

ГЕНЕРАТОР ЕЛЕМЕНТІВ ІНТЕРФЕЙСУ ПУЛЬТА ІНСТРУКТОРА АВІАЦІЙНОГО ТРЕНАЖЕРУ

Національний авіаційний університет

Запропоновано підхід до вдосконалення інтерфейсу пульта інструктора шляхом створення генератору елементів та доменно-орієнтованої мови.

Вступ

Забезпечення повної безпеки польотів - основна задача повітряного транспорту, головна складова якісної діяльності цивільної авіації (ЦА). Вона включає в себе: взаємозв'язані положення, норми та вимоги, що пред'являються до проектування, створення та експлуатації повітряного судна (ПС), надійність та безвідмовність авіаційної техніки; високу кваліфікацію та дисципліну посадових осіб [1]. Нехтування хоча б однією із цих умов безпеки польотів неминуче веде до авіаційних подій. Аналіз льотних подій [2] із беззаперечною ясністю доводить, що в переважній більшості випадків вони відбуваються не із-за відмов авіаційної техніки, а по причині низького рівня професійної підготовки авіаційного персоналу, нехтування своїми обов'язками та недисциплінованості, тобто завдяки так званому «людському фактору».

Вартість сучасної авіаційної техніки стрімко зростає, тому навчання пілотів (як військових так і цивільних) вимагає застосування найсучасніших технічних та програмних засобів. Використання літальних апаратів як навчального засобу приводить до надмірних витрат, тому в теперішній час, в умовах економічної та екологічної кризи, для підготовки пілотів різних ПС і для підтримки їх професійного рівня широке застосування отримують авіаційні тренажери.

Авіаційний тренажер (АТ) – це частина пілотажно-навчального комплексу «літак-тренажер», який застосовується для підготовки пілотів і представляє собою складну апаратно-програмну моделюючу систему, яка імітує кабінку літака з органами управління та індикаторами, візуальне і звукове оточення пілотів, і поведінку літака при здійсненні різних фаз польотів [3].

Тренування пілотів на тренажері управляється і контролюється інструктором, в обов'язки якого входить: установка початко-

вих умов польоту, контроль якості пілотування, імітація взаємодії пілотів з диспетчером, управління імітацією відмов літака, розбір помилок і особливостей пілотування. Для виконання цих задач в склад тренажеру входять спеціальні засоби, основних з яких є пульт з необхідними індикаторами та органами управління [3].

Аналіз рівня технологій, що використовуються для створення сучасних авіаційних тренажерів, показує експоненціальний ріст їх можливостей. Сучасні інформаційні технології дозволяють вирішувати в реальному масштабі часу будь-які задачі візуалізації та розрахунку динаміки руху. Особлива увага приділяється можливостям програмного забезпечення, які повинні реалізовувати апаратні можливості АТ.

Враховуючи одну з основних вимоги до розробки АТ – фінансового обґрунтування, особливої уваги набуває питання витрат на розробку АТ – пульта інструктора, зокрема, елементів інтерфейсу пульта.

Аналіз останніх досліджень та публікацій

На основі аналізу останніх досліджень та публікацій, можна стверджувати, що сучасні пульти інструктора (SLZ-242, Ми-8/Ми-17) реалізуються як апаратно-програмні системи на основі комп'ютерів, що включають засоби візуалізації інформації інтерактивної взаємодії з користувачем і інформаційного обміну з іншими компонентами тренажеру. Використання таких систем дозволяє відмовитися від дорогих спеціалізованих пристроїв відображення, вводу і фіксації інформації, що застосовувалися раніше (спеціалізовані індикатори, карти-планшети) і використати комп'ютерні і програмні компоненти [4]. Такі пульти інструктора складаються з одного чи декількох моніторів, миші, клавіатури та принтера. На моніторах за допомогою спеціалізованого програмного забезпечення формується зобра-

ження індикаторів приладової дошки і відображається динаміка їх роботи.

Висока вартість затрачених ресурсів (часу і фінансів) на розробку та поняття інтелектуальної власності як самих програмних продуктів, так і конструкцій літальних апаратів робить неможливим знаходження програмного забезпечення для створення пульта інструктора авіаційного тренажера з відкритими джерелами програмного коду.

Як наслідок закритості такого виду інформації, є практична відсутність доступності широкому загалу інших типів програмного продукту, окрім спрямованих в першу чергу на ігроманів (*Microsoft Flight Simulator X* і *X-Plane*).

Найчастіше використовується лише 2 види спеціалізованого програмного забезпечення тренажерів [5], яке може використовуватися при створенні елементів інтерфейсу пульта інструктора.

Система автоматизованого проектування Segambis дає можливість створювати динамічні образи різноманітних індикаторів та цілих приладових дощок.

