

ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
№2(10) 2012

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
МОЛОДІ ТА СПОРТУ УКРАЇНИ**

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

**ІНЖЕНЕРІЯ
ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ**

№ 2(10) 2012

НАУКОВИЙ ЖУРНАЛ

SOFTWARE ENGINEERING

КИЇВ

Журнал публікує оригінальні та оглядові статті науково-технічного характеру, звіти про наради, конференції, рецензії на монографії, матеріали проблемного та дискусійного характеру. У журналі публікуються статті, що охоплюють такі розділи:

- Теоретичні основи інженерії програмного забезпечення
- Реверсивна інженерія та реінженерія програмного забезпечення
- Емпірична інженерія програмного забезпечення
- Супровід і еволюція програмного забезпечення
- Тестування, валідація та верифікація програмного забезпечення
- Оцінка витрат та вартості програмного забезпечення
- Якість програмного забезпечення
- Удосконалення процесів життєвого циклу програмного забезпечення
- Технології розробки та супроводження програмного забезпечення
- Управління проектами
- Колективна розробка програмного забезпечення
- Доменний аналіз
- Екологія програмного забезпечення
- Освіта та інженерія програмного забезпечення
- Стандарти та інженерія програмного забезпечення
- Захист та інженерія програмного забезпечення
- Internet та інженерія програмного забезпечення
- Бази даних, бази знань та інженерія програмного забезпечення
- Прикладні домени і прикладне програмне забезпечення

РЕДАКЦІЙНА РАДА

М.О.Сидоров	доктор техн. наук, професор – голова
П.І. Андон	академік НАН України, доктор фіз.-мат. наук, професор
А.В.Анісімов	чл.-кор. НАН України, доктор фіз.-мат. наук, професор
Є.О.Башков	доктор техн. наук, професор
В.М.Буйвол	доктор фіз.-мат. наук, професор
А.М.Воронін	доктор техн. наук, професор
М.М.Глазунов	доктор фіз.-мат. наук, професор
В.П.Денисюк	доктор фіз.-мат. наук, професор
А.Ю.Дорошенко	доктор фіз.-мат. наук, професор
Ю.К.Зіатдінов	доктор техн. наук, професор
К.М.Лаврищева	доктор фіз.-мат. наук, професор
Р.Х.Садихов	доктор техн. наук, професор
Б.Г.Трусов	доктор техн. наук, професор
І.Б.Туркін	доктор техн. наук, професор
Д.В.Федасюк	доктор техн. наук, професор

АДРЕСА РЕДАКЦІЇ: 03058, м. Київ,
пр.-т Космонавта Комарова 1, Національний
авіаційний університет, факультет комп'ютерних
наук, корп. 6, к. 108.
Телефон: 044-406-73-96
E-mail: se_journal@livenau.net
<http://portal.csfнау.kiev.ua/journal/>

A journal publishes the original and survey articles of scientific and technical character, reports on conferences, reviews on monographs, materials of problem and debatable character. The articles are published in the following sections:

- Theoretical bases of software engineering
- Reverse software engineering
- Software reengineering
- Empirical software engineering
- Software maintenance and software evolution
- Software testing, validation and verification
- Software cost estimation
- Software quality
- Processes improvement of software life cycle
- Software and development technologies
- Project management
- Collaborative software development
- Domain analysis
- Software ecology
- Education and software engineering
- Standards and software engineering
- Defence and software engineering
- Internet and software engineering
- Databases and software engineering
- Applied domains and application software

EDITORIAL BOARD

N.A.Sidorov	Sc. D, Professor – editor-in-chief
P.I. Andon	Academician of NAS of Ukraine, Sc. D, Professor
A.V.Anisimov	Corresponding Member of the AS of Ukraine, Sc. D, Professor
E.O.Bashkov	Sc. D, Professor
V.N.Buyvol	Sc. D, Professor
A.N.Voronin	Sc. D, Professor
M.M.Glazunov	Sc. D, Professor
V.P.Denisyuk	Sc. D, Professor
A.Yu.Doroshenko	Sc. D, Professor
Yu.K.Ziatdinov	Sc. D, Professor
K.M.Lavrishcheva	Sc. D, Professor
R.H.Sadykhov	Sc. D, Professor
B.G.Trusov	Sc. D, Professor
I.B.Turkin	Sc. D, Professor
D.V.Fedasyuk	Sc. D, Professor

ADDRESS EDITORIAL OFFICE: 03058, Kyiv,
1 Cosmonaut Komarova Prospect, National aviation
university, faculty of computer sciences, of. 6.108.

