

ДОМЕННИЙ АНАЛІЗ

УДК 004.413:338.5

Національний авіаційний університет

Ю.М. Рябокінь

ДОМЕННИЙ АНАЛІЗ ПРИ СТВОРЕННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРФЕЙСУ ПУЛЬТА ІНСТРУКТОРА АВІАЦІЙНОГО ТРЕНАЖЕРУ

Вступ

Щорічно послугами повітряного транспорту користуються понад 2,5 млрд. пасажирів. Наприклад, аеропорти Північної Америки перевозять близько 1,2 млрд. пасажирів, більш як 700 млн. пасажирів обслуговують аеропорти Європи і понад 500 млн. аеропорти Азіатсько-Тихоокеанського регіону. Що стосується вантажних перевезень, то аеропорти північного континенту щорічно перевозять близько 25 млн. тонн вантажів, Азіатсько-Тихоокеанського регіону - 12 млн. тонн і Європи - 10 млн. тонн [1]. В Україні за минулі роки, згідно статистичних даних, послугами українських авіакомпаній скористалось понад 6 млн. пасажирів, перевезено 87,9 тис. тонн вантажу [2].

Поряд з високими досягненнями авіації існують і недоліки – авіакатастрофи. В середньому щорічно в світі відбувається близько 500 різних авіакатастроф. В таких умовах особливу

Интерфейс пульта инструктора авиационного тренажера представляет собою засіб, через який інструктор отримує інформацію про політ і здійснює управління тренажером. Для зниження витрат на розробку програмного забезпечення пульта шляхом формування повторно використовуваних рішень пропонується застосувати доменний аналіз. В якості метода доменного аналізу використовується Організаційне доменне моделювання.

Интерфейс пульта инструктора авиационного тренажера представляет собой устройство, через которое инструктор получает информацию о полете и осуществляет управление тренажером. Для снижения затрат на разработку программного обеспечения пульта путем формирования повторно используемых решений предлагается применить доменный анализ. В качестве метода доменного анализа используется Организационное доменное моделирование.

The interface of instructor air simulator console provides a device through which the instructor receives the flight information and controls the simulator. For reduce the costs on software development of the console by creating reusable solutions proposed to use domain analysis. As a method of domain analysis is used Organizational domain modeling.

Ключові слова: авіаційний тренажер, пульт інструктора, повторне використання, доменний аналіз, методи доменного аналізу, Організаційне доменне моделювання

актуальність набуває проблема забезпечення безпеки польотів [3].

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 та багато іншого. Нехтування хоча б однією із цих умов безпеки польотів неминуче веде до авіаційних подій. Аналіз льотних подій [3] 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 неді-

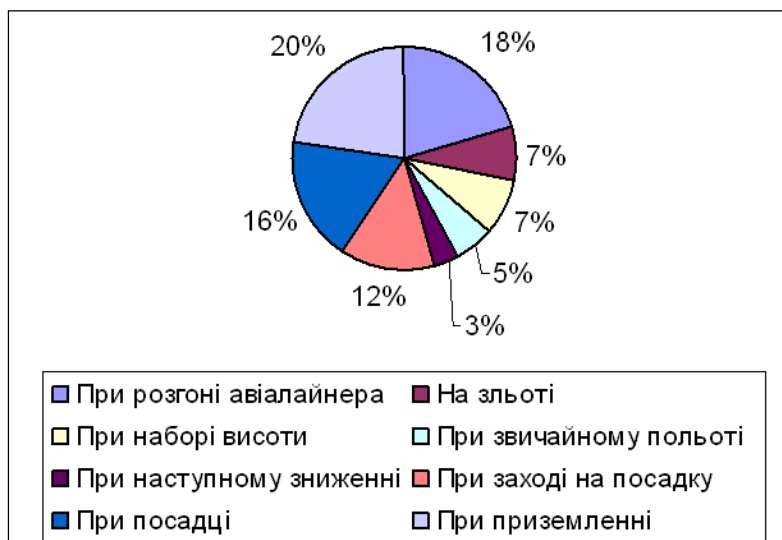


Рис. 1. Льотні події з вини «людського фактору»

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 повітряних суден і для підтримки їх професійного рівня широко застосовують авіаційні тренажери. Наприклад, для бойової авіації співвідношення вартості навчання на тренажері та в польоті складає 1/20, для широкофюзеляжних літаків доходить і до 1/40, навіть при тому, що вартість сучасних тренажерів рівня D складає більше ніж 9 млн. євро [4].

В даний час тренажерна підготовка авіаційного персоналу є складовою частиною державної програми забезпечення безпеки польотів повітряних суден цивільної авіації. Виконання цієї програми в повному обсязі може знизити в країні кількість аварій, пов'язаних з людським фактором, в 2-2,5 рази.

Аналіз рівня технологій, що використовуються для створення сучасних авіаційних тренажерів, показує експоненціальний ріст їх можливостей.

Сьогодні комп'ютери дозволяють вирішувати в реальному масштабі часу будь-які задачі візуалізації, розрахунку динаміки руху та інш. На перший план виступають можливості програмного забезпечення, які повинні вміти реалізовувати існуючі апаратні можливості.

Для написання такого програмного забезпечення потрібні навички не лише програмістів, але й цілої низки інших фахівців авіаційної

галузі. Зокрема, на даний час, основними розробниками програмного забезпечення (ПЗ) у галузі авіації є самі авіакомпанії та конструкторські, авіабудівні підприємства. Що пов'язане в першу чергу із наявністю знань у спеціалістів компаній, як повинна функціонувати система. Тобто, ведеться розробка доменно-орієнтованого програмного забезпечення.

