловлена актуальность

задачи проектирования и разработки программных средств, которая состоит в поиске компромисса между разработчиком и заказчиком на той стадии проекта, когда техническое задание (ТЗ) согласовано, но утвержденные в нем требования к пользовательскому интерфейсу нуждаются в проработке, уточнении. Одним из путей решения этой задачи является использование гибкой методологии разработки проекта [1–8], при которой будет гарантировано плотное взаимодействие представителей заказчика (в том числе представителей эксплуатирующей организации) и разработчика.

Цель статьи — разработать методический подход к созданию универсального пользовательского интерфейса (далее — подход), позволяющий выбрать такую архитектуру программных средств, которая гибко и оперативно будет подстраиваться под изменяющиеся требования пользователей ПССВ.

Подход к созданию универсального средства построения пользовательского интерфейса программных средств подготовки данных. В основе подхода лежит предлагаемая в [9] UML нотация, используемая для описания структуры программы (рис. 1). В работах [10–12] приводятся доводы в пользу необоснованно редкого использования макетирования при формализации интерфейсов.

На рис. 1 изображены следующие основные элементы подхода:

- мониторинг действий оператора;
- применение типовой системы показателей качества пользовательского интерфейса;
- автоматизация программирования и документирования пользовательского интерфейса;
- интерпретация сценариев воздействий пользователя.

Прежде чем раскрывать назначение перечисленных элементов, необходимо определить, что подразумевается под «Единой структурой сведений о предметной области».

Под единой структурой предметной области (рис. 2) понимается унифицированная пятиуровневая система структуры модели данных программного средства, сформированная на основе анализа предметной области (либо декомпозиции функций существующего программного средства).

1. Задачи. Здесь перечисляются все корневые режимы программы, основные функции. Часть из них может соответствовать отдельным информационно-расчетным задачам, перечисленным в ТЗ на разработку программного средства. Например, в программе коммуникатора один из корневых режимов — «Телефонный справочник».

2. Операции. Данный уровень подразумевает разделы и подразделы, подчиненные по своей семантике некоторой задаче (или задачам), и служит для описания переходов между операциями, вложенности операций. В программе коммуникатора для режима «Телефонный

справочник» примером операции может быть «Добавить новый контакт».

3. Действия. Содержится перечень полей модели данных, подлежащих ассоциированию с элементами управления пользовательского интерфейса программы в зависимости от операции. Например, в программе коммуникатора для режима «Телефонный справочник» операция «Добавить новый контакт» может состоять из следующих действий: ввод имени контакта, ввод номера телефона, сохранение и т. п.

4. Атомарные операции. Определяются все сопутствующие действия по изменению состояния элементов управления, которые должен выполнить оператор. Все атомарные операции определяются типом элементов управления.

5. Воздействия. Отражаются все воздействия оператора от манипуляторов ввода, которые программной средой преобразуются в атомарные операции.