Telephone: 044-406-73-96
E-mail: se_journal@livenau.net
<http://portal.csfнау.kiev.ua/journal/>

Національний авіаційний університет

**ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
№ 2 (10) 2012**

НАУКОВИЙ ЖУРНАЛ ЗАСНОВАНИЙ У КВІТНІ 2009 РОКУ
Свідоцтво про реєстрацію KB № 15075-3647P від 15.04.2009
Рекомендовано до друку вченою радою Національного авіаційного університету
Протокол № 8 від 24.10.2012 року
ПРЕДСТАВЛЕНИЙ В ІНТЕРНЕТІ <http://portal.csfnau.kiev.ua/Pages/Default.aspx>

ЗМІСТ

РЕВЕРСИВНА ІНЖЕНЕРІЯ ТА РЕІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ <i>В.А.Хоменко, Е.М.Сидоров</i> Повторно используемое решение для программного обеспечения пульта инструктора авиационного тренажера.....	5
ІНТЕРНЕТ ТА ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ <i>Пугач Я.Ю.</i> Разработка технологии для иформационного обеспечения клиентов в сетях USENET.....	12
БАЗИ ДАНИХ, БАЗИ ЗНАНЬ ТА ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ <i>Азарян А.А., Азарян В.А., Гриценко А.Н., Кайгородов Р.А., Мирошник Д.Ю.</i> Информационное обеспечение автоматизированной системы контроля качества при добыче железородного сырья в условиях карьеров	18
<i>Лисак Володимир Васильович</i> Інженерія аналітичних знань.....	27
ПРИКЛАДНІ ДОМЕНИ І ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ <i>Мучник М.М., Степанов К.Л.</i> Предпроцессорная обработка изображений.....	34



Мета конференції

Популяризувати та активізувати серед аспірантів і студентів наукові дослідження з інженерії програмного забезпечення. Ознайомити учасників конференції з науковою діяльністю університетів, Національної академії наук в галузі інженерії програмного забезпечення та (на прикладі технологій Microsoft) дати можливість студентам і аспірантам набути відповідних вмінь та навиків

Основні напрями роботи конференції

- Теоретичні основи інженерії програмного забезпечення
- Еволюція і супровід програмного забезпечення
- Реверсивна інженерія та реінженерія програмного забезпечення
- Колективна розробка програмного забезпечення
- Тестування, валідація та верифікація програмного забезпечення
- Оцінка витрат та вартості програмного забезпечення
- Якість програмного забезпечення
- Удосконалення процесів життєвого циклу

- Технології розробки програмного забезпечення
- UML
- Проектний менеджмент
- Освіта та інженерія програмного забезпечення
- Стандарти та інженерія програмного забезпечення
- Захист та інженерія програмного забезпечення
- Internet та інженерія програмного забезпечення
- Бази даних, бази знань та інженерія програмного забезпечення.
- Прикладні домени і програмне забезпечення

Детальну інформацію можна отримати на сайті за адресою:

<http://www.se2012.org.ua>
conf@se2012.org.ua

РЕВЕРСИВНА ІНЖЕНЕРІЯ ТА РЕІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.413:338.5

В.А.Хоменко, Е.М.Сидоров,
Национальный авиационный университет

ПОВТОРНО ИСПОЛЬЗУЕМОЕ РЕШЕНИЕ ДЛЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПУЛЬТА ИНСТРУКТОРА АВИАЦИОННОГО ТРЕНАЖЕРА

Для автоматизации процессов разработки программного обеспечения пульты инструктора авиационного тренажера предлагается использовать повторно, которое строится на основе доменного анализа. Решение проверяется на примере тренажера TLAI/PM.