В умовах економічної кризи завдання зниження витрат на створення програмного забезпечення авіаційних тренажерів, при збереженні якості та надійності програмного забезпечення, набуває особливої актуальності. Для досягнення цієї мети в статті пропонується застосовувати повторно використовувані рішення (ідеї, знання, вимоги та інш.); доменний аналіз (ДА), що передуює процесу встановлення вимог до програмного забезпечення.

Аналіз останніх досліджень і публікацій

В літературі широко описується принцип повторного використання [5-6]. Найбільш ефективно принцип досліджується в наукових центрах США та Європи: Software Productivity Consortium, University of Maryland, University of Texas, Loral Federal Systems, Gorge Manson University, Microsoft Corp., Microelectronics and Computer Technology Corp., Software Engineering Institute.

Повторне використання практикується відомими виробниками ПЗ, наприклад, AT&T, IBM, Ericsson, Hewlett-Packard, Motorola, Nec, Toshiba, при цьому, частка повторно використовуваного коду оцінюється наступними значеннями [7-9]: Hewlett-Packard – 25-50%; AT&T – 40-92% для телекомунікаційних засто-

сувань; Ericsson AXE – до 90%; Motorola – 85%. Крім того, завдяки реалізації цього шляху, спостерігається скорочення часу виходу ПЗ на ринок в 2-5 раз, зменшення числа помилок в 5-10 раз, зменшення вартості супроводження в 5-10 раз і зменшення загальної вартості розробки від 15 до 75% [7]. Вказані значення досягаються за рахунок скорочення витрат на пошук рішень задач, що виникають при розробці ПЗ, зменшення витрат на кодування, підвищення надійності роботи ПЗ, оскільки застосовуються готові, добре протестовані компоненти.

За останні роки виробники розробили власні технології повторного використання і створили відповідні підрозділи для їх впровадження.

Принцип повторного використання досліджували багато вчених: Л. Бабенко, І. Вельбицький, М. Сидоров, V. Basili, T. Biggerstaff, G. Caldiera, E. Chisofsky, E. Horovitz, R. Pioto-Diaz, H. Rombach, I. Sommerville, P. Wegner.

Незважаючи на велику кількість програмних компонент, що були доступні для повторного використання, більшість розробників надавали перевагу створенню власних компонент, оскільки, існуючі компоненти були важкими як для розуміння так і для адаптування до нових застосувань.

Для вирішення цієї проблеми у 80-х роках було запропоновано ввести поняття доменного аналізу як фундаментального кроку щодо створення реальних повторно використовуваних компонентів.

Вперше термін «доменний аналіз» ввів Neighbors в 1980 році [10]. Він розглядав доменний аналіз «як діяльність (процес) по визначенню об'єктів і операцій класу подібних систем в домені».

Базуючись на своїх дослідженнях з системою Dгасо, він зазначив, що ключ до створення програмного забезпечення багатократного використання, знаходиться в ДА, який допускає повторне використання аналізу і проекту – не коду. Його ідеї були втілені в життя в проекті Common Ada Missile Packages (СAMP) [11]. Проект СAMP – перший в своєму роді проект, безпосередньо проведений з використанням ДА. Ні Neighbors, а ні учасники проекту СAMP не розглядали питання про те, як провести ДА. Вони концентрували увагу на результатах, а не на процесі.

McCain [12] та М.О. Сидоров [13] в своїх роботах розглядали ДА як частину процесу розробки ПЗ. Вони запропонували ряд рекомендацій для здійснення ДА – розбиття процесу на три основні етапи, які багатократно повто-

рюються для різних компонент: 1) ідентифікація об'єктів багатократного використання, 2) абстрагування і узагальнення, 3) класифікація і каталогізація для багатократного використання. Спеціалізована модель McCain була успішно протестована в декількох програмах IBM Federal Systems.

Частково на основі вище описаних досліджень Prietto-Diaz [14] запропонував більш узагальнюючу модель для ДА. Ця модель основана на методології для отримання спеціалізованих класифікаційних фасетних схем в бібліотекознавстві [15]. Цей метод був успішно застосований в державних системах корпорації General Telephone & Electronics. Пізніше метод було переглянуто і доопрацьовано для STARS Reuse Library Process Model. В оновленому методі акцент зроблено на набутті знань та їх структуруванню.

Arango [16] здійснюючи дослідницьку діяльність запропонував свій 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 багатократного використання» як джерело набуття знань. Тоді ДА є безперервним процесом створення і збереження інфраструктури багатократного використання.

Модель Аранго була успішно протестована в Фінляндії (Cone Corporation) в області розробки ПЗ ліфта.

Prieto-Diaz [15] запропонував впровадити ДА в процес розробки ПЗ, в якому продукти ДА безперервно переглядаються і уточнюються при створенні нових систем; визначив конкретні діяльності (процеси) і проміжкові продукти, які передають розуміння процесу ДА; визначив входи і виходи процесу ДА; описав процес ДА колекцією діаграм потоків даних, що відображають послідовність виконання ДА визначеної предметної області, та проміжкові результати процесів.

Software Engineering Institute розробив метод Feature-oriented domain analysis (FODA) [17]. Він базується на визначенні загальних (спільних) властивостей (можливостей) у класі систем і є продуктом вивчення і оцінки декількох методів ДА.

Кожного року проводяться декілька міжнародних конференцій, на яких обговорюються доменний аналіз, методи доменного аналізу, практичне застосування ДА. Видано близько сотні наукових статей по цій тематиці.

