

УДК 004.413:338.5

# ПОВТОРНО ИСПОЛЬЗУЕМОЕ РЕШЕНИЕ ДЛЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПУЛЬТА ИНСТРУКТОРА АВИАЦИОННОГО ТРЕНАЖЕРА

**В.А. Хоменко**

Кандидат технических наук, заместитель заведующего  
отделом  
Институт программных систем Национальной академии  
наук Украины  
пр. Глушкова, 40, г. Киев, Украина, 03187  
Контактный тел.: (044) 526-63-21  
E-mail: vlkhomenko@isofts.kiev.ua

*Пропонується отримане шляхом доменного аналізу повторно використовуване рішення для розробки програмного забезпечення пульта інструктору авіаційного тренажера*

*Ключові слова: авіаційний тренажер, програмне забезпечення, доменний аналіз*

---

*Предлагается полученное путем доменного анализа повторно используемое решение для разработки программного обеспечения пульта инструктора авиационного тренажера*

*Ключевые слова: авиационный тренажер, программное обеспечение, доменный анализ*

---

*The domain analysis reusable solution for the development of the air simulator teacher console software is proposed*

*Key words: air simulator, software, domain analysis*

## 1. Введение

Комплексный авиационный тренажер представляет собой информационно-моделирующую систему, имитирующую самолет и предназначенную для подготовки пилотов [1, 2]. Один из неотъемлемых компонентов авиационного тренажера – пульт инструктора обучения пилотов. Современные pulpты инструктора реализуются как аппаратно-программные системы на основе компьютеров, включающие средства визуализации информации, интерактивного взаимодействия с пользователем и информационного обмена с другими компонентами тренажера (рис.1). Программное обеспечение пультов создается на основе современных технологий с применением операционных систем, системного программного обеспечения и передовых средств разработки [3, 4].

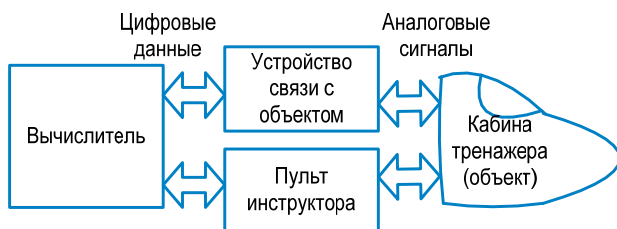


Рис. 1. Основные компоненты авиационного тренажера

## 2. Постановки задачи

Авиационные тренажеры с точки зрения разработчиков программного обеспечения являются хорошо определенным прикладным доменом, для которого известны типичные объекты и их поведение, что позволяет использовать для разработки программного обеспечения доменную инженерию [5]. Первой фазой доменной инженерии является доменный анализ, направленный на идентификацию доменов, выявление их границ, исследование общности и различий систем.

В статье предлагаются результаты доменного анализа, решения и архитектуры средств для разработки программного обеспечения пультов инструктора авиационных тренажеров путем доменной инженерии. Приводится пример реализации доменной инженерии для создания пульта инструктора авиационного тренажера TL-410 Национального авиационного университета.

## 3. Доменный анализ

Контекст доменного анализа пульта инструктора (рис. 2) очерчивает его границы и определяет окружение в виде внешних входов и выходов (результатов).

