

ПОВТОРНО ВИКОРИСТОВУВАНІ РІШЕННЯ ПРИ СТВОРЕННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРФЕЙСУ ПУЛЬТА ІНСТРУКТОРА АВІАЦІЙНОГО ТРЕНАЖЕРУ

Запропоновано підхід до формування повторно використовуваних рішень при створенні програмного забезпечення інтерфейсу пульта інструктора авіаційного тренажеру, що ґрунтується на використанні доменного аналізу. В якості методу доменного аналізу використовується Organization domain model.

Ключові слова: авіаційний тренажер, пульт інструктора авіаційного тренажеру, інтерфейс пульта інструктора, доменний аналіз, модель домену, повторно використовувані рішення.

Вступ. Сучасна авіаційна техніка енергонасичена, з високим рівнем автоматизації бортових систем і інтелектуальним бортовим обчислювальним комплексом, тому процес опанування експлуатації такої техніки неможливий без проходження тренажерної підготовки. Використання авіаційних тренажерів для здійснення підготовки пілотів дає можливість отримати їм відповідні навички управління літаком в наземних умовах. На тренажері під час тренування можна відтворити аварійні ситуації літака, які в реальних умовах польоту реалізувати не уявляється можливим із міркувань безпеки [1].

Одним із основних компонентів авіаційного тренажеру є пульт інструктора, що відображає інформацію про політ та стан тренажеру. Існуючі пульти більшості успадкованих тренажерів реалізовані апаратно, морально застаріли та фізично зношені в процесі тривалої експлуатації. Це є причиною заміни апаратного пульта інструктора на нові апаратно-програмні засоби на основі сучасних комп'ютерних технологій [2].

На жаль, в Україні, на даний час відсутні спеціалізовані підприємства, діяльність яких була б орієнтована на розробку пультів інструкторів та самих авіаційних тренажерів. Для виходу із ситуації, що склалася, все частіше використовуються апаратно-програмні засоби для побудови пультів інструкторів. В якості апаратного забезпечення використовуються персональні або промислові комп'ютери загального призначення. Програмне забезпечення включає системне програмне забезпечення і прикладне програмне забезпечення, яке реалізує функціональність.

Постановка задачі. Програмне забезпечення пульта інструктора авіаційного тренажеру є спеціалізованим доменом, що характеризується об'ємністю та складністю. При розробці такого програмного забезпечення виникає потреба у витраті значних ресурсів (часу, фінансів), що є неприйнятним в умовах світової економічної кризи та недостатнього фінансування розробок в авіаційній галузі. Тому питання зниження витрат на розробку програмного забезпечення пульта інструктора є актуальним.

Одним із способів вирішення цієї проблеми є повторне використання при створенні програмного забезпечення. Тобто, використання раніше отриманих рішень-результатів: ідей, знань, вимог, результатів аналізу, проектування, тестування, документування та коду під час розробки нового програмного забезпечення.

Для формування повторно використовуваних рішень в роботі пропонується використати метод доменного аналізу, який розглядається в контексті домену програмне забезпечення інтерфейсу пульта інструктора.

Аналіз останніх досліджень та публікацій. Принцип повторного використання як шлях підвищення продуктивності і зменшення витрат на розробку програмного забезпечення почав досліджуватися в 80-х роках та продовжується і нині. Сутність його полягає в зменшенні обсягу новоствореного програмного забезпечення за рахунок застосування компонентів існуючого програмного забезпечення.

Питання повторного використання при розробці програмного забезпечення досліджували багато вітчизняних та закордонних вчених: І. Вельбицький, М. Сидоров, R. Prieto-Diaz, I. Sommerville, V. Basilini та інші [3].

Незважаючи на велику кількість наявних компонент, що були доступні для повторного використання, більшість розробників надавали перевагу створенню власних

компонент. Мотивуючи це тим, що існуючі компоненти були важкими для розуміння та адаптування до нових розробок.

Для виходу із ситуації, що склалася, було запропоновано використовувати доменний аналіз при створенні повторно використовуваних компонент.