Програма *DeskSim* (імітатор приладової дошки) дозволяє із заготовок (стрілок, шкал, табло) створювати макети приладових дощок, та легко змінювати розміщення приладів за певними рекомендаціями.

Проаналізовані системи для створення сучасних пультів інструкторів використовують підхід «фіксована функціональність», сутність якого полягає в тому, що склад і функціональність пульта визначаються на основі відомих типових задач, що виконуються інструктором при навчанні пілотів [5]. Основним недоліком такого підходу є те, що після створення пульта, види і форми інформації, що відображається та вводиться не змінюються. Для внесення змін необхідна наявність розробника. Тобто, кожен елемент, що міститься на пульті інструктора, розташовується у чітко визначеному місці і не може бути видалений у разі, якщо він не потрібний за даних умов (наприклад, при відпрацюванні певної польотної ситуації).

Постановка проблеми

Функції інструктора дуже різноманітні, складні і трудомісткі. Але в наслідок обмежених технічних можливостей в наявних пультах, в ергономічному сенсі, діяльність інструктора організована не оптимально. Засоби ви-

дображення інформації належним чином не пристосовані до його функцій, а будь-які пристрої для обробки цієї інформації зазвичай відсутні.

Для забезпечення належного виконання обов'язків інструктору необхідно мати у своєму розпорядженні зручний інтерфейс, який він зможе самостійно створювати і через який отримуватиме інформацію про результати тренування та здійснюватиме управління навчанням.

Основна мета роботи полягає в удосконаленні контролю тренування шляхом підвищення інформативності про хід навчання та зручності імітації польотних ситуацій. Для цього пропонується створити генератор елементів інтерфейсу, який дозволить оперативно формувати інтерфейс пульта. Вхідними даними генератору виступають специфікації, об'єднані в конфігураційний файл. Специфікації базуються на результатах проведення доменного аналізу та описуються за допомогою доменно-орієнтованої мови. Вихідні дані – код на мові C#, який реалізує специфікації (візуалізує елементи інтерфейсу). В результаті, всі елементи інтерфейсу, що представлені в конфігураційному файлі будуть відображатися на пульті інструктора.

Виклад основного матеріалу дослідження

АТ з точки зору розробників програмного забезпечення є чітко визначеним прикладним доменом, для якого відомі типові об'єкти і їх поведінка, що дозволяє використовувати під час розробки програмного забезпечення доменний аналіз [6].

Основна мета ДА – зробити все, щоб інформація була легко доступною.

Для кращого розуміння ДА на рис. 1 показані вхідні і вихідні дані, елементи управління і механізми. Відповідно до рис. 1 інформація збирається із існуючих систем у вигляді вихідного коду, документації, проекту, керівництва користувача і планів проведення тестування, нарівні зі знаннями про домен і вимогами до теперішньої і майбутньої систем. Експерти та аналітики домену добувають відповідну інформацію і знання. Вони аналізують і резюмують її. Доменні спеціалісти організовують і фіксують отримані знання і абстракції у вигляді моделей домену, стандартів, класифікацій і набору компонент багатократного використання. Процес здійснюється за допомогою

методів і технологій, а також методикою управління.

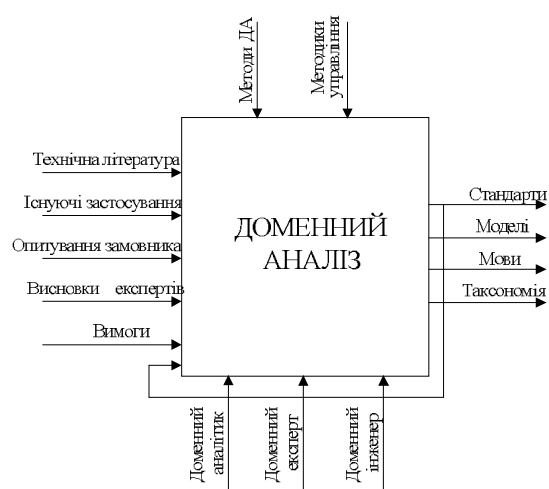


Рис. 1. Контекст ДА

Доменом в даному контексті виступає користувальницький інтерфейс інструктора - інтерфейс, через який інструктор отримує інформацію про тренування і здійснює управління навчанням.

Одним із основних результатів ДА виступає таксономія елементів інтерфейсу пульта інструктора. Вона представляє собою ієрархічну класифікацію основних об'єктів і понять домену. Це допомагає зрозуміти послідовність і організацію елементів інтерфейсу пульта інструктора.

Після визначення основних об'єктів та операцій в домені, важливим є створення доменно-орієнтованої мови, яка буде використовуватися для опису специфікацій домену [7].