Для автоматизації процесів розробки програмного забезпечення пульта інструктора авіаційного тренажера пропонується повторно використовувати рішення, яке будується на основі доменного аналізу. Рішення перевіряється на прикладі тренажера TLAI / PM.

To automate the processes of software development instructor console simulators offered a reusable solution that is based on a domain analysis. The solution is tested for example simulator TLAI / PM.

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

Введение

Комплексный авиационный тренажер представляет собой информационно-моделирующую систему, имитирующую самолет при подготовке пилотов [1, 2]. Имитируются входят органы управления, индикаторы, визуальное, звуковое окружение и подвижность кабины, поведение самолета. Основное назначение тренажера – обучение самолетовождению и повышение его качества путем выработки у пилотов правильных навыков пилотирования, навигации, коммуникации и парирования нештатных ситуаций. Обучение пилотов производится в соответствии с программами подготовки (рис.1), которые формируются в зависимости от конкретных задач обучения, и предусматривают проведение серии учебных полетов в заданных условиях. Программы подготовки могут изменяться в период эксплуатации самолета на основе приобретаемого опыта самолетовождения.

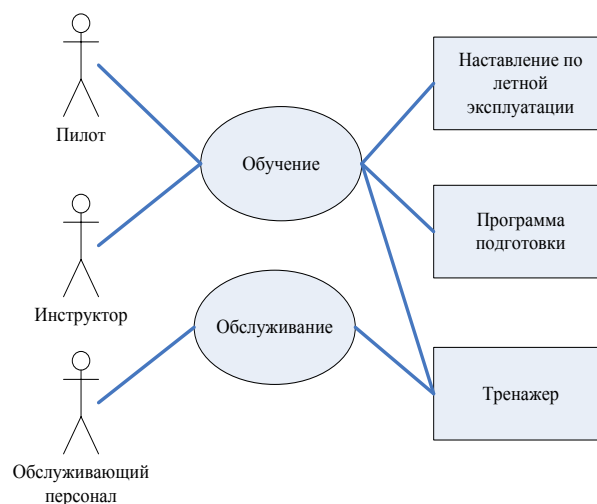


Рис. 1. Использование тренажера

Постановка задачи

Обучение пилотов на тренажере управляется и контролируется инструктором. Инструктор в соответствии с программой подготовки задает начальные условия полета, отслеживает его выполнение, имитирует взаимодействие пилотов с диспетчером,

изменяет условия полета (погода, воздушная обстановка, отказы), оценивает качество самолетовождения и проводит разбор ошибок с пилотами (рис.2).

Для оценки качества самолетовождения и разбора полетов инструктор получает значения необходимых параметров полета с их фиксацией на носителе.



Рис. 2. Функции инструктора

Основной инструмент инструктора для выполнения своих функций – пульт инструктора. Пульта инструктора тренажеров разработки 60-80-х годов представляют собой аппаратные устройства, включающие набор индикаторов и органов управления, и взаимодействующие с другими компонентами тренажера (вычислители, системы визуализации, кабина и т.п.) с помощью аналоговых и дискретных сигналов (рис. 3). Информация, выводимая на индикаторы, и виды управляющих воздействий, осуществляемых с пульта, жестко фиксировались при разработке тренажера.

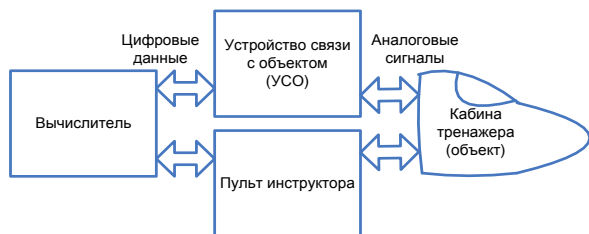


Рис. 3. Основные компоненты авиационного тренажера

Современные пульта инструктора авиационного тренажера реализуются как компьютеризованные аппаратно-программные системы включающие средства визуализации

информации, интерактивного взаимодействия с пользователем и информационного обмена с другими компонентами тренажера [3]. Использование таких систем позволяет отказаться от дорогих специализированных устройств отображения, ввода и фиксации информации, применявшихся ранее (специализированные индикаторы, коммутационные панели, карты-планшеты) и применять стандартные компьютерные и программные компоненты. Программное обеспечение пультов может создаваться на основе современных технологий с применением операционных систем, системного программного обеспечения и современных средств разработки [4, 5].