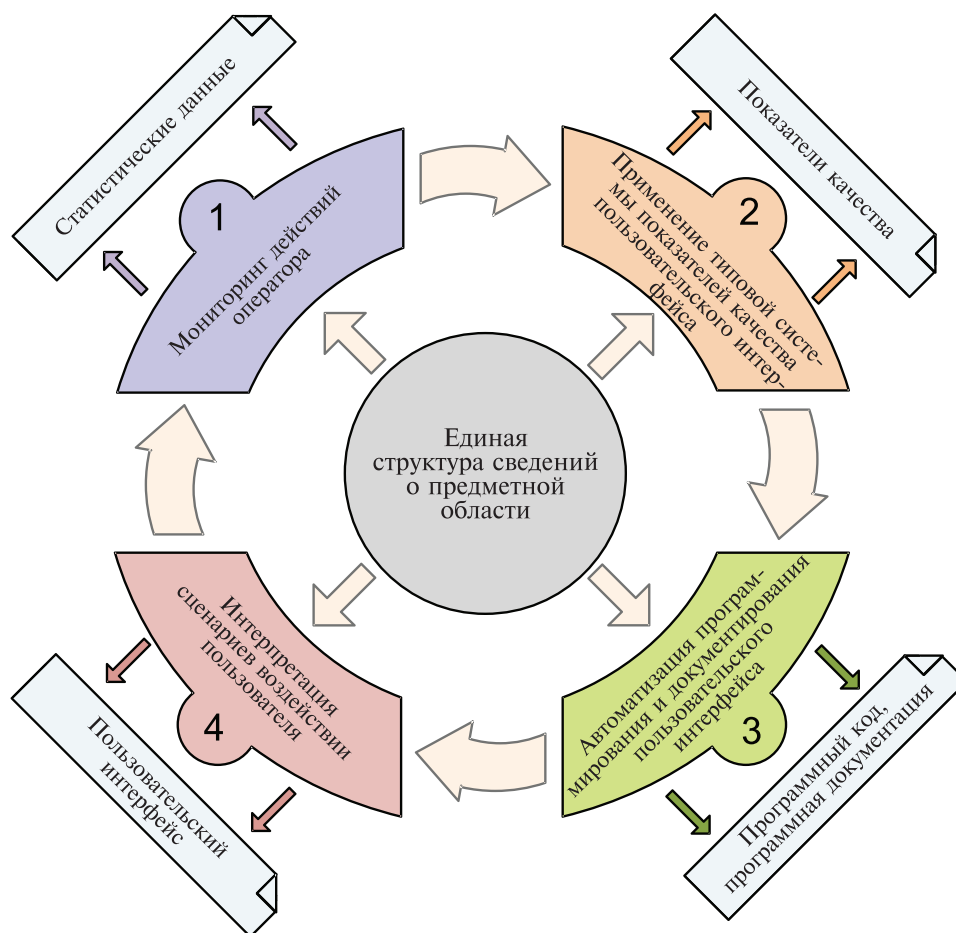


Рис. 1. Составные элементы подхода к созданию универсального средства построения пользовательского интерфейса программных средств

