

ДОМЕННА ІНЖЕНЕРІЯ ПРИ СТВОРЕННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПУЛЬТІВ ІНСТРУКТОРА АВІАЦІЙНИХ ТРЕНАЖЕРІВ

Національний авіаційний університет

yulia.ryabokin@livenau.net

Розглянуто проблеми підготовки авіаційних спеціалістів та створення пультів інструкторів авіаційних тренажерів. Запропоноване застосування доменної інженерії при створенні програмного забезпечення пультів інструктора авіаційних тренажерів з метою зменшення фінансових та часових витрат

Ключові слова: доменна інженерія, доменний аналіз, методи доменної інженерії, онтологія, програмне забезпечення, багаторазово використовуваний компонент, авіаційний тренажер, пульт інструктора авіаційного тренажеру

Вступ

В даний час в світі склалася ситуація, коли для зменшення впливу людського фактору на безпеку польотів, авіаційне керівництво все більше посилює вимоги до навчання авіаційного персоналу, тоді як авіакомпанії прагнуть скоротити витрати на підготовку до мінімуму. Одним з виходів є використання технічних засобів навчання авіаційного персоналу. Їх використання дозволяє підвищити якість підготовки за рахунок використання сучасних комп'ютерних технологій і при цьому знизити витрати на навчання [1].

Світова та вітчизняна практика показує, що в питаннях підвищення рівня підготовки авіаційних фахівців велику роль відіграють тренажери.

Авіаційний тренажер (АТ) – це імітатор польоту, призначений для наземної підготовки пілотів [1-2].

Основними перевагами використання АТ є [2]: висока економічність, невеликі витрати часу на тренування, можливість всебічного контролю процесу тренування, широкі варіації умов і ситуацій при тренуваннях, можливість «заморожування» умов, повторення та зміна часового масштабу (прискорення чи уповільнення) тренувальної вправи, можливість виконання «польоту» на літаку з гіпотетичними параметрами, незалежність від метеорологічних умов та насиченості по-

вітряним транспортом зони аеропорту, безпека.

Однак, тренажери з рухомою кабіною літака і реальним бортовим обладнанням дуже дорогі у виготовленні та при експлуатації. Це робить їх важкодоступними особливо для середніх, дрібних авіакомпаній та закладів освіти, що здійснюють підготовку авіаційного персоналу. Як наслідок, за показами української та світової статистики, за останнє десятиріччя збільшилась кількість авіаційних пригод та катастроф [2].

При підготовці авіаційних фахівців, на даний час, застосовуються, як сучасні тренажери так і тренажери 70-90-х років. Під час експлуатації тренажерів, що розроблені в 70-90-х роках, інформація, яка виводиться на індикатори і види керуючих впливів, що здійснюються з пульта інструктора жорстко (апаратно) фіксуються при розробці тренажеру [3-4]. Крім того, такі пульти незручні у роботі через те, що введення відмов та початкових умов польоту здійснюється вручну за допомогою кнопок, ручок та тумблерів на пульті, а вся інформація надається інструктору одночасно, без врахування доцільності її перегляду, при відпрацюванні конкретної вправи. Як наслідок, інструктор змушений досить багато часу витрачати на багатокрокові процедури введення відповідних умов польоту та на відбір інформації, яка йому необхідна на той чи

інший момент часу. Отже, внаслідок обмежених технічних можливостей в ергономічному сенсі діяльність інструктора організована не оптимально. Засоби відображення інформації належним чином не пристосовані до його функцій, а будь-які пристрої для обробки цієї інформації відсутні.

Постановка задачі

В Україні значна частина використовуваних АТ – успадковані. Пульти інструкторів більшості успадкованих АТ реалізовані апаратно, на даний час морально застарілі та фізично зношені і не підлягають ремонту із-за відсутності виробництва їх компонентів та запасних частин. Тому постає завдання заміни апаратних частин пульта інструктора на нові, на основі сучасних комп'ютерних інформаційних технологій [3-4]. При цьому, в якості апаратного забезпечення пропонуються персональні або промислові комп'ютери загального призначення, а в якості програмного забезпечення – системне програмне забезпечення (операційні системи, протоколи обміну) і прикладне програмне забезпечення (реалізація функціональності пульта інструктора) [5].