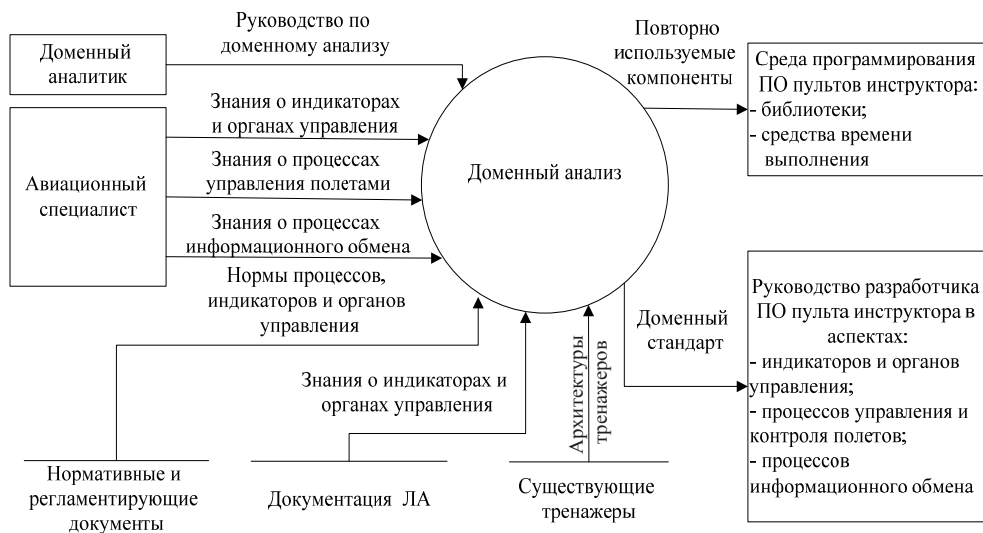


Рис. 2. Контекст доменного анализа

Выполнение доменного анализа включает ряд последовательных процессов – подготовка информации домена, анализ домена, создание повторно используемых рабочих продуктов [5]. Доменный анализ для пульта инструктора проводится в два этапа – общий анализ домена и анализ домена по трем отдельным аспектам: интерфейсные элементы; процессы управления и контроля обучения; информационный обмен.

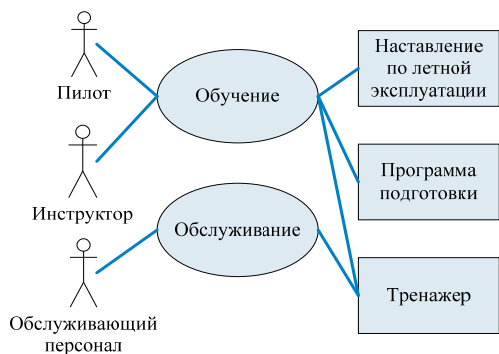


Рис. 3. Использование тренажера

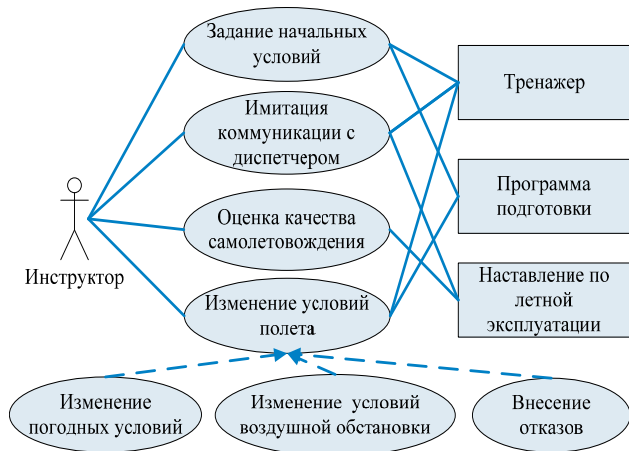


Рис. 4. Функции инструктора

Обучение пилотов производится в соответствии с программами подготовки (рис. 3), которые формируются в зависимости от конкретных задач обучения, и предусматривают проведение серии учебных полетов в заданных условиях под контролем инструктора (рис. 4). Для оценки качества самолетовождения и разбора полетов инструктор получает значения необходимых параметров полета с их фиксацией на бумажном или электронном носителе.

В авиационном прикладном домене

большинство видов информации, циркулирующей в организационно-технических системах, известно и описано, также как и традиционные формы ее представления для авиационного персонала. Для каждого специалиста определены типовые должностные функциональные обязанности, требования к знаниям и квалификации, то есть его роль.