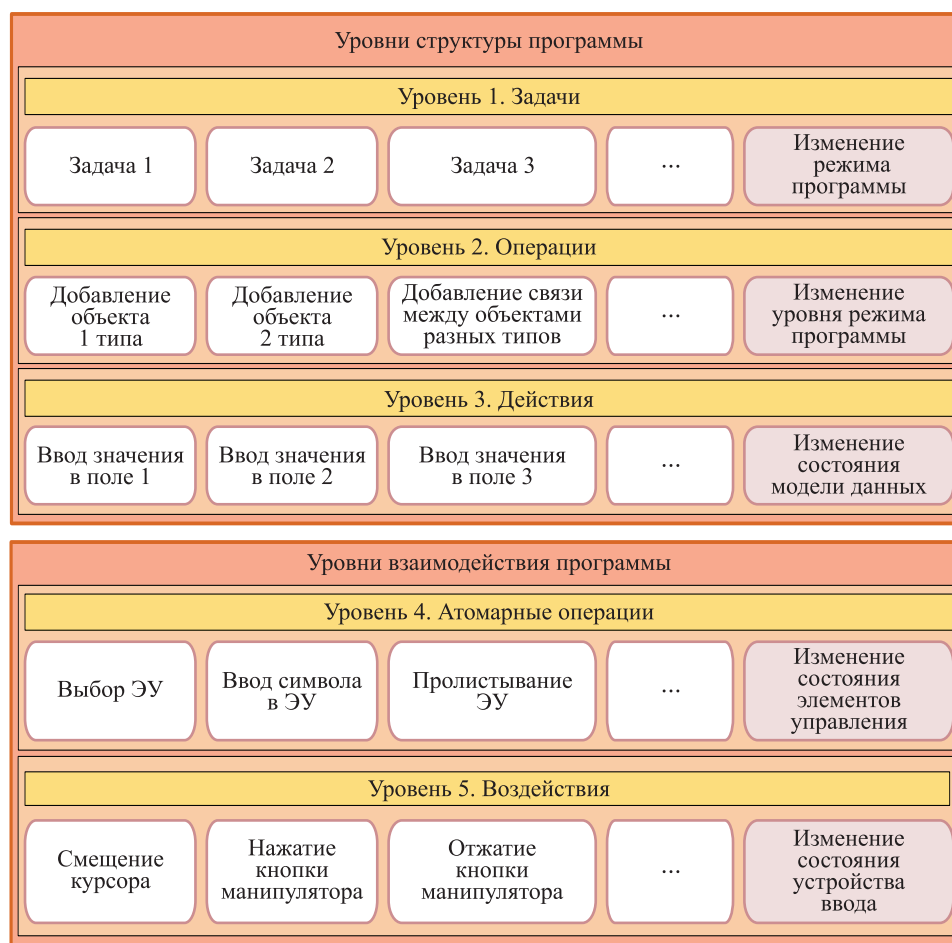


Рис. 2. Единая структура сведений о предметной области:
ЭУ — элементы управления

Разработка программного продукта в соответствии с подходом подразумевает интеграцию в программный код технологии, позволяющей осуществлять мониторинг действий оператора (блок 1 на рис. 1) в процессе ввода данных — трудозатратного процесса при взаимодействии с программным средством, который требует от оператора концентрации и наличия навыков применения средств ввода (ввод данных характеризуется самой высокой активностью оператора).

Оператор вводит входные параметры с помощью технических средств ввода данных: клавиатуры, манипулятора «мышь», фотокамеры, микрофона, сканера, специализированных панелей кнопок и переключателей, внешних носителей информации, телеграфного аппарата и т. п. Поэтому время на ввод данных состоит из времени отклика средства ввода данных от действия оператора (например, нажатие на кнопку, ввод текстовой строки, формирование голосовой

команды) и времени преобразования полученного значения в объекты программы.

Таким образом, атомарные операции характеризуют единичное обращение (ввод) оператора к элементу управления пользовательского интерфейса с применением средств ввода данных.

Время отклика средства ввода (манипулятора) зависит от физических возможностей и навыков оператора работы на манипуляторе и технических характеристик манипулятора. Каждая атомарная операция основывается на низкоуровневых сигналах от средства ввода, которые программной средой за счет событийной обработки преобразуются в воздействия ввода оператора (например, такие как нажатие левой кнопки мыши, смещение курсора, отпускание клавиши Ctrl, перевод фокуса ввода между элементами управления, изменение текста в поле ввода, пролистывание значений в выпадающем списке и т. д.). Воздействия ввода обладают рядом параметров, среди которых координаты (XУ) курсора на рабочей области программы, состояния кнопок мыши и клавиатуры. Элементы управления пользовательского интерфейса, расположенные на рабочей области программы, как правило, прямоугольной формы, характеризуются параметрами: XУ координатами левого верхнего угла, шириной и высотой, а также набором других свойств (состояний), определяющих функциональное предназначение. Путем применения манипуляторов ввода в ходе выполнения атомарных операций с элементами управления интерфейса программное средство реализует функции в соответствии с его назначением.

Действия программы формируются из последовательностей атомарных операций. Например, ввод имени нового контакта в программе коммуникатора состоит из набора атомарных операций: выбора текстового поля ввода (установление фокуса ввода), ввода символов. Каждое действие программы можно разложить на атомарные операции, выполняемые оператором. Время выполнения операции состоит из суммарного времени выполнения атомарных операций.

Мониторинг действий оператора позволяет осуществлять сбор и накопление статистики деятельности оператора во время эксплуатации программных средств. Используемый в [13] методический аппарат основывается на обобщении сведений о взаимодействии пользователя с пользовательским интерфейсом, что позволяет исследовать и оценить качество интерфейса, а также найти пути повышения его качества.

Оценка качества пользовательского интерфейса обеспечивается за счет:

- применения типовой системы показателей качества (блок 2 на рис. 1);
- разработки методики оценки качества;
- выполнения мероприятий по оценке показателей качества.

Как было указано выше, мониторинг действий оператора позволяет осуществлять сбор сведений по всем воздействиям оператора, а также по всем событиям программы. Система показателей качества пользовательского интерфейса основана на сведениях следующих типов:

- аппаратные события;
- события приложения;
- пользовательские события.