Створення програмного забезпечення (ПЗ) пульта інструктора АТ потребує значних фінансових та часових витрат. А скорочення витрат на розробку такого типу ПЗ, на сьогодні, є актуальною задачею.

Для вирішення цієї задачі в роботі пропонується застосування доменної інженерії при створенні ПЗ з метою визначення (встановлення) багаторазово використовуваних рішень: знань, вимог, результатів аналізу, проектування, тестування, документування та коду, що можуть бути застосовані під час розробки нового ПЗ.

Аналіз останніх досліджень та публікацій

Для вирішення проблем, пов'язаних із застосуванням створених багаторазово використовуваних рішень (адаптування та розуміння) у 80-х роках було введено по-

няття доменного аналізу як фундаментального кроку при створенні реально повторно використовуваних рішень.

Вперше термін «доменний аналіз» (ДА) введено в 1980 році [6]. Він розглядався «як діяльність (процес) по визначенню об'єктів і операцій класу подібних систем в домені». Базуючись на проведених дослідженнях [6, 7], було визначено, що ключ до створення ПЗ багаторазового використання, знаходиться в доменному аналізі, який допускає повторне використання аналізу і проекту – не коду. Ці ідеї були втілені в життя в проекті *SAMP*.

В роботах [8, 9] ДА розглядався як частина процесу розробки ПЗ. Було запропоновано ряд рекомендацій для здійснення ДА – розбиття процесу на три основні етапи, які багаторазово повторюються для різних компонент:

- 1) ідентифікація об'єктів багаторазового використання;
- 2) абстрагування і узагальнення;
- 3) класифікація і каталогізація для багаторазового використання.

Частково на основі вище описаних досліджень в роботах [10, 11] було запропоновано більш узагальнюючу модель для ДА. Ця модель заснована на методології для отримання спеціалізованих класифікаційних фасетних схем в бібліотекознавстві.

В роботах [11, 12] запропонований інший підхід до ДА. Вихідне положення цього підходу полягає в тому, щоб розглядати повторне використання як систему набуття знань. Процес розробки ПЗ розглядається як само вдосконалена система, яка використовує «інфраструктуру багаторазового використання» як джерело набуття знань. Тоді ДА є безперервним процесом створення і збереження інфраструктури багаторазового використання.

Продовжуючи дослідження в даному напрямку в роботі [8, 13] було запропоновано впровадити ДА в процес розробки ПЗ, в якому продукти ДА безперервно переглядаються і уточнюються при створенні нових систем; визначено конкретні процеси і проміжкові продукти, які

передають розуміння процесу ДА; визначено входи і виходи процесу; описано процес ДА колекцією діаграм потоків даних, що відображають послідовність виконання ДА визначеної предметної області, та проміжкові результати процесів.

З 90-х років свою дослідницьку діяльність пов'язують з ДА та доменною інженерією відомі вчені [8, 12]. В своїх роботах вони розглядають ДА як частину процесу розробки ПЗ. Крім того, ними запропоновано ряд рекомендацій для здійснення ДА – розбиття процесу на три основні етапи, які багаторазово використовуються для різних компонентів: ідентифікація об'єктів багаторазового використання, абстрагування та узагальнення, класифікація та каталогізація для багаторазового використання.

З 2000-х років термін «доменний аналіз» зустрічається в літературних джерелах в якості першого процесу доменної інженерії.

Виклад основного матеріалу дослідження

Доменна інженерія – діяльність, що направлена на збір, систематизацію і збереження минулого досвіду побудови систем або частин систем в конкретному домені в формі повторно використовуваних ресурсів (активів, засобів), та забезпечення належних засобів для повторного ви-

користання цих ресурсів (тобто, пошук, розповсюдження, адаптація, збірка) при створенні нових систем [14, 15].

Шляхом застосування доменної інженерії здійснюється створення сімейства систем на основі опису специфіки домену (предметної області) за допомогою доменно-орієнтованої мови, моделі характеристик членів сімейства і накопичення їх в репозитарії [15].