Постановка завдання

Головною проблемою, що виникає при повторному використанні є створення компонент, що можуть бути використані в застосуваннях, відмінних від тих застосувань, для яких вони спочатку розроблялися. Нажаль, незважаючи на наявність великої кількості програмних компонент для повторного використання, програмісти вважають за краще створювати свої власні, оскільки наявні компоненти важко зрозуміти і адаптувати до нових застосувань.

Для подолання цього недоліку та створення реально повторно використовуваних компонент застосовується доменний аналіз, який розглядається в статті на прикладі створення ПЗ інтерфейсу пульта інструктора АТ.

Поняття доменного аналізу

В загальному вигляді домен – це сфера дій та інтересів, область [13].

В контексті ПЗ це поняття розглядається як область застосування, сфера, в якій розроблюються програмні системи [13].

На даний час існують різні варіанти визначення поняття ДА [18], зокрема:

- 1) ДА – визначення операцій, об'єктів даних, властивостей і абстракцій підходящих для проектування рішень проблем в даному домені;
- 2) ДА – діяльність (процес) доменної інженерії в якій (ому) знання домену вивчаються та формалізуються як визначення домену і специфікація домену. Повторне використання ПЗ підхід, що включає об'єднання програмних

компонент, підсистем і т.д., в єдину прикладну систему;

3) ДА – процес визначення, організаційного збору, аналізу і представлення моделі домену і програмної архітектури з вивчення (дослідження) існуючих систем, що лежить в основі теорії розвиваючої технології і історій розробки в межах інтересу домену;

4) ДА – аналіз систем в домені для знаходження подібності та відмінності між ними.

Доменний аналіз забезпечує виявлення спільних об'єктів, операцій і відношень серед подібних систем в домені з метою їх повторного використання в розробці нових застосувань цього та інших доменів [19]. ДА схожий на системний аналіз, але замість виконання аналізу однієї системи це робиться для декількох пов'язаних систем.

Під час розробки ПЗ, породжується інформація декількох видів, від аналізу вимог до специфікацій проектів та до вихідного коду.

Однією із цілей ДА – зробити цю інформацію доступнішою.

При прийнятті рішення про повторне використання, тобто, при намаганні вирішити чи можна компонент використовувати багатократно, програміст повинний розуміти контекст, який запропоновано проектувальником для побудови (зборки) компоненту, як він є.

Для кращого розуміння ДА на рис. 2 показані вхідні і вихідні дані, елементи управління і механізми.



Рис. 2. Контекст ДА

Відповідно до рис. 2 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.

За допомогою доменних спеціалістів знання і абстракції організовуються і фіксуються у вигляді моделей домену, стандартів, класифікацій і набору компонент багатократного використання.

Процес здійснюється за допомогою методів і технологій, а також методикою управління. ДА може бути однією із умов розробки ПЗ, так як, це постійний процес, що безперервно удосконалюється.

Оскільки ресурси багатократного використання становляться доступними і створюються нові системи, вони використовуються для удосконалення існуючих моделей домену і сприяють багатократному використанню бібліотек.

Класифікація методів доменного аналізу

На сьогодні існує велика кількість методів доменного аналізу. Це розмаїття може розглядатися як перевага (завжди можна запропонувати метод, який відповідає конкретним потребам розробника), або як недолік (виникає питання чи є розмаїття відповіддю на недоліки або спеціалізацією існуючих методів).

Arango та Wartik здійснили оцінку різноманітних методів ДА [20], зосередивши увагу на відмінностях в процесах ДА (як отримується доменна модель), а не на інших особливостях методів. Порівняльні дослідження на декількох методах ДА привели їх до висновку, що всі методи ДА слідує загальним процесам і існує більше подібностей між методами ніж відмінностей.

Автори X.Ferre і S.Vegas в своїй роботі [21] здійснили оцінку різноманітних методів ДА, фокусуючись на відмінностях в процесах ДА та запропонували класифікувати методи ДА в залежності від типу елемента, що буде багатократно використовуватись:

- для повторного використання програмних продуктів;
- для повторного використання програмних процесів;
- для повторного використання програмних технологій;
- для повторного використання досвіду.

Переважає більшість досліджень спрямована на повторне використання програмних продуктів. В зв'язку з цим методи ДА для повторного використання програмних продуктів можна розділити на чотири підкласи:

1) повторного використання компонентів (Draco, McCain, Prieto-Diaz);

2) повторного використання активів (ресурсів) (HP, ODM);

3) повторного використання архітектури/проекту (FODA, IDeA, STARS, DADO);

4) повторного використання вимог до ПЗ (Synthesis, JODA).

В останні роки увага акцентується на двох методах ДА – FODA та ODM [22], оскільки вони більш зрілі і краще задокументовані. В роботі пропонується скористатися методом ODM.

Опис підходу ODM

Organizational Domain Modelling (ODM) – організаційне доменне моделювання – це систематичний підхід до доменної інженерії, розроблений Mark Simos і Organon Motives протягом останніх восьми років [23]. ODM базується на концепціях, які виникли на основі робіт багатьох дослідників і практиків в області "систематичного повторного використання програмного забезпечення".

ODM був розроблений з метою заповнення прогалів в методах і процесах доменної інженерії.

ODM – позначається як легко пристосовувана та настроювана модель процесу проектування, що використовується в багатьох організаціях і сферах діяльності, і легко інтегровувальна з різними процесами, методами й технологіями інженерії ПЗ [23]. ODM можна легко застосувати для планування й проектування проектів у контексті загальної програми повторного використання компонентів.