Подход к созданию программного обеспечения пульта инструктора

Авиационные тренажеры с точки зрения разработчиков программного обеспечения являются хорошо определенным прикладным доменом, для которого известны типичные объекты и их поведение. Такое состояние домена позволяет применять для разработки в нем методы доменного анализа [6], которые дают возможность анализировать предметную область и определять повторно используемые решения. В качестве такого решения для пульта инструктора тренажера предлагается определить абстрактный пульт, для которого специфицировать абстрактные объекты и их обобщенное поведение. Спецификация такого решения в виде шаблона программного обеспечения [7] позволит создавать пульт тренажера путем уточнения и реализации шаблона.

Первичный анализ составляющих и задач пульта позволяет выделить следующие объекты для доменного анализа: индикаторы и органы управления; процессы управления и контроля учебных полетов; информационный обмен с другими компонентами тренажера.

Индикаторы и органы управления

Индикаторы и органы управления пульта тренажера составляют пользовательский интерфейс между тренажером и инструктором. Набор и вид элементов такого интерфейса индивидуальны для тренажера самолета каждого типа, однако существенно похожи. Каждый элемент интерфейса определяется двумя составляющими – типом информации и формой ее представления. В авиационном прикладном домене большинство типов используемой информации известно и описано, также как и обобщенные формы ее

представления. В качестве основы для организации элементов интерфейса предлагается использовать понятие авиационной роли - совокупности функциональных обязанностей, присущих авиационному специалисту (пилот, диспетчер посадки, эксперт по расследованию авиационных происшествий и т.п.), являющимся источником и потребителем информации. Интерфейсные элементы компоуются в логически и эргономически увязанные группы - ролевые виды (role view) (рис. 4). Например, для пилота к таким видам можно отнести окружающую визуальную обстановку из кабины самолета, навигационную приборную панель, панель управления силовой установкой; для авиадиспетчера - экран радиолокационной станции, индикаторы информации, получаемой с борта самолета, и т.п.

Роль инструктора синтезируется путем выбора и настройки определенных интерфейсных элементов. Все интерфейсные элементы по назначению делятся на два класса – индикаторные, предназначенные для отображения информации (индикаторы скорости, высоты, транспаранты состояний, изображения карт и т.п.), и управляющие, предназначенные для ее ввода (переключатели режимов, регуляторы, кнопки ввода значений и т.п.).

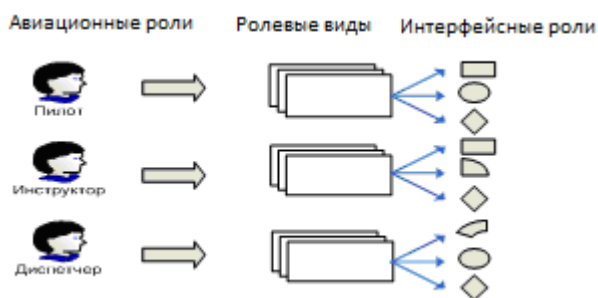


Рис.4. Роли, виды и интерфейсные элементы

Индикаторные элементы по форме отображения можно разделить на следующие:

- шкала-стрелка (например, выотомер, индикатор тяги двигателя, курсовой индикатор);
- табло (например, флаг «Минимальный запас топлива», флаг «Пожар», индикатор пролета маркерного маяка);
- визирные (например, курсо-глицсадный указатель, авиагоризонт);

– цифровые (например, частота радиосвязи, координаты).

Управляющие элементы можно разделить на аналоговые (рукояти управления тягой, регуляторы громкости и т.п.), дискретные (переключатели) и цифровые (клавиатуры).