Доменна інженерія призначається для покращення якості розроблених програмних продуктів через повторне використання програмних артефактів. Вона показує, що більшість розроблених програмних систем являються не новими системами, а варіантами інших систем в цьому ж домені (області). Як результат, через використання доменної інженерії, підприємство може максимізувати прибуток; скоротити час виведення нового продукту на ринок, використовуючи концепти (поняття) і реалізацію попередніх систем, застосовуючи їх в цільовій системі.

Доменна інженерія, подібно до прикладної інженерії, складається із трьох основних процесів: аналізу, проектування і реалізації (рис. 1) [15, 16].

Проте, прикладна інженерія фокусується на єдиній системі, а доменна інженерія на сімействі систем.

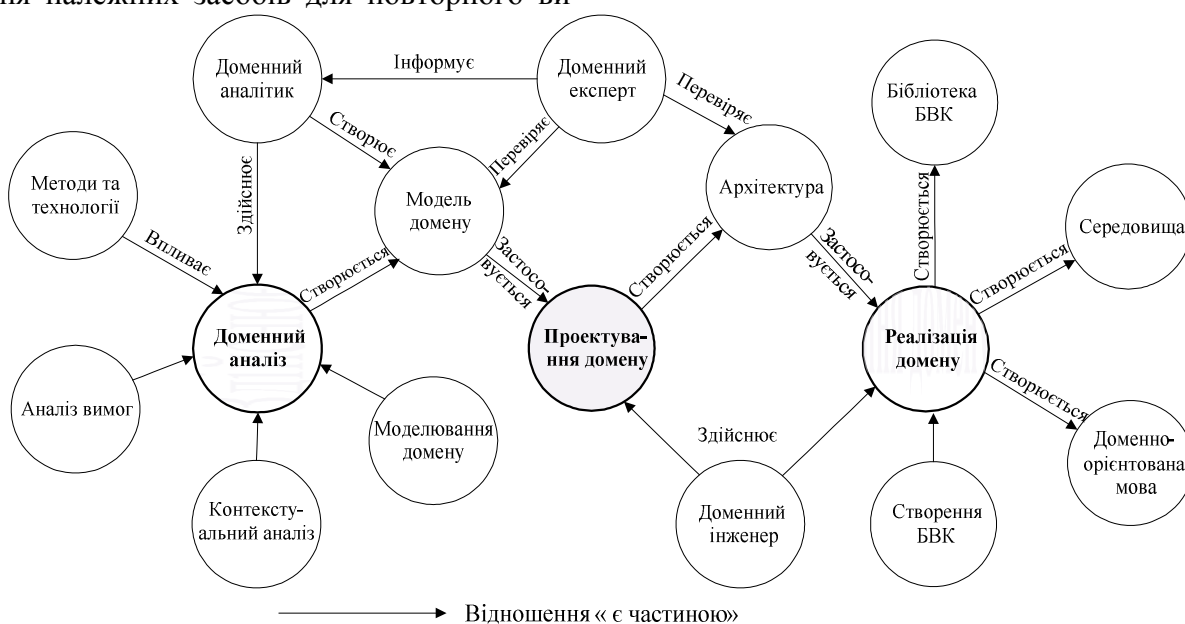


Рис.1. Схема онтології доменної інженерії (IDEF5)

Доменний аналіз – перший та основний процес доменної інженерії, що використовується для визначення доменів та їх границь, збору інформації про домен, виявлення подібностей і відмінностей систем в домені, формування моделі домену [15].

ДА здійснюється під керівництвом доменного аналітика. Найважливішими джерелами інформації, що використовуються під час проведення ДА є існуючі застосування та доменні експерти. Доменним експертам часто властиві знання, які відсутні деінде. Крім того, вони слугують в якості консультантів під час здійснення ДА, визначаючи майбутній напрямок і допомагаючи в перевірці продуктів ДА.