Аппаратные события возникают от действия всех технических средств при работе с периферийным оборудованием, а также вследствие использования каналов связи и удаленных подключений.

События приложения связаны с операциями получения, обработки и преобразования данных, обеспечения политики безопасности и уровней доступа, т. е. все эти события появляются в процессе функционирования программы в программной среде.

Пользовательские события, предусмотренные программой, происходят в результате воздействий пользователя на элементы управления интерфейса.

Величины, с помощью которых можно определить качество выполнения операции:

- среднее время выполнения операции;
- число ошибок (количество итераций), возникающих за период работы;
- среднее время возникновения ошибки.

При оценках этих величин применяется регистрационный метод. Оценка качества пользовательского интерфейса программных средств осуществляется путем обобщения, агрегирования и анализа статистических данных деятельности оператора, подсчета количества выполнения каждого воздействия пользователем, суммирования временных интервалов выполнения атомарных операций.

Показатели качества пользовательского интерфейса выражаются в формате:

- время выполнения операций (действий);
- число действий, необходимых для выполнения операции;
- число атомарных операций, необходимых для выполнения действия;
- число пустых воздействий, выполненных оператором;
- число ошибочных атомарных операций.

Таким образом, при разработке типовой системы показателей качества пользовательского интерфейса в соответствии с единой структурой сведений о предметной области для каждой задачи, операции и действия задаются и оцениваются показатели в приведенном выше формате.

При задании требований для показателей значения выбираются из диапазонов возможных значений, предусмотренных к оценке для каждого пользовательского интерфейса.

Следующий элемент подхода «Автоматизация программирования и документирования пользовательского интерфейса» (блок 3 на рис. 1), во-первых, подразумевает возможность автоматизированного документирования интерфейса программы. Для определения места документирования интерфейса в процессе его создания необходимо условно отделить процесс создания пользовательского интерфейса от более общего процесса разработки программного средства [14]. Основные мероприятия жизненного цикла пользовательского интерфейса:

- задание требований;
- проектирование;
- разработка, программирование (создание программного кода);
- оценка качества и испытания;
- документирование (описание).

В работе [14] показано, что каждое мероприятие жизненного цикла интерфейса взаимосвязано с UML моделями, описывающими интерфейс, которые в ходе итерационного процесса разработки корректируются и используются для получения документов. В качестве выходной продукции в соответствии с ГОСТ 19.101–77 «Виды программ и программных документов» кроме исполняемых файлов программы, динамически подключаемых библиотек и других вспомогательных файлов получают программную и эксплуатационную документацию. Среди программных документов, перечисленных в ГОСТ 19.101–77, могут упоминаться сведения об интерфейсе в части:

- описания последовательности действий пользователя над элементами управления программы;
- приведения снимков экранов (скриншотов) окон различных режимов функционирования программы;
- описания различных видов воздействий, предусмотренных программой;
- описания типов элементов управления, применяемых в пользовательском интерфейсе программы;
- приведения структуры программы в виде перечня функций и подфункций программы;
- приведения наборов входных данных при выполнении функций программы.

Во-вторых, в соответствии с подходом предусматривается автоматизация программирования макетов пользовательского интерфейса, основанная на таком выборе паттерна проектирования программного кода, который позволит инкапсулировать механизмы обработки

данных и управления от элементов управления интерфейса, но при этом обеспечит связь с моделью данных и мониторинг действий оператора. Инициализация всех свойств элементов управления, отражающих их визуальные характеристики, при автоматизированном программировании должна происходить на лету, основываясь на некоторых установках по умолчанию, в том числе с возможностью ее настройки. Такой подход реализуем при наличии средства автоматизированного построения UML диаграмм, описывающих пользовательские интерфейсы, которое позволит интерпретировать диаграммы в программные объекты и наоборот.