ODM розроблено для підтримки проектування й розробки домену для сфер діяльності, які є:

- зрілими (з наявним набором старих або існуючих систем);
- досить стабільними (варто розглядати хоча б деякі з існуючих систем);
- економічно життєздатними (планується створення нових систем).

ODM складається із трьох основних фаз, що називаються «планування домену», «моделювання домену» і «розробка компонент». Кожна фаза та її етапи представляються за допомогою графічних нотацій у вигляді діаграм IDEF0 (рис. 3). Термін «моделювання домену»

використовується по відношенню до набуття та представлення знань домену.

На етапі планування домену проводиться визначення і опис границь досліджуваного домену для відокремлення від інших (можливо

також зв'язаних) доменів. Цей етап складається із наступних основних послідовних під-етапів:

- визначення цілей моделювання;
- визначення границь доменної області;
- визначення домену.

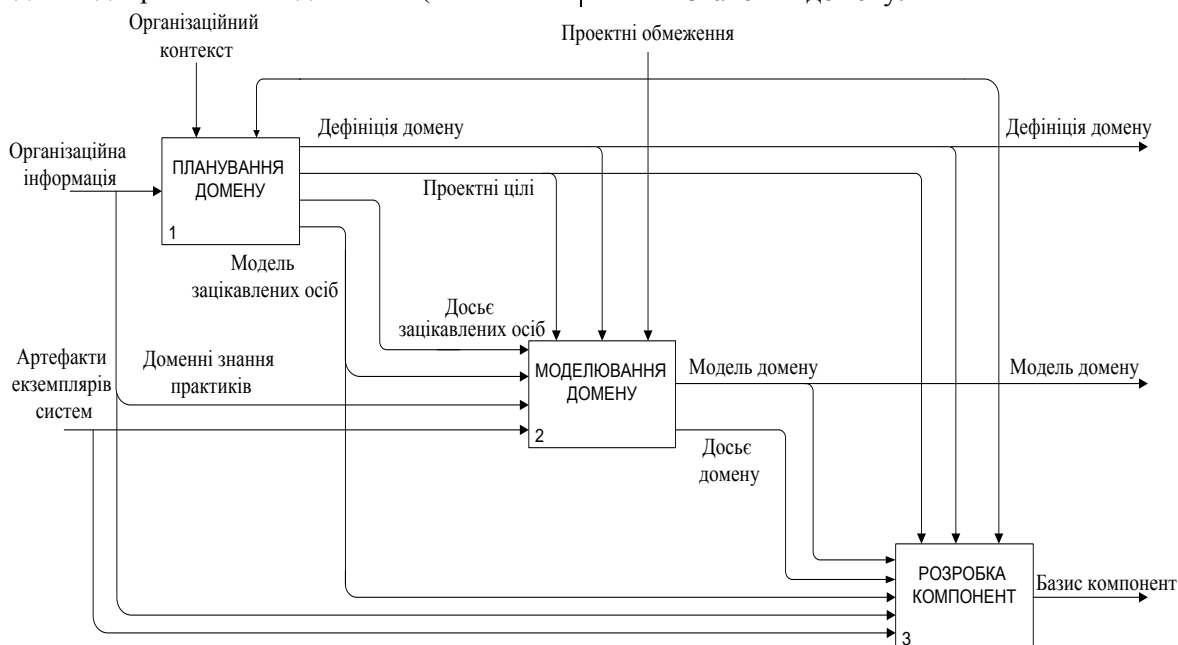


Рис. 3. Проведення доменної інженерії. Діаграма IDEF0

Метою під-етапу «визначення цілей моделювання» є виділення цілей, стосовно яких проводиться моделювання, враховуючи інтереси всіх зацікавлених сторін. На цьому етапі відбувається побудова діаграми зв'язків між зацікавленими сторонами і виділеними цілями моделювання.

Результатом етапу є визначення цілей, зацікавлених сторін, зв'язків між ними. Обидва набори сутностей вносяться до моделі зацікавлених сторін проекту, що служить первинним базисом для встановлення конкретних і досяжних цілей моделювання повторно-використовуваного рішення.

Важливим моментом на цьому етапі є визначення не посередніх, а реальних цілей дослідження.

Призначенням під-етапу «визначення границь доменної області» є визначення границь досліджуваного домену, а також виділення зв'язаних сутностей, їх відношення до досліджуваного домену і навпаки. Межі домену визначаються таким чином, щоб вони відповідали і не суперечили цілям, виділеним на попередньому етапі. Обраний домен називається сфокусованим доменом.

Також, на даному етапі виділяється організаційний контекст – набір доменів, пов'язаних з досліджуваним доменом і між со-

бою. Результатом цього етапу є список визначених досліджуваних доменів, що підтверджується проектними цілями.

Мета під-етапу «визначення домену дослідження» полягає в більш формальному та вузькому визначенні границь домену дослідження. Границі повинні відповідати стратегічним і концептуальним критеріям, виділеним раніше. Практично, цей етап є етапом верифікації попередніх результатів.

Аналітик, маючи вже більш конкретне подання про домен дослідження, може спробувати зробити начерки доменної моделі, що б зрозуміти, чи відповідає вибраний підхід цілям, визначеним на ранніх етапах дослідження. Якісна границя домену вважається границя з характеристиками: ясність, чіткість, здатність до розвитку та новизна.

Результатом цього етапу є визначення домену, що описує границі домену дослідження щодо інших зв'язаних доменів, які надалі, в процесі роботи, постійно оновлюються і вдосконалюються.