Отже, доменний аналітик здійснює ДА, визначає, збирає, організовує і представляє всю важливу інформацію в домені. Ця інформація базується на існуючих системах в домені; знаннях, отриманих від доменних експертів; політики, процедур і стандартів, що лежать в основі роботи організації; і впроваджених технологій, що розвиваються [11, 15].

Доменний експерт є одним із основних джерел знань відносно домену; передає вимоги і очікування замовника; допомагає доменному аналітику в зборі та аналізі інформації відносно домену [15].

Процес ДА здійснюється за допомогою методів та технологій.

Проектування домену – це визначення архітектури домену за допомогою програмних компонентів [15]. Управління процесом здійснює доменний аналітик, а сам процес проектування – доменний інженер. Доменний експерт здійснює перевірку отриманої архітектури домену.

Архітектура домену – каркас із багаторазово використовуваних компонентів (БВК), активів і формально визначених інтерфейсів, що погоджується із моделлю домену, стандартами організації і оцінюється на відповідність вибраної методології проектування. Архітектура може бути описана на різних рівнях абстракції, відображаючи компоненти, що використову-

ються при побудові систем, їх відношення, інтерфейси [15].

Вихідними даними даного процесу виступають: різноманітні архітектури в різних ракурсах та рівнях абстракції (діаграма класів, яка відображає ієрархію класів в статичному вигляді з інтерфейсами і взаємозв'язками; діаграми станів, що відображають внутрішню поведінку кожного окремого компонента, діаграми діяльності).

Реалізація домену – процес, що здійснюється доменним інженером та починається з визначення та створення повторно використовуваних активів [15].

Визначення відбувається на основі моделі домену і загальної архітектури, що були отримані на попередніх фазах. Інженер здійснює створення активів шляхом використання коду та компіляторів. Результати представляються у вигляді доменно-орієнтованої мови, бібліотек БВК та середовищ.

Реалізація процесів доменної інженерії здійснюється шляхом використання відповідних методів. Врахувавши те, що переважна більшість досліджень направлена на повторне використання програмних продуктів, методи цієї категорії було розділено на чотири підкласи [15, 17, 18]:

- методи для повторного використання компонентів (*Draco, McCain, Prieto-Diaz*);

- методи для повторного використання ресурсів/активів (*HP, ODM*);

- методи для повторного використання архітектури/проекту (*FODA, IDeA, STARS*);

- методи для повторного використання вимог до ПЗ (*Synthesis, JODA*).

Порівняльний аналіз основних представників зазначених підкласів приведено в таблиці 1.

Крім того, методи різняться тим, яким чином вони визначають домен, як використовують доступний домен, архітектуру і експертизу систем. В роботі [15] авторами була запропонована наступна класифікація методів доменної інженерії:

– методи, що базуються на аналізі домену – даний вид методів ґрунтується на ДА і проектуванні; вони відображають набір систем або функціональних областей, які показують подібну функціональність. Представниками даної групи методів є: ОДМ, *FODA*, *FORM*, *RSEB* і *DSSA*.

– методи, що базуються на лінійці програмних продуктів – зосереджені на сумісному використанні властивостей. Тобто, група продуктів, які сумісно використовують спільні властивості і задовольняють потреби тієї чи іншої області ринку. Представниками даних методів є *FAST* і *PuLSE*.

– об'єднання методів доменної інженерії і методів об'єктно-орієнтованого аналізу та проектування. Хоча методи об'єктно-орієнтованого аналізу і проектування не підтримують повторне використання, в них існує декілька застосувань, які намагаються вирішити дану проблему. Представниками цих методів є: *OOram*, *JODA*, *SHERLOCK*;

– методи *SODA* (оцінки і проектування стратегічних варіантів) – забезпечують підхід до проектування довговічних архітектур системи. Ця мета досягається аналізом різних стратегічних сцена-

рій до фінальних архітектур. Таким чином *SODA* намагається передбачити майбутні вимоги або технологічні зміни;

– методи архітектурного аналізу: перехід від моделювання домену до визначення архітектури домену – під час архітектурного аналізу доменний інженер вибирає відповідний підхід для побудови загального проекту. Цей підхід має відповідати вимогам і бути достатньо гнучким в забезпеченні відмінностей між окремими застосуваннями, що побудовані із загального проекту. Представниками даної групи методів є: *OCA*.