В даному контексті доменно-орієнтована мова – мова із синтаксисом і семантикою, розроблена для опису усіх можливих об'єктів (компонент) і дій над ними в описуваному домені. Дана мова може бути використана як повторно використовуване рішення при створенні нових елементів інтерфейсу.

На етапі ДА опис елементів інтерфейсу здійснюється за допомогою мови *XML*, визначаючи таким чином основну структуру елементів. Синтаксичні конструкції *XML* для опису індикаторів представлено в табл. 1.

Таблиця 1.

Синтаксичні конструкції для опису індикаторів

Назва	Опис
1	2
Gauge	Розділ, в якому описується індикатор
<i>Type</i>	Тип індикатора
Param	Розділ, в якому описується параметр літака
<i>ParamName</i>	Унікальне ім'я параметра літака
Label	Підпис індикатора
<i>Caption</i>	Текст підпису
<i>CaptionLocationX</i>	Координата підпису по осі <i>X</i>
<i>CaptionLocationY</i>	Координата підпису по осі <i>Y</i>
Scale	Шкала
<i>MinValue</i>	Мінімальне значення шкали
<i>MaxValue</i>	Максимальне значення шкали
<i>Radius</i>	Радіус шкали
<i>TextOffset</i>	Положення цифр відносно шкали
<i>TextSize</i>	Розмір цифр
<i>MinorTickCount</i>	Кількість малих поділок
<i>MajorTickCount</i>	Кількість великих поділок
<i>StartAngle</i>	Початковий кут шкали
<i>EndAngle</i>	Кінцевий кут шкали
Ribbon	Кольорова стрічка
<i>Name</i>	Назва стрічки
<i>StartValue</i>	Значення на шкалі, з якого починається стрічка
<i>EndValue</i>	Значення на шкалі, на якому закінчується стрічка

1	2
Needle	Стрілка
<i>ShapeType</i>	Кодове значення типу стрілки
<i>StartOffset</i>	Початковий відступ
<i>EndOffset</i>	Кінцевий відступ
SpindleCup	Вісь стрілки
<i>ShapeType</i>	Кодове значення типу осі
<i>Size</i>	Розмір осі
Bounds	Розділ, який містить відомості про границі індикатора
<i>SizeX</i>	Ширина
<i>SizeY</i>	Висота
<i>LocationX</i>	Координата індикатора по осі X
<i>LocationY</i>	Координата індикатора по осі Y

Генератор елементів інтерфейсу перетворює специфікації, представлені за допомогою доменно-орієнтованої мови в вихідний код C#, що реалізує (візуалізує) визначені специфікації.

Кінцева мета ДА інтерфейсу пульта інструктора АТ – забезпечення інструментом, який дозволить інструктору напряму визначати систему у вигляді специфікацій та автоматизувати створення коду із цих специфікацій. В авіаційному домені, так як і в багатьох інших доменах авіаційний спеціаліст не є спеціалістом з розробки програмного забезпечення. Це призводить до двохступінчатого процесу розробки з написанням специфікацій (як правило на неформальній мові) доменним спеціалістом

і їх реалізацію розробником програмного забезпечення, який не є доменним спеціалістом. Оскільки функціональні можливості АТ все частіше представляються за допомогою програмного забезпечення; в результаті виникає ситуація, коли програмне забезпечення розробляється особами, які мають лише обмежене розуміння програмного забезпечення. Така ситуація є джерелом неправильно представлені специфікації; яка призводить до того, що зміщується графік та зростає вартість розробки. Для уникнення такої ситуації пропонується використати редактор (рис. 2) для створення XML-описів (специфікацій), які були розроблені на етапі ДА.