На рис. 5 показана иерархия абстрактных классов индикаторных интерфейсных элементов, предназначенная для создания интерфейса пульта тренажера в среде операционной системы Windows.

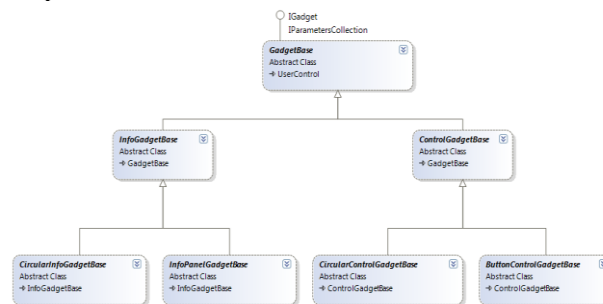


Рис. 5. Абстрактные классы интерфейсных элементов

Класс GadgetBase является базовым для всех информационных элементов и реализует интерфейсы IGadget для взаимодействия с моделями тренажера, IParametersCollection для универсальной работы с коллекциями имен и значений параметров. GadgetBase является производным от класса UserControl визуального элемента управления Windows. Далее цепочка наследования делится на две ветки: базовые классы индикаторов (InfoGadgetBase) и базовые классы управляющих элементов (ControlGadgetBase). Каждая из веток содержит классы для реализации элементов на основе шкал и стрелок (CircularGadgetBase, CircularControlBase) а также классы элементов на основе информационных табло и переключателей (InfoPanelGadgetBase, ButtonControlGadgetBase). Сложные элементы пульта инструктора, такие как авиагоризонт, курсо-глицсадный указатель пока созданы как наследники класса InfoGadgetBase.

Благодаря возможности расширять базу индикаторов и элементов управления, описанная выше иерархия является универсальной и может дорабатываться для любых видов тренажеров.

Для достижения высокой гибкости и эффективности разработки предлагается использовать шаблоны проектирования [7]. В частности для построения индикаторов предлагается использовать шаблон «декоратор», который позволяет гибко

расширять функциональность компонента. Шаблон дает возможность поместить объект индикаторный компонент в объект, «декоратор», что придает ему дополнительные возможности. Например, при наличии базового компонента со стрелочным индикатором, и двух компонентов – «шкала» и «рамка». С использованием декораторов можно достичь следующего решения: индикатор обернуть шкалой и шкалу обернуть рамкой. В результате получится элемент, состоящий из индикатора, шкалы и рамки. Поскольку декораторы «прозрачны», они могут вкладываться друг в друга, добавляя тем самым любое число новых функций. Модуль конструктор компонентов отображения данных использует шаблон декоратор для создания компонентов с любым количеством функций.

Процессы управления и контроля учебных полетов

Процесс управления учебным полетом заключается в задании условий перед началом полета и их изменении во время его совершения. Задание и изменение условий полета должны осуществляться в соответствии со сценарием – схемой полета, разработанной для достижения определенных задач подготовки (штатный взлет и посадка, парирование отказов, пилотирование в сложных метеоусловиях и т.п.). Изменение условий полета производится при наступлении определенных событий (выполнении самолетом определенного маневра, достижении заданной высоты, времени полета и т.п.), которые могут идентифицироваться по совокупности значений определенных параметров.

Контроль полета заключается в получении инструктором параметров полета, необходимых для оперативного управления, и их записи для последующего анализа действий пилотов.

Абстрактный пульт в аспекте управления и контроля полетов должен включать шаблоны программных компонентов для установки начальных условий, изменений условий полета, отображения и записи predetermined параметров полета. Шаблоны должны специфицировать данные (начальные условия, изменения условий, отображаемые и фиксируемые параметры) и поведение (загрузка начальных условий, изменение условий полета по событиям, управление отображением и записью параметров). В качестве средства для записи и хранения сценариев целесообразно

использовать специализированный гипертекстовый язык.