#### 4. Интерфейсные элементы

Пользовательский интерфейс инструктора составляют интерфейсные элементы, которые разделяют на управляющие, предназначенные для ввода информации (переключатели режимов, регуляторы, кнопки ввода значений) и индикаторы, предназначенные для отображения информации (индикаторы скорости, высоты, изображения карт). В свою очередь, управляющие элементы и индикаторы можно классифицировать по способу ввода и отображения информации. Например, индикаторы по форме отображения можно разделить на шкалы-стрелки (высотомер, индикатор тяги двигателя, курсовой индикатор и т.п.), транспаранты (флаг «Минимальный запас топлива», флаг «Пожар», индикатор пролета маркерного маяка), визиры (курсогладисадный указатель, авиагоризонт), цифровые табло (частота радиосвязи, координаты).

Органы управления можно разделить на аналоговые (рукоятки управления тягой, регуляторы громкости и т.п.), дискретные (переключатели) и цифровые (клавиатуры).

Каждый индикатор отображает определенный параметр полета. Множество индикаторов, доступных инструктору составляет реализацию его ролевого представления и должно определяться подмножеством всех параметров полета, и способами их представления, принятыми в авиации, с учетом конкретного типа самолета. Органы управления пульта определяются потребностями управления тренажером. Каждый интерфейсный элемент  $ei = (it, pf)$  определяется двумя составляющими – видом информации (параметром)  $it$  и формой ее представления  $pf$ . Ролевое пред-

ставление  $rv$  можно описать как множество входящих в него интерфейсных элементов  $RV = \{ei \mid ei \in RV\}$ , а роль  $R$  в свою очередь – как множество панелей представления  $R = \{RV \mid RV \in R\}$ . Все множество интерфейсных элементов домена  $EI = \{ei_i \mid ei_i \in EI, i=1..d\}$  должно быть определено как множество двоек  $ei = (it, pf)$  всех его подмножеств  $EI_r$ , определенных для ролей:  $P(EI) = \{EI_r \mid EI_r \subseteq EI\}$ .

Интерфейс для роли инструктора синтезируется путем выбора и настройки определенных интерфейсных элементов:  $r_{instr} = RV_{instr} \subseteq \{ei_i = (it, pf) \mid i=1..l\}$ .

Каркасы ИЭ представлены в виде абстрактных классов, находящихся в иерархии с базовым абстрактным классом «Интерфейсный элемент» [6] (рис. 5). Класс *GadgetBase* является базовым для всех интерфейсных элементов и реализует интерфейсы *IGadget* для взаимодействия с моделями тренажера, а *IParametersCollection* – для универсальной работы с коллекциями имен и значений параметров. Далее цепочка наследования делится на две ветки – базовые классы индикаторов (*InfoGadgetBase*) и органов управления (*ControlGadgetBase*).

**1. Процессы управления учебными полетами**

Задание и изменение условий полета должны осуществляться в соответствии со сценарием – программой полета, разработанной для достижения определенных задач подготовки (штатный взлет и посадка, парирование отказов, пилотирование в сложных метеословиях и т.п.). Изменение условий полета производится при наступлении событий (выполнении самолетом определенного маневра, достижении заданной высоты, времени полета и т.п.), которые могут идентифицироваться по совокупности значений заданных параметров.