Враховуючи результати порівняння методів ДА та доменної інженерії за визначеними критеріями, що представлені в таблиці 1, набуття популярності методів ОДМ та *FODA* в роботі пропонується скористатися методом ОДМ, оскільки він є більш зрілим та задокументованим, та акцентує свою увагу на доменній інженерії наслідуваних систем військової авіації.

Метод організаційного доменного моделювання (ОДМ) [19] розроблено *Mark'om Simos*, для систематизування ключових аспектів процесу доменного моделювання і забезпечення повного каркасу життєвого циклу доменної інженерії.

Таблиця 1. Порівняння методів доменної інженерії

№ п/п	Методи доменної інженерії	Мета методу	Задокументованість	Орієнтованість на	Технологія аналізу
1.	<i>Neighbors</i>	Удосконалення БВК	Низька	Військові системи	–
2.	<i>Prieto-Diaz</i>	Удосконалення БВК	Середня	Військові та урядові системи	Функціональна декомпозиція
3.	<i>HP</i>	Конструювання БВК	Низька	Системи управління периферійними пристроями	Функціональна декомпозиція
4.	<i>JODA</i>	Конструювання БВК	Низька	Військові системи, авіоніка	Об'єктно-орієнтований
5.	<i>DADO</i>	Наповнення бібліотеки БВК	Низька	Системи управління телекомунікаціями	Об'єктно-орієнтований
6.	<i>Synthesis</i>	Зменшення вартості адаптації	Низька	Системи управління супутниками	Об'єктно-орієнтований
7.	ОДМ	Конструювання БВК	Висока	Системи військової авіаційної галузі	Концептуальний аналіз
8.	<i>FODA</i>	Конструювання БВК	Висока	Військові системи та системи управління телекомунікаціями	Функціональна декомпозиція

Метод ОДМ використовується різними організаціями і піддається інтеграції з різноманітними процесами програмної інженерії, методами і технологіями реалізації. ОДМ зосереджується на концепціях, які виникли на основі робіт багатьох дослідників і практиків в області «систематичного повторного використання ПЗ».

Основні процеси, які реалізуються згідно методу ОДМ є наступними [19]:

- визначення домену;
- збір та накопичення інформації щодо домену;
- доменний аналіз;
- формалізоване зображення компонент;
- розробка середовища БВК.

Перший процес – визначення домену, визначає загальні риси та межі домену, зв'язок з іншими доменами.

Під час виконання процесів збору та накопичення даних здійснюється визначення методів збору та джерел інформації щодо домену.

ДА представляє собою один із найважливіших процесів доменної інженерії за будь-яким її методом, оскільки відображає властивості, ролі та зв'язок основних сутностей домену, які поєднуються в моделі домену. На основі моделі здійснюється визначення та формування БВК.

Процес ідентифікації та формалізованого зображення компонент полягає в створенні та формалізованому представленні компонент, збереженні їх у середовищі БВК.

В роботі, областю дослідження виступають АТ, що представляють собою наземне обладнання, на яких здійснюються тренування, перенавчання льотчиків і екіпажів літаків для польотів на літаках Л-410 та Ту-154 [20, 21].

Вхідними даними процесу є організаційна інформація, зокрема: технічні описи літальних апаратів та АТ, архітектури пультів інструктора АТ, управляючі та регулюючі документи, які властиві авіаційній галузі, знання відносно пультів інструктора, притаманні авіаційному персоналу. На етапі виконання даної фази

виділяються кандидати зацікавлених сторін; формується досье зацікавлених сторін, визначаються проектні цілі та межі домену.

В роботі доменом виступають пульти інструктора АТ. Обраний домен включає необхідні та загально прийняті елементи, що складають пульти інструктора АТ. Домен представляється набором систем (панеллю пульта інструктора, управління та контролю польоту, інформаційного обміну), що взаємодіють між собою.