В работе [9] предлагается UML модель сценариев применения пользовательского интерфейса (в дальнейшей работе она была переименована на модель сценариев воздействий пользователя (СВП)), описывающая различные стороны функционирования программы, которая выражает на определенном уровне абстракции порядок взаимодействия пользователя с программным средством. Модель используется для декомпозиции и формирования однозначного понимания сведений по совокупности функций и режимов работы разрабатываемого программного средства (декомпозиции и анализа существующего программного средства), а также для обеспечения возможности автоматизированного документирования и построения кода интерфейса программы [15].

С помощью UML модели СВП решаются следующие задачи:

- прототипирование и/или макетирование пользовательского интерфейса при разработке программного средства;
- формализация существующего пользовательского интерфейса;
- описание деятельности пользователя при эксплуатации программных средств, выраженное в виде набора осуществленных сценариев воздействий на элементы управления программы.

Отображение некоторого абстрактного сценария осуществляет механизм его интерпретации (блок 4 на рис. 1) в стандартные программные процедуры, характерные для выбранной программно-аппаратной платформы. Наличие интерпретатора предписывает применение некоторой формальной логики (языка), с помощью лексем которой выражаются любые сценарии. Одна лексема описывает типовую атомарную операцию над элементом управления пользовательского интерфейса, идентифицирующие сведения о которой содержатся в параметрах лексемы. Лексемы могут описывать как строгие сценарии, при которых воздействия приведены последовательно, так и нестрогие, при которых пользователь может выбрать воздействие из нескольких альтернатив.

Каждая атомарная операция, которая вызвана воздействиями пользователя, на диаграмме СВП представляет собой дугу графа,

вершины графа — элементы управления пользовательского интерфейса, веса графа — комплексные показатели, характеризующие:

- количество выполнений атомарной операции;
- время выполнения атомарной операции;
- количество ошибок, связанных с выполнением атомарной операции.

Перечисленные показатели атомарных операций зависят от уровня подготовки, способностей и физического состояния пользователя (параметры технических средств принимаются за постоянную величину). Если влияние пользователя на время выполнения атомарных операций выразить через набор параметров и принять в качестве постоянной величины, то варьирование параметров элементов управления, таких как размеры и координаты расположения, позволяет управлять временами выполнения атомарных операций. С учетом особенностей среды, в которой подразумевается функционирование программы, исследование статистики деятельности оператора позволяет обобщить для каждого интерфейса времена выполнения атомарных операций в виде диапазонов значений.

Применение подхода к описанию воздействий пользователя на элементы интерфейса программы через модель СВП позволяет структурировать подобным образом программный код. Использование единого подхода к формализации интерфейса и к его программированию создает благоприятные условия для автоматизации процесса создания программы. Каждая новая итерация генерирует новый интерфейс, подлежащий оценке качества за счет натуральных испытаний потенциальными пользователями (представителями эксплуатирующей организации), причем для оценки интерфейса необязательно выполнение всех реальных функций программы — достаточно использовать программные заглушки [16]. Одним из способов сократить количество вариантов макетов интерфейса — оценка интерфейса с помощью виртуального пользователя, который в соответствии с [17] постоянно повторяет свой пользовательский сценарий, пока тест не закончится. Применение виртуального пользователя обеспечивает возможность тестировать и отлаживать взаимодействие пользователя с программой, сокращающее время реальных действий.

Заключение. Предлагаемая перспективная технология автоматического программирования программных средств подготовки данных позволяет:

- использовать абстракцию сценариев воздействий пользователя, формализованную в виде UML моделей, при описании интерфейса;
- генерировать программный код набора макетов интерфейса, а также программную документацию (разделы программных документов);

- осуществлять сбор и хранение информации, раскрывающей последовательность действий пользователя, в целях получения обоснованных выводов по возникающим ошибкам, деструктивным и пустым действиям;
- проводить оценки качества пользовательского интерфейса с помощью регистрационного метода на основе статистических данных мониторинга действий оператора;
- выполнять тестирование и отладку взаимодействия пользователя с программой посредством применения «виртуального пользователя».