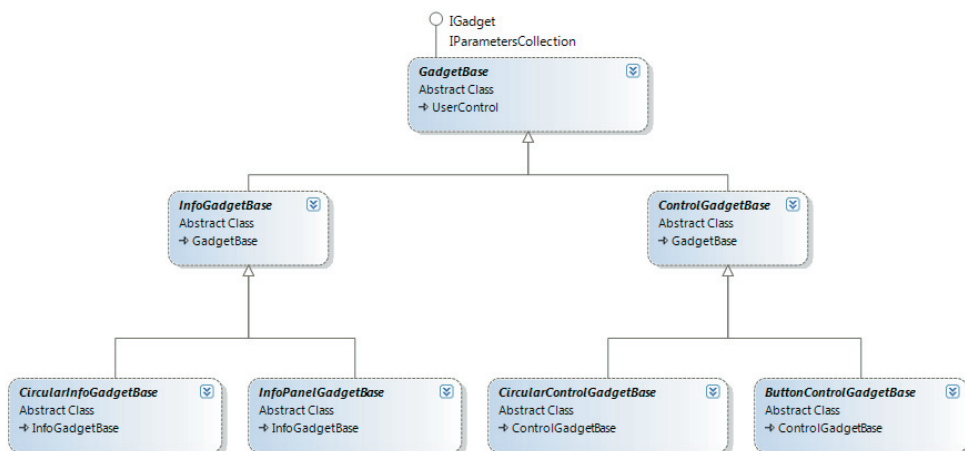


Рис. 5. Абстрактные классы интерфейсных элементов

Контроль полета включает получение инструктором параметров полета, необходимых для оперативно-го управления, и их записи для последующего анализа действий пилотов.

Стандартное решение пульта в аспекте управления и контроля полетов должно включать шаблоны программных компонентов для установки начальных

условий, изменений условий полета, отображения и записи предопределенных параметров полета.

**2. Информационный обмен с другими компонентами тренажера**

Современный пульт инструктора тренажера функционирует как часть распределенного информационно-моделирующего комплекса с обменом через вычислительную сеть. Обмен пульта на прикладном уровне должен быть специфицирован с учетом того, что параметры, которыми пульт обменивается с другими узлами вычислительной системы, могут значительно различаться по составу в зависимости от типа и режима работы тренажера и составлять относительно большое количество (сотни и тысячи), кроме того, тренажер является системой реального времени. При разработке шаблона компонента обмена это обуславливает требования высокой масштабируемости обмена, простоты и гибкости включения новых параметров, возможности передачи набора необходимых параметров несколькими относительно небольшими пакетами.

Для реализации первого требования предлагается использовать в обмене на прикладном уровне XML-пакеты, определенного формата [7] (рис. 6), который обеспечивает передачу-прием пакетов с произвольным набором и числом параметров, а также с произвольной длиной обозначений и значений.



Рис. 6. Формат XML-пакета обмена

Для реализации второго требования предлагается использовать специальный запрос-ответный прикладной протокол обмена с выделением ведущего узла, который обеспечивает синхронизацию обмена и ограничение длины пакета, передаваемого в один цикл вычисления тренажера. Основными составляющими протокола являются схемы обмена и типы используемых пакетов. Протокол включает три возможные схемы обмена между узлами (рис. 7).

Первая схема (рис. 7а) предназначена для периодической передачи информации от ведущего узла к ведомым (пакет типа «инфо»). Вторая схема (рис. 7б) предназначена для передачи информации от ведомых узлов к ведущему при наличии готовности к приему. Третья схема (рис. 7в) предназначена для передачи данных от ведущего узла к ведомому по запросу последнего. Каждый пакет обмена состоит из трех блоков: заголовка типа пакета; длина блока данных; данные в XML-пакете обмена.

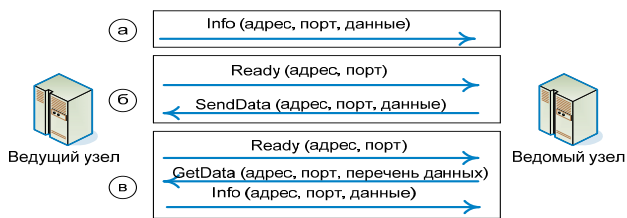


Рис. 7. Схемы обмена

### 3. Архитектура программного обеспечения

Компоненты шаблона программного обеспечения объединяются архитектурой, которая показана на рис. 8.

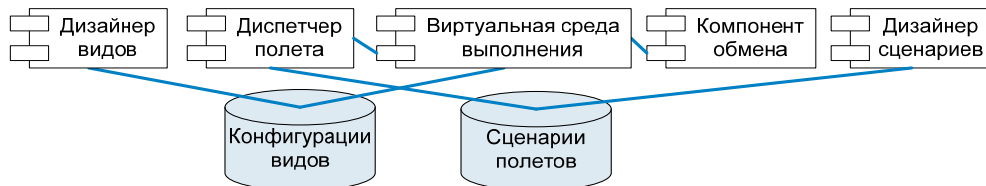


Рис. 8. Архитектура средств для реализации пульта инструктора

Дизайнер сценариев предназначен для описания сценариев полетов в рамках программы подготовки. Выполнение сценария во время полета обеспечивается специальным компонентом времени выполнения - диспетчером полета.