Наступна фаза – фаза «моделювання домену». Складові цієї фази наступні:

- накопичення інформації по домену;
- розробка описових моделей домену;
- деталізація моделі домену.

Під-етап «накопичення інформації по домену» характеризується найбільшим залученням третіх осіб у процес ДА.

Систематичний збір інформації, на якій буде побудована доменна модель, виконується з використанням різноманітних джерел інформації. Здебільшого аналізується документація, проводяться анкетування та опитування інформаторів (експертів доменної області).

Результатом цього етапу є складання досє домену - запис всієї зібраної інформації, що є базисом для наступного кроку моделювання.

На етапі «розробки описових моделей домену» інформація, зібрана на попередніх етапах, обробляється з метою побудови формальних моделей, що виділяють спільність (загальне) і варіативність (відмінне) серед представлених систем домену. На цьому кроці розробляється ряд незв'язаних моделей, що описують різні системи і області, що вивчаються доменом.

Також, результатом є доменна мова, що застосовується до опису складових домену, їх зв'язків та поведінки.

Порівнюючи з природною мовою, словникові терміни визначають термінологію мови, концепції в домені складають семантику мови, а властивості співвідносяться з самими реченнями в мові домену.

Під-етап «деталізація моделі домену». Використання результатів попереднього етапу – набори індивідуальних моделей, які, будучи об'єднаними, надають описову модель для загальних і варіативних характеристик у межах обраних репрезентативних систем домену та надають всю необхідну інформацію для виділення списку загальних і варіативних компонентів. Основними цілями під-етапу є:

- перевірка моделей, створених раніше;
- розробка теорії, або моделі, що пояснює обмеження або інші особливості репрезентативних систем домену;
- визначення зв'язків між функціями домену, спільного і відмінного.

Кінцевим результатом етапу моделювання домену є модель домену та доменна мова.

Третя фаза ODM – «розробка базису компонент» складається з наступних під-етапів:

- визначення базису компонент;
- проектування базису компонент;
- реалізація базису компонент.

Визначення базису повторно-використовуваних компонентів на основі описових і функціональних моделей проводиться з метою отримання набору основних компонентів, необхідних для функціонування систем домену.

Проектування базису компонентів направлена на створення моделі компонентів, що визначає набір характеристик, можливих для визначеного домену чи системи. Також повинні бути визначені зовнішні та внутрішні обмеження компонентів.

Реалізація базису компонентів. Результатом цього етапу є готовий набір сутностей – доменної моделі, доменної мови, базису компонентів і їх реалізації.

Результати застосування ODM до інтерфейсу пульта інструктора АТ

Вхідними даними для процесу «планування домену» є організаційна інформація, що несе в собі дані про замовників, зацікавлених осіб, які будуть брати участь у проекті. Основними джерелами організаційної інформації є наступні: неформальні знання учасників, інформаційні брошури, звіти. На управління процесом впливає організаційний контекст та проектні обмеження.

Поверхневий аналіз існуючої організаційної інформації дає можливість визначити домен.

В даному контексті доменом є розробка авіаційного тренажера, що представляє собою частину пілотажно-навчального комплексу «літак-тренажер», який застосовується для навчання пілотів і представляє собою складну апаратно-програмну модельючу систему, що імітує кабінку літака з органами управління та індикаторами, візуальне і звукове оточення пілотів, і поведінку літака при різноманітних фазах польоту [24].

Одним із невід'ємних компонентів авіаційного тренажера є пульт інструктора, який представляє собою інструмент для навчання пілотів (рис. 4). За допомогою пульта здійснюється установка початкових умов польоту, контроль якості пілотування, імітація взаємодії пілотів з диспетчером, управління імітацією відмов літака. Для оцінки якості пілотування і розбору польотів інструктор отримує значення відповідних параметрів з фіксацією їх на паперовому чи електронному носіях [25].

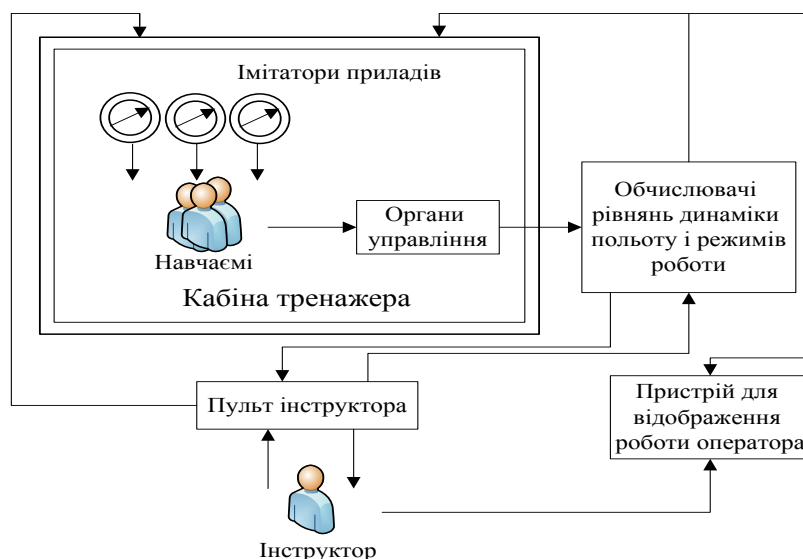


Рис. 4. Структурна схема АТ

Виходячи із загального аналізу процесу навчання пілотів, виділяють основні аспекти частини прикладного домену «авіаційний тренажер» - пульта інструктора. До них відносяться [26]:

- інтерфейс пульта через який інструктор отримує інформацію про політ і здійснює управління тренажером, називається користувацьким інтерфейсом інструктора. Інтерфейс інструктора становить безліч інтерфейсних елементів (елементи для відображення інформації та для управління процесом навчання), які призначені для полегшення роботи інструктора;

- процеси управління і контролю навчанням включаються в заданні умов перед початком польоту і їх зміни під час його здійснення. Згідно з аналізом, задання і зміна умов польоту повинні здійснюватись у відповідності із сценарієм – програмою (схемою) польоту, яка розроблена для досягнення визначених задач підготовки (штатний зліт і посадка, пілотування в складних метеоумовах). Зміна умов польоту відбувається при настанні подій (виконанні літаком визначеного маневру, досягнення заданої висоти, часу польоту), які можуть ідентифікуватися за сукупністю значень заданих параметрів.

- інформаційний обмін забезпечує функціонування пульта інструктора АТ як частини розподіленого інформаційно-моделюючого комплексу, що отримує і відправляє інформацію через обчислювальну мережу.

Після аналізу організаційного контексту виділено наступний список потенційних субдоменів дослідження:

- «інтерфейс пульта інструктора»;
- «процеси управління і контролю польотів»;
- «процеси інформаційного обміну».

За критеріями відбору: зрілість, майбутня окупність щодо результатів дослідження, стабільність технології, доступність стандартизованих знань у домені, досвід у домені аналітичної групи було виділено сфокусований субдомен «інтерфейс пульта інструктора»

Результатами даної фази, крім виділення сфокусованого домену, є досє зацікавлених сторін та цілі проекту.

Досє зацікавлених осіб/сторін – документ, що містить в собі інформацію про осіб/сторони, що зацікавлені в даному проекті. Стосовно даної роботи, зацікавленими особами/сторонами виступають:

- організація-замовник – освітня державна організація, зокрема, факультет комп’ютерних наук;
- проектна команда – автор статті;
- інформатори по дослідженню – особи, що надають інформацію стосовно домену дослідження (декан факультету, викладачі, працівники тренажерного центру);
- споживачі результатів – особи, організації, що будуть використовувати результатами дослідження (співробітники факультету комп’ютерних наук).

Інформація із досє зацікавлених осіб/сторін інтегрується в модель зацікавлених осіб. Модель зацікавлених осіб/сторін – сукупність інформації про ролі та обов’язки зацікавлених осіб/сторін, та про їх відношення.

Основні цілі проекту при використанні ДА наступні:

- обґрунтування можливості доказового формування повторно-використовуваних рішень при створенні інтерфейсу пульта інструктора АТ, шляхом використання методів ДА;
- формування повторно використовуваних рішень, шляхом ДА;
- виділення списку компонент, що складають інтерфейс пульта інструктора АТ;
- класифікація доменних сутностей;
- розробка бібліотеки класів;

- розробка доменно-орієнтованої мови.

Процес досягнення цілі – формування повторно-використовуваних рішень представлено на діаграмі діяльності (рис. 5). На діаграмі відображається послідовність виконання визначених дій, які в сукупності приводять до отримання належного результату – формування повторно використовуваних рішень та передачі їх організації-замовнику.

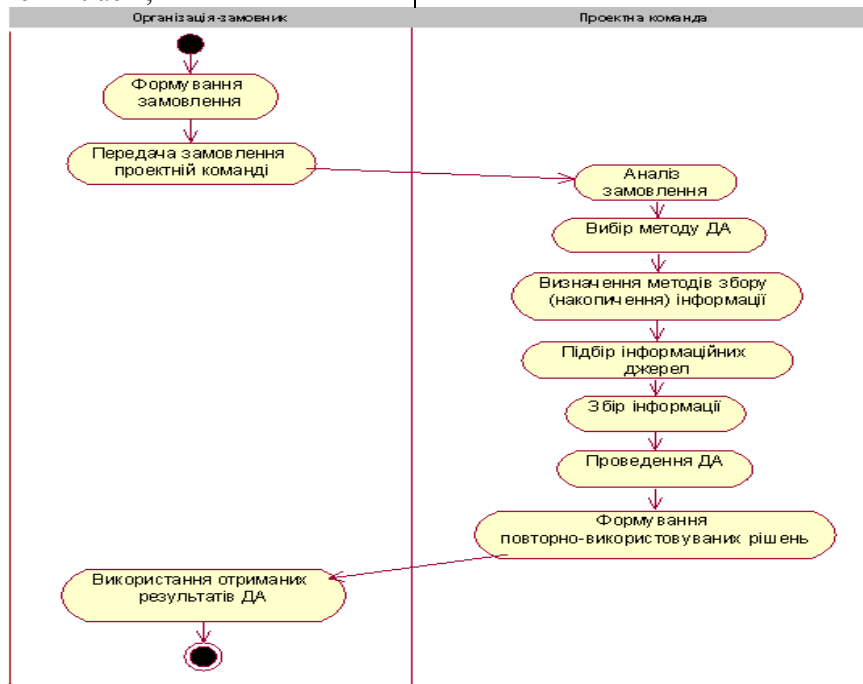


Рис. 5. Відношення між зацікавленими сторонами

Процес «модельовання домену» починається з визначення методів збору (накопичення) інформації про домен. Серед методів збору (накопичення) даних виділяють наступні [27]:

- інтерв'ю замовника та експертів прикладного домену;
- анкетування;
- спостереження;

- вивчення документів та аналогічних систем;
- «мозковий штурм».