Вперше термін «доменний аналіз» ввів *Neighbors* [4]. Він використовував доменний аналіз для позначення дій, що забезпечують виявлення об'єктів, операцій та відношень в домені з метою їх повторного використання при розробці нових застосувань в даному та інших доменах.

Дослідження в даному напрямленні продовжили *McCain* [5] та М. Сидоров [6]. Вони розглядали доменний аналіз як частину процесу розробки програмного забезпечення та запропонували ряд рекомендацій для здійснення доменного аналізу – розбиття процесу на три основні етапи, які багатократно повторюються для різних компонент: 1) ідентифікація об'єктів багатократного використання, 2) абстрагування та узагальнення, 3) класифікація та каталогізація для багатократного використання.

Базуючись на зазначених дослідженнях *Prietto-Diaz* [7] запропонував впровадити доменний аналіз в процес розробки програмного забезпечення, в якому продукти доменного аналізу безперервно переглядаються і уточнюються при створенні нових систем; визначив конкретні процеси і проміжкові продукти, які передають розуміння процесу доменного аналізу; визначив входи і виходи процесу доменного аналізу; описав процес доменного аналізу колекцією діаграм потоків даних, що відображають послідовність виконання доменного аналізу визначеної предметної області, та проміжкові результати процесів.

В своїх дослідженнях *Arango* [8] запропонував метод доменного аналізу в якому розглядав повторне використання як систему набуття знань. Він акцентував увагу на доменному аналізі як на безперервному процесі створення і збереження інфраструктури багатократного використання.

Протягом останніх десяти років увага акцентується на двох методах доменного аналізу [9] – *Feature-oriented domain analysis* та *Organization domain model*, оскільки вони є більш зрілими і краще задокументованими. В роботі пропонується скористатися останнім.

Виклад основного матеріалу дослідження. *Organization domain model (ODM)* – метод, що розробив *Mark Simos* з метою систематизування ключових аспектів процесу доменного моделювання і забезпечення повного каркасу ЖЦ доменної інженерії [10]. Метод *ODM* корисний для різних організацій та доменів і піддається інтеграції з різноманітними процесами програмної інженерії, методами і технологіями реалізації.

ODM складається із трьох основних процесів: визначення домену; моделювання домену; інженерії бази компонент.

Кожний процес *ODM* характеризується певними діями, що переводять вхідні дані у вихідні. Перші два процеси *ODM* відповідають процесу доменного аналізу.

Для здійснення проведення доменного аналізу за методом *ODM* необхідно визначити первинну вхідну інформацію стосовно організації, яка буде проводити дослідження. До неї відноситься інформація наступної категорії: організаційний контекст; проектна декларація; проектні обмеження; список замовників.

Організаційний контекст представляється організацією, яка проводить дослідження. Стосовно даної роботи, організацією виступає факультет комп'ютерних наук Національного авіаційного університету.

Проектна декларація – документ, що містить інформацію про замовників та виконавців проекту [10]. В роботі замовником і виконавцем виступає факультет комп'ютерних наук, в особі працівників факультету. Замовників представляють працівники тренажерного центру, а виконавців – викладачі факультету.

Проектні обмеження накладаються вітчизняними та закордонними стандартами, апаратним забезпеченням, термінами та бюджетом проекту.

Після визначення вхідної інформації відбувається перехід до виконання першого процесу доменного аналізу – визначення домену. Основна мета процесу полягає у аналізі

контексту досліджуваної проблеми. Виходячи з аналізу, формуються цілі дослідження, список зацікавлених сторін, визначаються межі домену, створюється модель зацікавлених сторін та визначається та уточнюється сам домен дослідження.

В рамках даної роботи список зацікавлених сторін представляється групами учасників: організація замовник – факультет комп'ютерних наук;

2) проектна команда – автор статті;

3) інформатори по дослідженню – працівники тренажерного центру Національного авіаційного університету та авіаційного науково-технічного комплексу «АНТОНОВ», інструктори;

3) споживачі результатів – факультет комп'ютерних наук.