Панель пульта інструктора представлена екземплярами систем, в контексті даного домену, двома групами: індикаторами та елементами (органами) керування.

Система управління та контролю польотів представляє собою сукупність компонент, що забезпечують здійснення управління тренувальним польотом та контролю якості пілотування.

Система інформаційного обміну – система обміну параметрами з різними компонентами тренажеру.

Для збору та накопичення інформації, в роботі, були використані наступні методики збору даних: прямі, непрямі та незалежні.

Серед прямих методик збору даних використано інтерв'ю, загальне та активне спостереження; непрямих – віддалене спостереження; незалежних – аналіз документації.

Збір даних за прямими методиками здійснено за допомоги співробітників тренажерного центру, колишніх пілотів та інженерів-конструкторів авіаційних приладів.

За роботою інструктора та використанням пультів під час проведення тренувальних польотів на АТ було використано віддалене та активне спостереження.

Незалежна методика збору даних полягає в аналізі документації, і була використана при ознайомленні з технічними описами літальних апаратів, АТ.

Після визначення домену та збору інформації щодо домену, ґрунтуючись на аналізі зацікавлених сторін і їх інтересів,

здійснюється перехід до наступного процесу методу – ДА. Основна мета процесу ДА – створення моделі пульта інструктора АТ.

Модель пульта інструктора АТ представляється у вигляді системи понять, що визначаються термінологією, якою користуються авіаційні експерти; відносин між поняттями та їх інтерпретацію. В роботі модель пульта інструктора АТ зображується у вигляді онтології.

Для зображення знань, отриманих на етапі застосування методу доменної інженерії, у вигляді БВК, в роботі пропонується формалізоване подання цих компонент у вигляді онтологій: графічні та аналітичні нотації, математичні моделі доменних сутностей, шаблони програмного коду.

Формалізоване зображення БВК у роботі задається декларативними моделями виду:

$$C_i = (g, a, m, t),$$

де C_i – позначення моделі i -го компоненту;

g – графічний опис онтології (стандарт *IDEF5*), a – аналітичний опис онтології у вигляді формул логіки предикатів першого порядку;

m – математичне зображення функціональності i -го компонента;

t – шаблон програмного коду i -го компоненту.

При створенні середовища БВК в роботі застосовується адаптивний підхід, сутність якого полягає у використанні наперед визначеної багаторазово використовуваної архітектури застосування і

множини БВК для заповнення вільних місць в архітектурі ними.

Відповідно, середовище представляється як:

$$S = (R, A),$$

де S – середовище; R – репозитарій БВК; A – багаторазово використовувана архітектура доменно-орієнтованого застосування.

Компонентами репозитарію БВК є інтерфейсні елементи, параметри, журнал та програми польоту, архітектура доменно-орієнтованого застосування. Їх пошук проводиться на основі запиту відповідно до вимог розробки.

Виконуюча частина кожного БВК, залежно від його типу, в середовищі зображується двійкою виду:

$$БВК_i = (t_i, e_i),$$

де t_i – шаблон програми, який забезпечує візуальне представлення об'єкту (V_i); e_i – редактор $БВК_i$.

В роботі, створено середовище (програмний засіб), що забезпечує можливість створення ПЗ пультів інструктора АТ, та можливість імітації їх роботи. Основні функціональні можливості програмного середовища зображено на рис. 2.

Основними акторами наведеної діаграми виступають: інструктор та уповноважений спеціаліст. Інструктор здійснює керівництво ходом тренування, а уповноважений спеціаліст (розробник) – створення ПЗ та налаштування відповідної панелі пульта інструктора, журналу, параметрів та програм польоту.