Информационный обмен с другими компонентами тренажера

Современный пульт тренажера функционирует как часть распределенного информационно-моделирующего комплекса, который получает и отправляет информацию через коммуникационные интерфейсы или вычислительную сеть. Поскольку в конкретных тренажерах могут использоваться различные коммуникационные интерфейсы и сетевые протоколы, то обмен абстрактного пульта должен быть специфицирован на прикладном уровне с учетом следующих особенностей тренажера:

- параметры, которыми пульт обменивается с другими узлами вычислительной системы, могут значительно различаться по составу и составлять относительно большое количество (сотни и тысячи);

- тренажер является системой реального времени и, как правило, имеет постоянный цикл вычисления, который должен точно выдерживаться.

При разработке шаблона компонента обмена это обуславливает следующие требования:

- высокая масштабируемость обмена;
- простота и гибкость включения новых параметров;
- возможность передачи набора необходимых параметров несколькими относительно небольшими пакетами.

Для реализации первого требования предлагается использовать в обмене на прикладном уровне XML-пакеты, определенного формата [5] (рис. 6). Такой пакет состоит из заголовка (метка <HD>) и набора параметров (метка <P>). Набор параметров состоит из размеченных пар строк «обозначение – значение», что обеспечивает передачу-прием пакетов с произвольным набором и числом параметров, а также с произвольной длиной обозначений и значений. Для работы с пакетами на каждом узле возможно использовать унифицированное решение, включающее специальную структуру данных, в которой сохраняются принимаемые и отправляемые параметры, и процедуры для формирования и разбора XML-пакетов. Если разработчик тренажера установит сквозные обозначения параметров во всей системе,

возможно, также использовать единые конфигураторы для XML-анализаторов.

<HD>	<T>	ID получателя	<T>	<S>	ID отправителя	<S>	<HD>	<P>	ID параметра	<P>	</>	Значение	</>	...	<P>
------	-----	---------------	-----	-----	----------------	-----	------	-----	--------------	-----	-----	----------	-----	-----	-----

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).

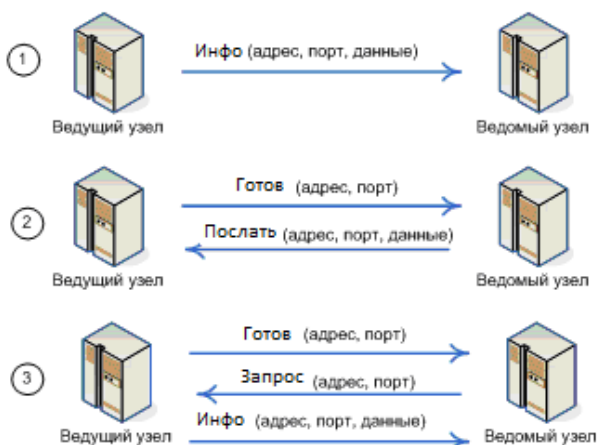


Рис. 7 Схемы обмена

Первая схема предназначена для передачи информации от ведущего узла к ведомым и заключается в пересылке информационного пакета (пакет типа «инфо»), который должен быть принят адресуемым ведомым узлом.

Вторая схема предназначена для передачи информации от ведомых узлов к ведущему. Для этого ведомый узел ожидает от ведущего присылки пакета «готов к обслуживанию», после получения которого пересылает пакет «инфо».

Третья схема предназначена для передачи данных от ведущего узла к ведомому по запросу последнего. Для этого ведомый узел, после получения от ведущего пакета «готов к обслуживанию» пересылает ему пакет «запрос» с перечнем обозначений запрашиваемых значений параметров, а ведущий отвечает пакетом «инфо».

Архитектура программного обеспечения

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

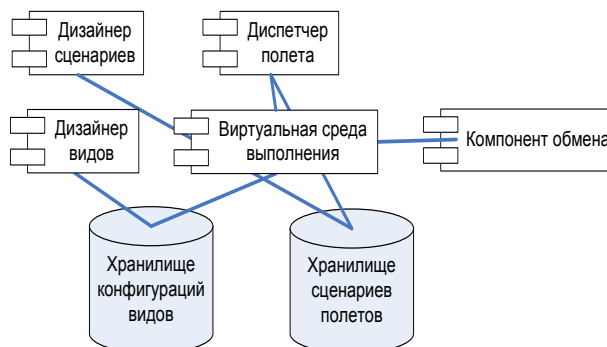


Рис.8. Архитектура средств для реализации пульта инструктора

Дизайнер сценариев предназначен для описания сценариев полетов в рамках программы подготовки и может использоваться инструктором или работником тренажерного центра. В описание входят значения начальных условий полета, правила их изменения в процессе полета и перечень параметров, фиксируемых во время полета на носитель. Выполнение сценария во время полета обеспечивается специальным компонентом времени выполнения - диспетчером полета.