Дизайнер представлений предназначен для создания конфигураций представления инструктора, который может создавать и редактировать конфигурации панелей пульта с использованием коллекций интерфейсных элементов. Для реализации представления, специфицированного в конфигурационном файле, используется виртуальная среда выполнения.

Компонент обмена обеспечивает обмен пульта с другими подсистемами тренажера с помощью прикладного протокола обмена, функционирующего поверх сетевого протокола.

### 4. Пример создания программного обеспечения пульта инструктора

Результаты доменного анализа были применены во время реинженерии авиационного тренажера TL410, осуществляемой в киевском Национальном авиационном университете [1, 7]. В этом проекте на сегодняшний день реализованы два аспекта абстрактного

пульта: индикаторы и органы управления; информационный обмен с другими компонентами тренажера. На основе шаблонов реализован программный комплекс «Рабочее место инструктора», состоящий из двух приложений: визуального дизайнера представления и виртуальной среды выполнения.

Визуальный дизайнер представления позволяет создавать конфигурации и в визуальном режиме настраивать свойства панелей (размер, местоположение), сохраняемые в XML-файл. Каждый элемент представления реализуется в виде отдельного программного модуля (Net-сборки), который подключается динамически, что

позволяет создавать виртуальное место из отдельных элементов.

В задачи виртуальной среды выполнения входит отображение определенных панелей и интерфейсных элементов, а также обеспечение обмен информа-

цией и взаимодействие с моделями тренажера.

Взаимодействие между виртуальной средой выполнения и модулями интерфейсных элементов реализовано на основе шаблона проектирования Observer, который позволяет реализовать подписку множества клиентов (модулей элементов управления) на получение информации от одного источника событий (модель) и рассылку им уведомлений.

### Выводы

Разработка повторно используемых решений – это действенный способ снижения затрат на реализацию типовых проектов. Состояние отрасли авиационных тренажеров позволяет создавать высокоуровневые, многократно востребованные решения программно-обеспечения. Предлагаемый абстрактный пульт инструктора тренажера – пример одного из таких решений, основанный на анализе домена и современных технологиях разработки программного обеспечения и апробированный при реинженерии авиационного тренажера.

### Литература

1. Сидоров, Н.А. Реинженерия программного обеспечения информационно-моделирующих тренажерных комплексов [Текст] / В.Т. Недоводеев, В.А. Хоменко, И.П. Сердюк, Е.Н. Сидоров // Управляющие системы и машины. – 2008. – №4. – С.68-74.
2. Моржов, В.І. Призначення і класифікація авіаційних тренажерів та моделювальних комплексів [Текст] / В.І.Моржов. – К.: КМУЦА, 1997 р. – 74 с.
3. Design of a flight simulator software architecture. Goran Ancker, Jan Wallenberg. – School of Mathematics and Systems Engineering, Vaxjo University. – 2002. – 91 p.
4. Сидоров, М.О. Методологічні принципи реінженерії програмного забезпечення успадкованих авіаційних тренажерів [Текст] / Л.М.Иванова, В.А. Хоменко // „Авіа-2007” – 2007. – Т.1, С. 13.119–13.122.
5. Prietto-Diaz R. Domain Analysis: An Introduction.-Software engineering Notes. - 1990. - V. 15, № 2. - p. 47-54.
6. Оленников В.Ю. Применение шаблонов проектирования для создания программного обеспечения пульта инструктора авиационного тренажера: Матеріали Всеукраїнської конф. аспірантів і студентів «Інженерія програмного забезпечення 2008». – К.: НАУ – 2009. – с. 62.
7. Хоменко, В.А. Шаблон программного обеспечения устройств связи с объектом авиационных тренажеров [Текст] / Е.Н.Сидоров, И.Б.Мендзевровский // Проблемы программирования. - 2008. – №3- С. 30-40.