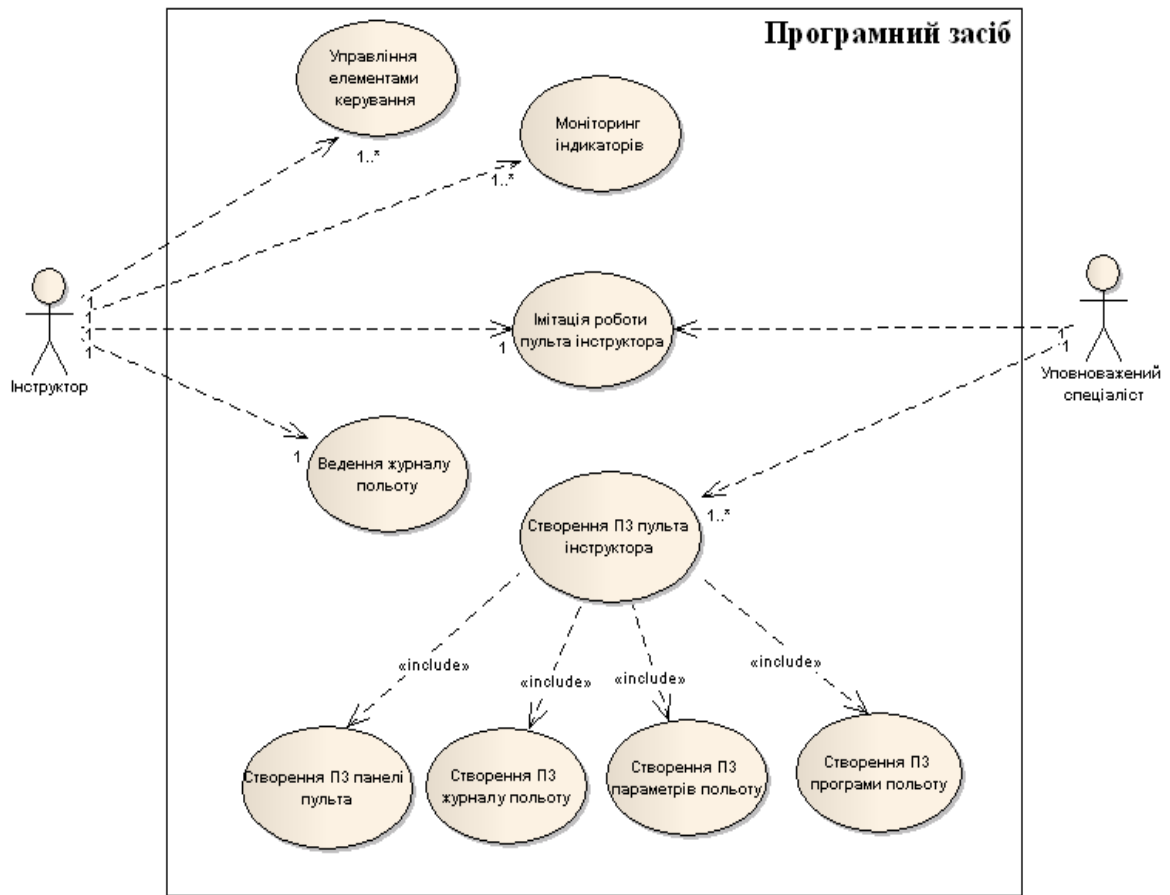


Рис. 2. Варіанти використання програмного середовища

Висновки

В роботі здійснено аналіз методів доменної інженерії, для застосування одного з них, як методологічної основи створення середовища БВК. Встановлено, що метод доменної інженерії, організаційне доменне моделювання найбільш відповідає основним вимогам для покладення його в основу розробки програмного середовища для створення ПЗ пультів інструктора, у зв'язку з його орієнтованістю на застосування в успадкованих системах авіаційної галузі.

Компоненти середовища створюються виходячи з досвіду, який накопичено в домені та мають задане формалізоване представлення у вигляді декларативних моделей.

Програмне середовище надає можливість створювати ПЗ пультів інструктора різних типів АТ із готових компонен-

тів, які зберігаються в репозитарії та налаштовувати їх відповідно до вимог замовника

Список літератури

1. Серегин Г.Н. Авиационные тренажеры – реальный путь к повышению безопасности полетов / Г.Н. Серегин // «Право и безопасность». – 2006. – №3-4. С. 20-21.
2. Кулик Н.С. Энциклопедия безопасности авиации / Н.С. Кулик, В.П. Харченко, М.Г. Луцкий, А.Г. Кучер, Ф.Н. Яновский и др. – К.: Техніка. – 2008. – 1000с.
3. Рябокінь Ю.М. Формування повторно використовуваних рішень при створенні програмного забезпечення пульта інструктора авіаційного тренажеру методом доменного аналізу / Ю. М. Рябокінь // Інженерія програмного забезпечення, 2010. – №2. – С.44-53.