Досье зацікавлених сторін представляється зазначеними групами учасників, розділених за пріоритетом з інформацією про їх інтереси в рамках дослідження. Модель зацікавлених сторін відображає зв'язки та ролі зацікавлених сторін.

Основна мета дослідження – формування повторно використовуваних рішень.

Ґрунтуючись на інтересах зацікавлених сторін, цілях дослідження та враховуючи потенціал для повторного використання визначається домен дослідження. Домен представляється набором систем, що йому належать – екземпляри систем. Екземпляри систем, в контексті даного домену представляються двома групами: індикаторами та елементами управління. До індикаторів відносять: авіагоризонт, варіометр, висотомір, сигнальне табло; до елементів управління – панель відмов та тумблери.

Після визначення домену дослідження та його складових здійснюється перехід до виконання другого процесу доменного аналізу – моделювання домену. На етапі виконання даного процесу формуються досье домену, доменна термінологія та створюється модель домену.

Основними під процесами моделювання домену є: накопичення доменної інформації, опис домену та уточнення моделі домену.

На етапі накопичення інформації здійснюється визначення методів збору, джерел інформації та, власне, збір інформації. Серед основних методів збору інформації виділяють: інтерв'ю замовників та інформаторів домену; анкетування; спостереження, вивчення документів та аналогічних систем. Основними джерелами інформації виступають авіаційні спеціалісти, працівники тренажерного центру, управляючі і регулюючі документи, інформаційні документи, існуючі пульти інструкторів. Результат під процесу є досье домену, що представляє собою документ, який є покажчиком на зібрану інформацію. В ньому зазначається, яка інформація була отримана, звідки і яким чином.

Під час виконання під процесу опису домену, ґрунтуючись на інформації, яка була відібрана на попередньому етапі, створюються описові моделі екземплярів систем домену, словник домену та таксономія елементів інтерфейсу пульта інструктора.

Описові моделі описують індикатори та елементи управління, фізику їх роботи та складові. Словник домену об'єднує терміни взяті із домену, анотовані з метою відображення інформації про кожний термін з доменно-орієнтованим змістом. Таксономія є ієрархічною класифікацією основних понять домену. Вона допомагає зрозуміти послідовність та організацію елементів інтерфейсу [11].

Під процес уточнення моделі домену характеризується інтеграцією та інтерпретацією описових моделей. Результатом даної фази виступає модель домену, яка відображає спільність та відмінність систем в домені.

Для створення моделі домену необхідно виявити важливі концепти/концептуальні класи домену їх зв'язки та властивості. Існує два способи виявлення концептуальних класів [12]: з використанням списку категорій концептуальних класів та на основі виділення іменників. В роботі пропонується поєднати зазначені методи для виявлення концептуальних класів інтерфейсу пульта інструктора. Основними концептуальними класами домену виступають: індикатор, елемент управління, панель відмов, параметр, стрілка, шкала та інші.

На основі словесного аналізу опису домену можна виділити зв'язки (асоціації) між концептуальними класами домену, що згодом будуть додані до моделі домену. Вони сприяють кращому розумінню моделі домену.

Використовуючи інформацію отриману на етапі опису домену виділяємо властивості, що характеризують концептуальні класи домену – будуємо модель домену (рис. 2).