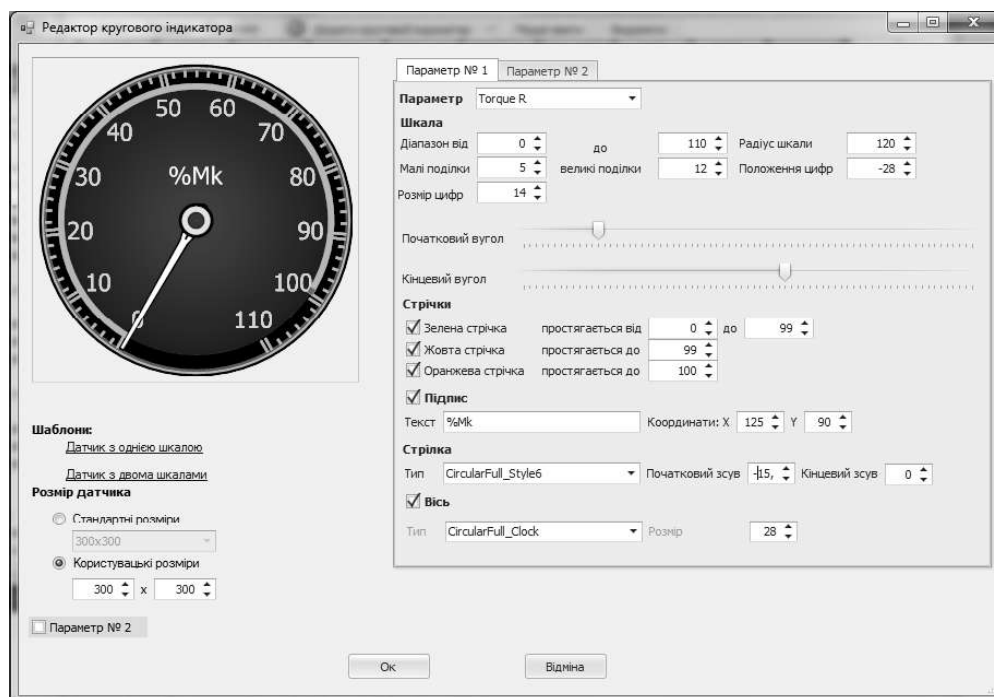


Рис. 2. Інтерфейс редактора кругового індикатора

```

<GaugeType="Circular">
  <ParamPramName="Torque R">
    <LabelCaption="%Mk" CaptionLocationX="125"
      CaptionLocationY="90"/>
    <ScaleMinValue="0" MaxValue="110" Radius="120" TextOffset="-
      28" TextSize="14" MinorTickCount="5" MajorTickCount="12"
      StartAngle="-240" EndAngle="60"/>
    <Ribbons>
      <RibbonName="Green" StartValue="0" EndValue="99"/>
      <RibbonName="Yellow" EndValue="99"/>
      <RibbonName="Red" EndValue="100"/>
    </Ribbons>
    <NeedleShapeType="6" StartOffset="-15" EndOffset="0"/>
    <SpindleCupShapeType="7" Size="28"/>
  </Param>
  <BoundsSizeX="300" SizeY="300" LocationX="545" LocationY="153"/>
</Gauge>

```

Рис. 3. Представлення специфікації кругового індикатору

Так, після налаштування кругового індикатору (вибору шаблону, встановлення розміру, зазначення параметру та властивостей складових: шкали, стрічки, стрілки, вісі) за допомогою редактору, отримуємо специфікацію, яка представлена на рис. 3.

Висновки

Запропоновано генератор, який перетворює специфікації, представлені в конфігураційному файлі, у код C#. За допомогою вихідного коду здійснюється реалізація специфікацій - візуалізація елементів інтерфейсу.

На основі ДА виділено відповідну інформацію домену, джерела інформації, доменну термінологію; створено таксономію елементів, яка допомагає зрозуміти послідовність і організацію елементів інтерфейсу. Інформація, зібрана на етапі ДА використовується при побудові специфікацій домену, які є вхідними даними генератору елементів інтерфейсу пульта. Впровадження та використання генератору надасть можливість спеціалістам значно скоротити час та фінанси, що витрачаються на модернізацію пульта інструктора.

Список літератури

1. Рябокін Ю.М. Доменний аналіз при створенні програмного забезпечення інтерфейсу пульта інструктора авіаційного трена-

жеру / Ю.М. Рябокін // Інженерія програмного забезпечення, –2011. – №1 (5). – С. 35-45.

2. Корченко А.Г. Авиационная безопасность: [Сборник международных документов] / А.Г. Корченко. – К.: НАУ, 2001. – 123 с.

3. Göran Ancker Design of a flight simulator software architecture / Göran Ancker, Jan Wallenberg. – School of Mathematics and System Engineering, Växjö University. – 2002. – 91 p.

4. Рябокін Ю.М. Формування повторно використовуваних рішень при створенні програмного забезпечення пульта інструктора авіаційного тренажеру методом доменного аналізу / Ю.М. Рябокін // Інженерія програмного забезпечення, 2010. – №2. – С. 44-53.

5. Желонкин В.И. Система поддержки исследований по выбору и оптимизации видов электронной индикации / В.И. Желонкин // Вестник Международной Академии проблем человека в авиации и космонавтике. – М. – 2007. № 3 (26). – С. 33-39.

6. Prieto-Diaz R. Domain Analysis: An Introduction. – Software engineering Notes. – 1990. – V. 15, – № 2. – P. 47-54.

7. Фаулер М. Предметно-ориентированные языки программирования / Мартин Фаулер; [пер. с англ.]. – М.: ООО «И.Д. Вильямс», 2011. – 576 с.