4. Луцький М.Г. Підтримка придатності та продовження експлуатації програмного забезпечення авіаційної техніки / М. Г. Луцький, М.О. Сидоров, Ю.М. Рябокінь // Проблеми програмування. – К., – 2010. – №2-3. Спеціальний випуск. – С. 229-236.
5. Сидоров М.О. Методологічні принципи реінженерії програмного забезпечення успадкованих авіаційних тренажерів / М.О. Сидоров, Л.М. Іванова, В.А. Хоменко // Матеріали VIII Міжн.наук-техн. конф. АВІА-2007.– 2007. – Т.1. – С.13.119-13.122.
6. Neighbors J.M. The Draco Approach to Constructing Software from Reusable Components / J.M. Neighbors // IEEE Trans. on Softw. Eng. – 1984. – №3. – P. 564–576.
7. Common Ada Missile Packages (CAMP): Technical Proposal Prepared by Raytheon Company, Missile Systems Division, Bedford / Airforce Armament Laboratory. – 1984. – Vol.1. – P.589–594.
8. Сидоров Н.А. Восстановление, повторное использование и переработка программного обеспечения. I / Н.А. Сидоров // УСим. – К. – 1998. – № 3. – С.74-83.
9. McCain R. Reusable Software Component Construction: A Product-Oriented Paradigm / R. McCain // IBM Federal Systems Division, Houston TX. – 1986. – P.136.
10. Prieto-Diaz R. Classifying Software for Reusability / R. Prieto-Diaz, P. Freeman // IEEE Software. – Vol. 4 (1). – 1987. – P.–6-16.
11. Prieto-Diaz R. Domain Analysis: An Introduction / R. Prieto-Diaz // Software engineering Notes. – 1990. – Vol. 15(2). – P.47–54.
12. Arrango G. Domain Analysis Methods. In Software Reusability / G. Arrango, R. Prieto-Díaz //Ellis Horwood. – 1994. – P.17-49.
13. Prieto-Diaz R. Domain Analysis for Reusability / R. Prieto-Diaz // Computer Software and Applications Conference Compass' 87. – 1987. – P. 112-119.
14. Wartik S. Criteria for comparing reuse-oriented domain analysis approaches / S. Wartik, R. Prieto-Diaz // Software productivity consortium. – 1991. – P.31-67.
15. Alana E. Domain engineering methodologies survey / E. Alana, A. Rodriguez // GMV AEROSPACE AND DEFENCE S.A. – 2007. – 38p.
16. Лаврищева Е.М. Методы и средства инженерии программного обеспечения. Учебник / Е.М. Лаврищева, В.А. Петрухин // Московский физико-технический институт. – 2006. – 304с.
17. Ferrés X. An Evaluation of Domain Analysis Methods / X. Ferrés, A. Vegas // In 4th CAiSE/IFIP8.1 International Workshop in Evaluation of Modeling Methods in Systems Analysis and Design. – 1999. – P.1-13.
18. Kang Kyo C. Future-Oriented Domain Analysis / C. Kang Kyo, G. Sholom, J. Hess // Feasibility Study: Software Engineering Institute. – 1990. – P.1.161.
19. Simos M., Creps D., Klinger C., Levine L Lockheed Martin Tactical Defense Systems. Organization Domain Modeling (ODM) Guidebook: Version 2.0 Manassas: STARS-VC-A025/001/00. – 1996. – 509p.
20. Техническое описание. Книга 1. Основные характеристики тренажера ТЛ-410. – 1978. – 102с.
21. Göran A. Design of a flight simulator software architecture / A. Göran, J. Wallenberg // School of Mathematics and System Engineering: Växjö University. – 2002. – P.91.

Статтю подано до редакції 24.12.2014