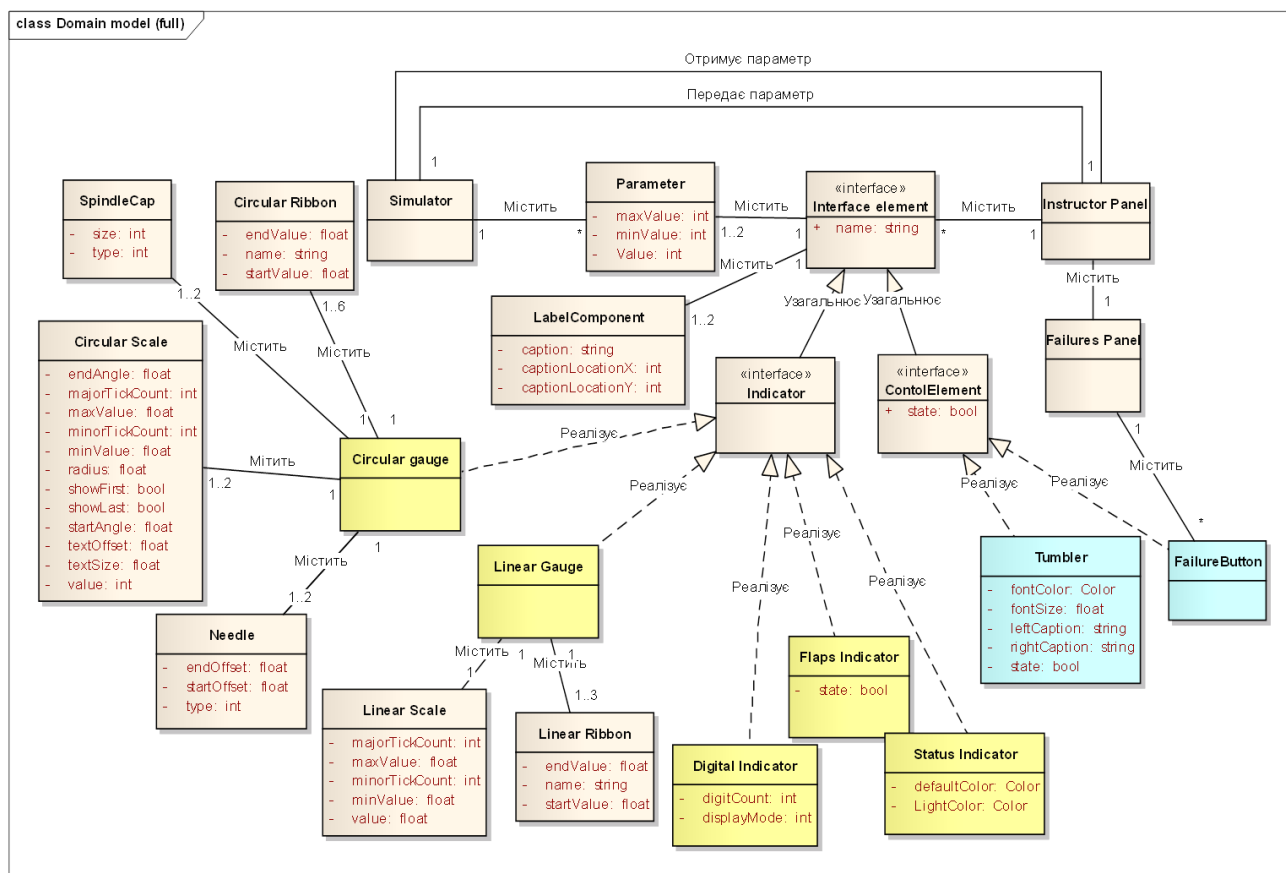


Рис. 2. Модель домену – інтерфейсу пульта інструктора

Вона відображає спільність та відмінність систем в домені. Модель домену може багатократно використовуватися в якості джерела знань домену для визначення та створення нових систем в домені та як основа для розробки програмних об'єктів.

Висновки. Отримані під час проведення доменного аналізу результати: словник домену, описові моделі екземплярів систем домену, таксономія інтерфейсних елементів та модель домену виступають в ролі повторно використовуваних рішень, що можуть використовуватися при побудові інтерфейсів пультів інструкторів різних типів авіаційних тренажерів. Що є дієвим способом зниження витрат на розробку типових проектів.

ЛІТЕРАТУРА

1. Ancker G. Design of a flight simulator software architecture / Göran Ancker, Jan Wallenberg. – School of Mathematics and System Engineering, Växjö University. – 2002. – 91 p.
2. Сидоров Н.А. Реинженерия программного обеспечения информационно-моделирующих тренажерных комплексов / Сидоров Н.А., Недоводев В.Т., Хоменко В.А., Сердюк И.П. // Управляющие системы и машины. – К. – 2008. – № 4. – С. 68-74.
3. Рябокін Ю.М. Доменний аналіз при створенні програмного забезпечення інтерфейсу пульта інструктора авіаційного тренажеру / Ю. М. Рябокін // Інженерія програмного забезпечення, 2011. – № 1 (5). – С. 35-45.
4. Neighbors J.M. The Draco Approach to Constructing Software from Reusable Components // IEEE Trans. on Softw. Eng. – 1984. – № 3. P. 564-576.
5. McCain R. Reusable Software Component Construction: A Product-Oriented Paradigm // IBM Federal Systems Division, Houston TX, 1986.
6. Сидоров Н.А. Повторное использование программного обеспечения // Кибернетика. – 1989. - № 3 – С. 46-51.