Збір інформації, на базі якої буде побудована доменна модель, здійснюється з використанням різноманітних джерел інформації [28-32]. Основні джерела в даному контексті представлені в таблиці:

Таблиця

№	Назва джерела	Опис	Представники
1.	Авіаційні спеціалісти	Особи, що володіють знаннями про складові пульта інструктора, їх конструкції	Інженери-конструктори авіаційної техніки, працівники тренажерного центру, інструктори, .
2.	Управляючі і регулюючі документи	Офіційні документи встановленої форми, що прийняті в межах компетенції уповноваженого органу (посадової особи), містять обов'язкові правила поведінки, що розраховані на визначене коло осіб і не однократне застосування	Конституція України, Повітряний кодекс України, документи ІСАО (Керівництво по запобіганню авіаційних подій, Керівництво по плануванню обслуговування повітряного руху), галузеві стандарти.
3.	Інформаційні документи	Документи, які призначені для оперативного ознайомлення з масивом опублікованих та неопублікованих джерел.	технічний опис літального апарату (АТ Л-410. Технічний опис. Книга 1. Основні характеристики тренажера), керівництво по льотній експлуатації.

4.	Існуючі пульти інструкторів	Робочі місця інструкторів у всіх розроблених тренажерах, що дозволяють проводити процес навчання.	Пульти тренажерів літаків Л-410, Ан-148, АН-24, Ту-154
----	-----------------------------	---	--

На основі аналізу джерел даних та їх вмісту формується доменна термінологія, що є основою доменного лексику.

Доменна модель представляється сукупністю систем домену та їх взаємозв'язків. Система в даному контексті – різновид інтерфейсного елементу.

Більшість інтерфейсів інструктора тренажерів оздоблені стандартним набором інтерфейсних елементів (елементи для відображення інформації та для управління процесом навчання), які призначені для полегшення роботи інструктора. Набір і вид інтерфейсних елементів індивідуальні для тренажера літака кожного типу, однак суттєво схожі за зовнішнім виглядом та складовими. Кожний інтерфейсний елемент визначається двома складовими $ie = (it, pf)$ – видом інформації – параметром it і формою її представлення pf . Вся множина інтерфейсних елементів домену $IE = \{ie_i \mid ie_i \in IE, i = 1..d\}$ визначається як множина двійок $ie = (it, pf)$.

Множина інтерфейсних елементів $IE = \{I \cup CIE\}$ представляється індикаторами – I та управляючими інтерфейсними елементами (органами управління) – CIE .

Індикатори $I \subseteq IE$ – здійснюють відображення інформації. До них відносяться індикатори швидкості, висоти, транспаранти станів, зображення карт.

Індикатори мають круглу або прямокутну форму, відповідно їх називають круговий індикатор (Circular Indicator (CI)) та прямокутний індикатор (Rectangular Indicator (RI)), тобто $I = \{CI \mid RI\}$.

Круговий індикатор (CI) представляється наступною описовою моделлю:

$$CI = \{CG \cup CS \cup TS \cup TM \cup TL \cup CR \cup P\},$$

де CG (Circular Gauge) – круговий прилад, CS (Circular Scale) – кругова шкала, TS (Tick Set) – множина величин, TM (Tick Mark) – мітка поділу шкали, TL (Tick Label) – підпис поділу шкали, CR (Circular Range) – кругова діапазонна мітка, P (Pointer) – стрілка.

Прямокутний індикатор (RI) представляє на пульти інструктора сигнальне табло. Табло складається із набору прозорих сигнальних індикаторів прямокутної форми, кожний із

яких характеризується назвою і вказує на настання певної події, тому

$$RI = \{RG \cup N\},$$

де RG (Rectangular Gauge) – прямокутний прилад; а N – назва індикатору.

Управляючі (керуючі) елементи $CIE \subseteq IE$ – елементи, які призначені для введення інформації, тобто для управління процесом навчання (перемикачі режимів, регулятори, кнопки введення значень і т.д.).

$$CIE = \{RG \cup N \cup SCG\},$$

де RG (Rectangular Gauge) – прямокутний прилад, N – назва індикатору, SCG (Small Circular Gauge) – малий круглий прилад.

Далі здійснюється деталізація складових приладу та виділення їх властивостей. Отримана інформація використовується для встановлення подібності та мінливості елементів інтерфейсу та для побудови мови домену.

Процес «розробка базису компонентів» проводиться з метою отримання основних компонентів (складових елементів), необхідних для функціонування зв'язаних систем домену (інтерфейсних елементів).

Дані компоненти можуть бути використані як повторно використовувані рішення при побудові візуального інтерфейсу пульта інструктора різних типів АТ. В процесі роботи здійснюється описове наповнення інтерфейсних елементів певними складовими.

Висновок

Використання ДА при створенні ПЗ інтерфейсу пульта інструктора АТ, зокрема, підходу ODM, дає можливість виділити відповідну інформацію домену, джерела інформації домену, доменну термінологію; побудувати доменну модель, яка є основою при розробці мову домену. Результуюча інформація, яка отримана в результаті застосування підходу ODM використовується в якості повторно використовуваних рішень при побудові інтерфейсів та елементів інтерфейсів різних типів АТ. Розробка та застосування повторно використовуваних рішень – є дієвим способом зниження витрат на реалізацію типових проектів та покращення їх якості.