ЛИТЕРАТУРА

- [1] Поликарпова Н.И., Шальто А.А. *Автоматное программирование*. Санкт-Петербург, СПбГПУ, 2008, 227 с.
- [2] Карпов В.Э. *Автоматное программирование и робототехника*. URL: <http://robofob.ru/materials/articles/pages/avprog.pdf> (дата обращения 20.09.2020).
- [3] Шальто А.А. Технология автоматного программирования. *Мир ПК*, 2003, № 10, с. 74–78.
- [4] Вольфсон Б.И. *Гибкие методологии разработки*. Санкт-Петербург, Питер, 2017, 144 с.
- [5] Кон М. *Scrum: гибкая разработка ПО*. Москва, Вильямс, 2016, 576 с.
- [6] Карпов Д.В. Гибкая методология разработки программного обеспечения. *Вестник Нижегородского университета им. Н.И. Лобачевского*, 2011, № 3, с. 227–230.
- [7] *Гибкая методология разработки программного обеспечения*. Москва, НОУ Интуит, 2016, 154 с.
- [8] Стеллман Э., Грин Д. *Постигая Agile. Ценности, принципы, методологии*. Москва, Манн, Иванов и Фербер, 2017, 446 с.
- [9] Казаков Г.В., Чемирисов В.В. Применение UML-модели для построения пользовательского интерфейса программных средств подготовки данных управления летательными аппаратами. *Ракетные комплексы и ракетно-космические системы. Проектирование, экспериментальная отработка, летные испытания, эксплуатация: Труды секции 22 имени академика В.Н. Челомея XLII Академических чтений по космонавтике. Вып. 6*. Л.С. Точилов, сост.; А.Г. Леонов, ред. Реутов, ВПК «НПО машиностроения», 2018, 582 с.
- [10] Тиханычев О.В. Пользовательские интерфейсы в автоматизированных системах: проблемы разработки. *Программные системы и вычислительные методы*, 2019, № 2, с. 11–22.
- [11] Левин Д.П., Люшнин С.А. База данных «Оружие нелетального действия» как инструмент прогнозирования рисков развития ОНД-технологий. *Инженерный журнал: наука и инновации*, 2015, вып. 3. DOI: 10.18698/2308-6033-2015-3-1378
- [12] Федотова А.В., Давыденко И.Т. Применение семантических технологий для проектирования интеллектуальных систем управления жизненным циклом продукции. *Известия высших учебных заведений. Машиностроение*, 2016, № 3, с. 74–81. DOI: 10.18698/0536-1044-2016-3-74-81
- [13] Раскин Дж. *Интерфейс: новые направления в проектировании компьютерных систем*. Москва, Символ-Плюс, 2005, 272 с.

- [14] Казаков Г.В., Чемирисов В.В. Анализ средств построения графического пользовательского интерфейса как неотъемлемой части программных средств подготовки данных. *Ракетные комплексы и ракетно-космические системы. Проектирование, экспериментальная отработка, летные испытания, эксплуатация: Труды секции 22 имени академика В.Н. Челомея ХLI Академических чтений по космонавтике. Вып. 5.* Л.С. Точилев, сост.; А.Г. Леонов, ред. Реутов, ВПК «НПО машиностроения», 2017, 556 с.
- [15] Ларман К. *Применение UML 2.0 и шаблонов проектирования. Практическое руководство.* 3-е изд. Москва, И.Д. Вильямс, 2013, 736 с.
- [16] Моск-объект и dummy-(функция заглушка) использование в тестировании. URL: <https://intellect.icu/mock-obekt-i-dummy-funtsiya-zaglushka-ispolzovanie-v-testirovanii-6187> (дата обращения 20.09.2020).
- [17] Лыфенко Н.Д. Виртуальные пользователи в социальных сетях: мифы и реальность. *Вопросы кибербезопасности*, 2014, № 5, с. 17–20.

Статья поступила в редакцию 27.10.2020

Ссылку на эту статью просим оформлять следующим образом:

Казаков Г.В., Корянов В.В., Чемирисов В.В., Уваров В.В. Методический подход к созданию универсального пользовательского интерфейса. *Инженерный журнал: наука и инновации*, 2020, вып. 11.