7. Wartik S., Prieto-Diaz R. Criteria for comparing reuse-oriented domain analysis approaches. – Software productivity consortium, 1991. P. 31 – 67.
8. Arango G.. Software Reusability, chapter 2. Domain analysis methods, Workshops M.E.Horwood, London 1994. P. 17 – 49
9. Alana E., Rodriguez A. Domain engineering methodologies survey. – GMV AEROSPACE AND DEFENCE S.A., Madrid, 2007, P. 1 –38.
10. Lockheed Martin Tactical Defense Systems. Organization Domain Modelling Guidebook: Version 2.0 Manassas STARS-VC-A025/001/00, 1996, 509 p.
11. Рябокін Ю.М. Генератор елементів інтерфейсу пульта інструктора авіаційного тренажера / Ю. М. Рябокін // Проблеми інформатизації та управління, 2012. – № 3 (39). – С. 130-134.
12. Ларман К. Применение UML и шаблонов проектирования. 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 624 с.

Надійшла: 24.10.2012 р.

Рецензент: д.т.н., професор Литвиненко О.Є.

УДК 378.1:004:303.732.4(477)

Мендзєбровський І.Б., Сидоров Є.М., Дуднік В.В.

РОЗШИРЕННЯ RTPA ДЛЯ ПРЕДСТАВЛЕННЯ СПЕЦИФІКАЦІЇ ДОМЕННОГО АНАЛІЗУ

Розглянуто задачу автоматизації виконання процесів доменного аналізу в розробці програмного забезпечення. Згідно запропонованого методу побудови на основі формальних специфікацій засобів, що автоматизують доменний аналіз, наведено розширення RTPA.

Ключові слова: інженерія програмного забезпечення, доменна інженерія, доменний аналіз, захист інформації, формальні специфікації, RTPA.

Вступ. Доменний аналіз, як частина доменної інженерії застосовується в інженерії програмного забезпечення при створенні програмних систем шляхом повторного використання [1]. Для автоматизації процесів доменного аналізу створюються засоби [2]. Повний аналіз досліджень показує, що на ринку і у відкритому поширенні знаходяться інструменти, які спрямовані на реалізацію окремих аспектів доменного аналізу (аналіз, моделювання, декомпозиція) і нема засобів, які б охоплювали всі процеси доменного аналізу та налаштовували на метод доменного аналізу [3].

В роботах [3,4] запропоновані метазасоби, які шляхом використання формального опису процесів доменного аналізу будують інструментальні засоби для їх виконання. Для формального опису специфікацій процесів доменного аналізу обрано Real Time Process Algebra (RTPA) [5], яка забезпечує всебічне представлення програмних систем. Однак, досвід застосування RTPA для опису засобів реалізації процесів доменного аналізу показав, що RTPA потребує розширення [4].

Постановка задачі. Створити розширення RTPA та перевірити його шляхом застосування при розробці програмного забезпечення авіаційних тренажерів.

Основна частина. Уводяться наступні розширення до RTPA [5]:

- в специфікацію Static Behaviors, на рівні специфікації класів, в опис процесу вводиться крім входів (I) і виходів (O), обмеження (R), які мають місце при описі процесів доменного аналізу [6];
- в схему процесу (Process Schema), яка ідентифікує процеси, вводиться опис середовища, що застосовується для виконання процесу (Environment – {E});
- для кожного середовища, кожного процесу, вводиться зв'язок середовища з виходом процесу - результатом (O), який можна отримати, застосовуючи це середовище (Environment Relations: EN ⇒ ON).