Список літератури

1. Кулик Н.С., Харченко В.П., Луцкий М.Г., Кучер А.Г., Яновский Ф.И. и др. Энциклопедия безопасности авиации – К.: Техніка, 2008. – 1000 с. (Російською мовою)

2. Олейніков О. О. Активізація економічного розвитку авіаційної галузі України // InVenture Investment Portal , 14 жовтня 2010р
3. Корченко А.Г., Гордиенко В.И., Старохатный П.И., Карпенко С.В. Авиационная безопасность: Сборник международных документов. К: НАУ, 2001. – 123с.
4. Черняков Е. Некоторые тенденции развития авиатренажеростроения // «Аэрокосмический курьер». 2010 № 5 С. 36-37.
5. Basili V.R., Briand L.C., Melo W.L. How Reuse Influences Productivity in Object-Oriented Systems. // Communication of the ACM, October 1996/ vol.39, No 10, p.150-158
6. Gaffney J.E., Durek Jr., Durek T.A. Software reuse – key to Enhanced productivity: some quantitative models // Information and Software Technology, v.31, N.5, 1989, p.258-267.
7. Ivar Jacobson, Martin Griss, Patrik Jonsson. Software Reuse. Architecture, Process and Organization for Business Success. – Pearson Education Asia. – 2002. - 497p.
8. Rada R, James Moore J. Standardizing Reuse // Communication of ACM. March 1997/Vol, 40, No. 3, p.34-38.
9. Ray R.K., Levy H.M. A compositional model for Software Reuse // The computer Journal, v.32, N4, 1989, p.312-322.
10. Neighbors J.M. The Draco Approach to Constructing Software from Reusable Components // IEEE Trans. on Softw. Eng. – 1984. – 10, №3. P. 564-576.
11. Common Ada Missile Packages (CAMP), Technical Proposal, Vol. 1. Prepared by Raytheon Company, Missile Systems Division, Bedford, Mass. May 7, 1984.
12. R. McCain, “Reusable Software Component Construction: A Product-Oriented Paradigm”, IBM Federal Systems Division, Houston TX, 1986.
13. Сидоров Н.А. Восстановление, повторное использование и переработка программного обеспечения. I. // УСим – К. – 1998. – № 3. – С.74-83.
14. Prietto-Diaz R. Domain Analysis: An Introduction.- Software engineering Notes. – 1990. – V. 15, № 2. – p. 47–54.
15. Prietto-Diaz R. Domain Analysis For Reusability // COMPASC’87. – Tokyo, Japan. – 1987, Oct. 7-9. p. 23–29.
16. G. Arango. Domain Engineering for Software Reuse. Ph.D. Thesis, Department of Information and Computer Science, University of California, Irvine, 1988.
17. Kyo C. Kang, Sholom G. Cohen, James A. Hess. Feature-Oriented Domain Analysis (FODA) Feasibility Study // Software Engineering Institute. – 1990. – 161 p.
18. Предметно-ориентированное проектирование: структуризация сложных программных систем.: Пер. с англ. – М. : ООО «И.Д. Вильямс», 2011. – 448 с.
19. Arango, G. and R. Prieto-Díaz, “Part 1: Introduction and Overview - Domain Analysis Concepts and Research Directions,” Domain Analysis and Software Systems Modeling, Prieto-Díaz, R. and G. Arango (eds.), IEEE Computer Society, Los Alamitos, CA, 1991, p. 9-32.
20. G. Arango. Software Reusability, chapter 2. Domain analysis methods, p. 17-49. Workshops M.E.Horwood, London 1994.
21. X. Ferrés and Vegas. An Evaluation of Domain Analysis Methods. // In 4th CAiSE/IFIP8.1 International Workshop in Evaluation of Modeling Methods in Systems Analysis and Design, 1999, p. 1-13.
22. K. Czarnecki and U. Eisenecker. Generative Programming: Methods, Techniques, and Applications. Addison-Wesley, 2000. – 864 p.
23. Lockheed Martin Tactical Defense Systems. Organization Domain Modeling (ODM) Guidebook: Version 2.0 Manassas: STARS-VC-A025/001/00, 1996, 509 p.
24. 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.
25. Луцький М.Г., Рябокін Ю.М. Метод створення програмного забезпечення пульта інструктора АТ. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук пр. – К.: Век+, – 2009. № 51. С. 78–83.
26. Рябокін Ю.М. Формування повторно використовуваних рішень при створенні програмного забезпечення пульта інструктора авіаційного тренажеру методом доменного аналізу// Науковий журнал «Інженерія програмного забезпечення». – К. – 2010. – № 2. – С. 46-55.
27. Вигерс Карл. Разработка требований к программному обеспечению/Пер, с англ. — М.: Издательский-торговый дом «Русская Редакция», 2004. —576с.
28. Конституція України. – К. – 2009р. – 51с.
29. Повітряний кодекс України. Розділ 5. – К. – 1993. – С. 10-11.
30. Документи ІКАО. Керівництво по аеропортовим службам. Керівництво по запобіганню авіаційних подій. Керівництво по плануванню обслуговування повітряного руху.
31. Авіаційний тренажер ТЛ-410. Технічний опис. Книга 1. – 1992 р.
32. Керівництво по льотній експлуатації ТЛ-410. – 1986. – 316с.

Відомості про автора



Рябокін Юлія Миколаївна - старший викладач кафедри інженерії програмного забезпечення факультету комп'ютерних наук Національного авіаційного університету.

Наукові інтереси: технології створення програмного забезпечення, шаблонне проектування, доменний аналіз.

Тел. 406-76-41

E-mail: yulia.ryabokin@livenau.net

yulia_r@rambler.ru

Стаття надійшла до редакції 24.01.2011 року