<http://dx.doi.org/10.18698/2308-6033-2020-11-2034>

Казаков Геннадий Викторович — канд. техн. наук, доцент, начальник управления ФГБУ «4 ЦНИИ» Минобороны России, почетный работник науки и техники Российской Федерации. e-mail: kgv.64@mail.ru (SPIN-code: 8553-9753)

Корянов Всеволод Владимирович — канд. техн. наук, доцент, первый заместитель заведующего кафедрой «Динамика и управление полетом ракет и космических аппаратов» МГТУ им. Н.Э. Баумана. e-mail: vkoryanov@bmstu.ru

Чемирисов Владимир Вячеславович — научный сотрудник ФГБУ «4 ЦНИИ» Минобороны России. e-mail: vladimir.chemir@yandex.ru

Уваров Александр Викторович — канд. техн. наук, старший научный сотрудник, старший научный сотрудник ФГБУ «4 ЦНИИ» Минобороны России. e-mail: vladimir.chemir@yandex.ru

Methodical approach to creating a universal user interface

© G.V. Kazakov¹, V.V. Koryanov², V.V. Chemirisov¹, A.V. Uvarov¹

¹Federal State Budgetary Institution 4th Central Research Institute of the Ministry of Defense of the Russian Federation,
Korolyov town, Moscow region, 141091, Russia

²Bauman Moscow State Technical University, Moscow, 105005, Russia

In the course of research in scientific organizations of the Ministry of Defense of the Russian Federation, the priority is given to the tasks related to the military sphere, and the obtained results, i.e. technologies, methodological support, models, etc., are intended for military specialists. In such research, the accompanying scientific results are technologies, methods and models which have a civilian focus and can be used by industrial enterprises in the course of military development. This paper describes a promising technology for automatic programming of data preparation software. In computer science, the term “automatic programming” identifies a type of computer programming in which some mechanism generates a computer program to allow a human programmer to write code at a higher level of abstraction. As a new abstraction, which is supposed to be applied in programming within the framework of the proposed technology, “the scenario of user actions on the user interface controls” is used. The information required to build a software architecture based on scenarios of user actions is: a list of program functions and the structure describing these data, types of controls, types of input manipulators and types of program events generated by control elements as a result of input manipulators.

Keywords: architecture, data input, quality, modeling, operator, software, design, user interface, development

REFERENCES

- [1] Polycarpova N.I., Shalyto A.A. *Avtomatnoe programmirovaniye* [Automatic programming]. St. Petersburg, SPBSTU, 2008, 227 p.
- [2] [Karpov V.E. *Avtomatnoe programmirovaniye i robototekhnika* [Automatic programming and robotics]. Available at: <http://robobob.ru/materials/articles/pages/avprog.pdf>
- [3] Shalyto A.A. *Mir PK (PC World)*, 2003, no. 10, pp. 74–78.
- [4] Volfson B.I. *Gibkie metodologii razrabotki* [Flexible development methodologies]. St. Petersburg, Piter Publ., 2017, 144 p.
- [5] Cohn M. *Scrum: gibkaya razrabotka PO* [Scrum: flexible development of software]. Moscow, Williams Publ., 2016, 576 p. (In Russ.).
- [6] Karpov D.V. *Vestnik Nizhegorodskogo universiteta im. N.I.Lobachevskogo — Vestnik of Lobachevsky University of Nizhni Novgorod*, 2011, no. 3, pp. 227–230.
- [7] *Gibkaya metodologiya razrabotki programmnogo obespecheniya* [Flexible software development methodology]. Moscow, NOI Intuit Publ., 2016, 154 p.
- [8] Stellman A., Greene J. *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. O'Reilly Media. 1st ed. 2013, 420 p. [In Russ.: Stellman A., Greene J. *Postigaya Agile. Tsennosti, printsipy, metodologii*. Mann-Ivanov-Ferber Publ., 2017, 446 p.].
- [9] Kazakov G.V., Chemirisov V.V. *Primeneniye UML-modeli dlya postroeniya polzovatelskogo interfeysa programnykh sredstv podgotovki dannykh upravleniya letatelnyimi apparatami* [Application of UML-model for construction of user interface of software tools of aircraft control data preparation]. *Sbornik trudov sektsii 22 imeni akademika V.N.Chelomeya XLII Akademicheskikh chteniy po kos-*