Дизайнер видов предназначен для создания ролевых видов с использованием коллекций информационных элементов. Для реализации видов и элементов управления, специфицированных в конфигурационном файле, используется виртуальная среда выполнения.

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

Хранилища конфигураций видов и сценариев полета обеспечивают сохранение соответствующих данных в XML-формате.

Пример создания программного обеспечения пульта инструктора

Предлагаемый подход к созданию программного обеспечения пульта инструктора был применен при реинженерии авиационного тренажера TL410, осуществляемой в Национальном авиационном университете [1]. Были 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 ролей и виртуальной среды выполнения (рис. 9).

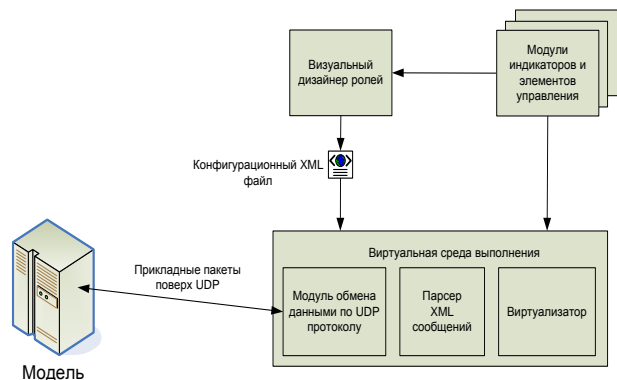


Рис.9 Схема комплекса Рабочее место инструктора

Визуальный дизайнер ролей обеспечивает создание ролей и в визуальном режиме настройку видов (размер, местоположение), их информационное и функциональное наполнение. Результатом работы дизайнера ролей является конфигурационный XML-файл. Каждый элемент вида реализуется отдельным программным модулем (Net-сборки), который подключается динамически и реализует функциональность элемента управления пользователя (Windows User Control). Так создается виртуальное место из отдельных элементов. Коллекции элементов представляют собой реализующие их программные классы, которые могут включаться в виды.

Второй частью комплекса для визуализации работы тренажера является виртуальная среда выполнения. Задача виртуальной среды выполнения – воссоздать из конфигурационного файла необходимые виды и элементы управления на них, а так же обеспечить обмен информацией и взаимодействие с моделью.

Таким образом, все визуальное и функциональное окружение рабочего места инструктора строится из набора элементов управления, каждый из которых представляет собой самостоятельный программный модуль, инкапсулирующий свое поведение и функциональность. Взаимодействие элементов управления с визуальным дизайнером и виртуальной средой выполнения реализовано по принципу позднего связывания, что

позволяет по мере необходимости расширять функциональность программного комплекса.

Взаимодействие между виртуальной средой выполнения и модулями элементов управления реализовано на основе паттерна проектирования Observer. Паттерн Observer позволяет реализовать подписку множества клиентов (модулей элементов управления) на получение информации от одного источника событий (модель) и рассылку им уведомлений. На рис. 10 показаны интерфейсы взаимодействия модулей элементов управления с виртуальной средой выполнения.