- monavtike* [Proceedings of Section 22 named after Academician V.N. Chelomey of the XLII Academic Readings on Cosmonautics]. No. 6, 2018, pp. 196–206.
- [10] Tikhanychev O.V. *Programmnye sistemy i vychislitelnye metody — Software Systems and Computational Methods*, 2019, no. 2, pp.11–22.
- [11] Levin D.P., Lyushnin S.A. *Inzhenerny zhurnal: nauka i innovatsii — Engineering Journal: Science and Innovation*, 2015, iss. 3.
DOI: 10.18698/2308-6033-2015-3-1378
- [12] Fedotova A.V., Davydenko I.T. *Izvestiya vysshikh uchebnykh zavedeniy. Mashinostroenie — BMSTU Journal of Mechanical Engineering*, 2016, no. 3, pp. 74–81.
- [13] Raskin J. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley Professional, 2000, 233 p. [In Russ.: Raskin J. *Interfeys: novye napravleniya v proektirovaniy kompyuternykh sistem*. Moscow, Simvol-Plyus Publ., 2005, 272 p.].
- [14] Kazakov G.V., Chemirisov V.V. Analiz sredstv postroeniya graficheskogo polzovatel'skogo interfeysa, kak neotemlemoy chasti programmnykh sredstv podgotovki dannykh [Analysis of tools for constructing a graphical user interface as an integral part of data preparation software]. *Sbornik trudov sekti 22 imeni akademika V.N.Chelomeya XLI Akademicheskikh chteniy po kosmonavtike* [Proceedings of Section 22 named after Academician V.N. Chelomey XLI of Academic Readings on Cosmonautics]. No. 5, 2017, pp. 429–440.
- [15] Larman C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3rd ed. Pearson, 2004, 736 p. [In Russ.: Larman C. *Primenenie UML 2.0 i shablonov proektirovaniya. Prakticheskoe rukovodstvo*. 3rd ed. Moscow, JSC I.D.Vilyams Publ., 2013, 736 p.].
- [16] *Mock-obekt i dummy-(funktsiya zaglushka) ispolzovanie v testirovanii* [Mock object and dummy-(stub function) use in testing]. Available at: <https://intellect.icu/mock-obekt-i-dummy-funksiya-zaglushka-ispolzovanie-v-testirovanii-6187>
- [17] Lyfenko N.D. *Voprosy kiberbezopasnosti — Cybersecurity issues*, 2014, no. 5, pp. 17–20.

Kazakov G.V. (b. 1964), Cand. Sc. (Eng.), Assoc. Professor, Head of the Federal State Budgetary Institution 4th Central Research Institute of the Ministry of Defense of the Russian Federation, honorary worker of science and technology of the Russian Federation. Author of over 70 works in the field of automated control system reliability. e-mail: kgv.64@mail.ru (SPIN-code: 8553-9753)

Koryanov V.V. (b. 1982) graduated from Bauman Moscow State Technical University in 2006. Cand. Sc. (Eng.), Assoc. Professor, First Deputy Head of the Department of Dynamics and Flight Control of Rockets and Spacecraft, Bauman Moscow State Technical University. Author of over 40 works in the field of ballistics modelling and dynamics of spacecraft and descent vehicle motion. e-mail: vkoryanov@bmstu.ru

Chemirisov V.V. (b. 1984), Research Fellow, Federal State Budgetary Institution 4th Central Research Institute of the Ministry of Defense of the Russian Federation. Author of over 10 works in the field of software development of automated control systems. e-mail: vladimir.chemir@yandex.ru

Uvarov A.V. (b. 1937), Cand. Sc. (Eng.), Senior Research Fellow, Federal State Budgetary Institution 4th Central Research Institute of the Ministry of Defense of the Russian Federation. Author of over 150 works in the field of ballistics modelling and automated control systems. e-mail: vladimir.chemir@yandex.ru