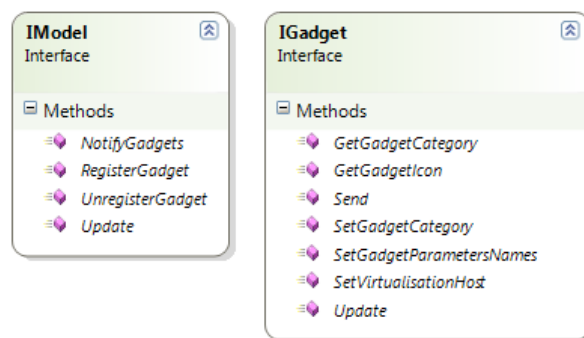


Рис.10. Интерфейсы взаимодействия

Виртуальная среда выполнения реализует интерфейс IModel, который содержит методы подписки и отмены подписки (RegisterGadget(), UnregisterGadget()) элементов управления на получение данных от модели, а также метод NotifyGadgets() который рассылает информацию всем подписанным клиентам. Каждый элемент управления, который предполагает взаимодействие с моделью, реализует интерфейс IGadget. Два основных метода интерфейса IGadget, представляющих интерес, это Send() и Update(). Метод Update() вызывается из интерфейса IModel и обеспечивает взаимодействие «модель – элемент управления». Метод Send() вызывает метод Update() интерфейса IModel и таким образом реализует обратное взаимодействие «элемент управления – модель».

Для того, что бы максимально упростить процесс создания новых элементов управления и по возможности снизить необходимость в их программировании был разработан ряд программных шаблонов.

Обмен данными пульта производится с использованием реализованного прикладного протокола (поверх UDP) и анализатора XML. Каждый пакет состоит из трех блоков (рис.11): заголовок типа пакета (2 байта); длина блока

данных (2 байта); данные в XML-пакете обмена.

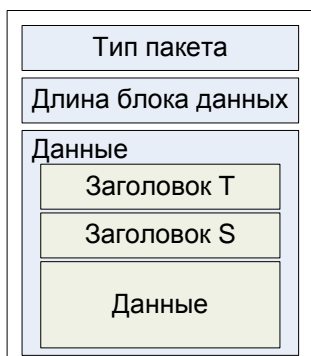


Рис.11. Структура пакета обмена данными

Литература

1. Сидоров Н.А., Недоводеев В.Т., Хоменко В.А., Сердюк И.П., Сидоров Е.Н. Реинженерия программного обеспечения информационно-моделирующих тренажерных комплексов. // Управляющие системы и машины. – К. - 2008. – № 4. – С.68-74.
2. Моржов В.І. Призначення і класифікація авіаційних тренажерів та моделювальних комплексів, КМУЦА, 1997 р.
3. Design of a flight simulator software architecture. Göran Ancker, Jan Wallenberg. – School of Mathematics and Systems Engineering, Växjö University. – 2002. – 91 p.
4. Сидоров М.О., Иванова Л.М., Хоменко В.А. Методологічні принципи реінженерії про

Выводы

Разработка повторно используемых решений – это эффективный способ снижения затрат на реализацию типовых проектов. Зрелость отрасли авиационных тренажеров позволяет создавать высокоуровневые, многократно востребованные решения программного 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 TL410 Национального авиационного университета.

- грамного забезпечення успадкованих авіаційних тренажерів // Мат. VIII Міжнар. наук.-техн. конф. „Авіа-2007”. – К.: 25-27 квітня 2007. – т.1, С. 13.119–13.122.
5. Хоменко В.А, Сидоров Е.Н, Мендзевровский И.Б. Шаблон программного обеспечения устройств связи с объектом авиационных тренажеров. // Проблемы программирования. – К. – 2008. – С. 30 –40.
 6. Prietto-Diaz R. Domain Analysis: An Introduction.-Software engineering Notes. – 1990. – V. 15, № 2. – p. 47–54.
 7. Ларман К. Применение UML и шаблонов проектирования. – М.: «Вильямс», 2002. – 624 с.

Сведения об авторах:



Хоменко Владимир Анатольевич – к.т.н, доцент кафедры инженерии программного обеспечения Национального авиационного университета. Научные интересы: реинженерия программного обеспечения, доменный анализ, технологии разработки и сопровождения программного обеспечения.
e-mail: vlkhomenko@ukr.net



Сидоров Евгений Николаевич – к.т.н, доцент кафедры компьютерных информационных технологий Национального авиационного университета. Научные интересы: доменный анализ, прикладные домены и прикладное программное обеспечение.
e-mail: se_journal@livenau.net