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 Mertin Tactical Defunse Systema. Organization Domain Modelling Guidebook: Version 2.0 Manassas STARS-VC-A025/001/00, 1996, 509 p.
11. Рябокінь Ю.М. Генератор елементів інтерфейсу пульта інструктора авіаційного тренажеру / Ю. М. Рябокінь // 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) **Мендзебровський І.Б., Сидоров Є.М., Дуднік В.В.**

РОЗШИРЕННЯ RTPA ДЛЯ ПРЕДСТАВЛЕННЯ СПЕЦІФІКАЦІЇ 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 $\Rightarrow$ON).

Розглянемо приклад з трьох процесів доменної інженерії методу ODM [6]. На рис. 1 наведено схему процесів доменного аналізу.

В специфікації застосовуються такі позначення:

Планування домену.

Вхідні данні: I1 - Організаційна інформація.

Вихідні данні: O1 - Визначення домену; O2 - Проектні цілі; O3 - Досьє зацікавлених сторін; O4 - Доменна модель зацікавлених сторін

Середовища: E1 – MS Word; E2 – MS Word; E3 – MS Excel; E4 – MS Visio.

Обмеження: R1 - Організаційний контекст; R2 - Проектні обмеження; Деталізація процесу; DP1 – Набор цілей; DP2 – Визначення домену; DP3 – Область домену.

Моделювання домену.

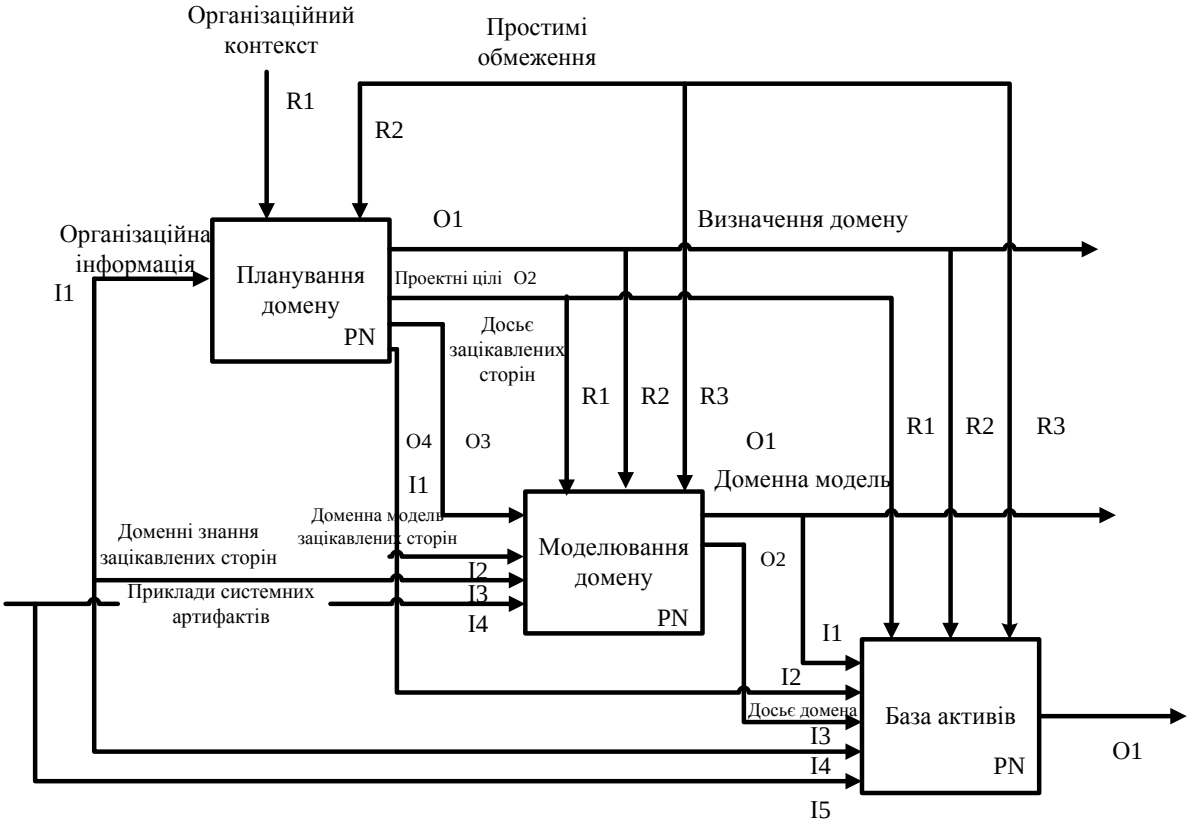

Рис. 1. Процеси доменного аналізу

Вхідні данні: I1 – Досьє зацікавлених сторін; I2 – Доменна модель зацікавлених сторін; I3 – Доменні знання зацікавлених сторін; I4 – Приклад системних артефактів.

Вихідні данні: O1 – Доменна модель; O2 – Досьє домену.

Середовища: E1 – MS Visio; E2 – MS Word.

Обмеження: R1 – Визначення домену; R2 – Проектні обмеження; R3 – Проектні цілі.

Деталізація процесу: DP1 – Визначення границь домену; DP2 – Фокусування на домені; DP3 – Визначення розміщення домену.

База активів.

Вхідні данні: I1 – Доменна модель; I2 – Досьє домену; I3 – Доменна модель зацікавлених сторін; I4 – Приклад системних артефактів; I5 – Організаційна інформація.

Вихідні данні: O1 – База активів.

Середовища: E1 – MS Excel.

Обмеження: R1 – Визначення домену; R2 – Проектні обмеження; R3 – Проектні цілі.

Деталізація процесу: DP1 – Визначення базового набору активів; DP2 – Визначення архітектурного набору активів; DP3 – Реалізація набору активів.

Специфікація системи, яка виконує процеси доменного аналізу, що представлені на рис 1 буде мати наступний вигляд:

Static behavior

(Process Schema PLAN DOMAIN = PN 1
|| {ProcessID: PLAN DOMAIN ({I: I1};{O: O1, O2, O3, O4};{R:R1, R2})} 3{Environments: E1,E2,E3,E4}
|| {DetailedProcesses: DP1, DP2, DP3}
|| {EnvironmentRelations: O1 = E1|| E2,O2 = E2,O3=E3,O4 = E4}
)

(Process Schema MODEL DOMAIN = PN 2
|| {ProcessID: MODEL DOMAIN ({I:I1, I2, I3, I4};{O:O1, O2};{R:R1, R2, R3}3 {Environments: E1,E2}
)
}
|| {DetailedProcesses: DP1, DP2, DP3}

|| {EnvironmentRelations: O1 = E1|| E2,O2 = E2 }
)

(Process Schema ENGINEER ASSET BASE = PN 3
|| {ProcessID: ENGINEER ASSET BASE ({I: I1, I2, I3, I4, I5};{O: O1};{R:R1, R2, R3})}
|| {DetailedProcesses: DP1, DP2, DP3}
|| {Environments: E1}
|| {EnvironmentRelations: O1 = E1}
)

Dynamic behavior

Process Dispatch ≙ § ⇢
(@Event1 DOMAIN AHALISYS ⇢
{PN0:DP1,DP2,DP3})⇢ §

Діаграму залежності середовищ {E} на компонентах MS Office 2010 наведено на рис.2.

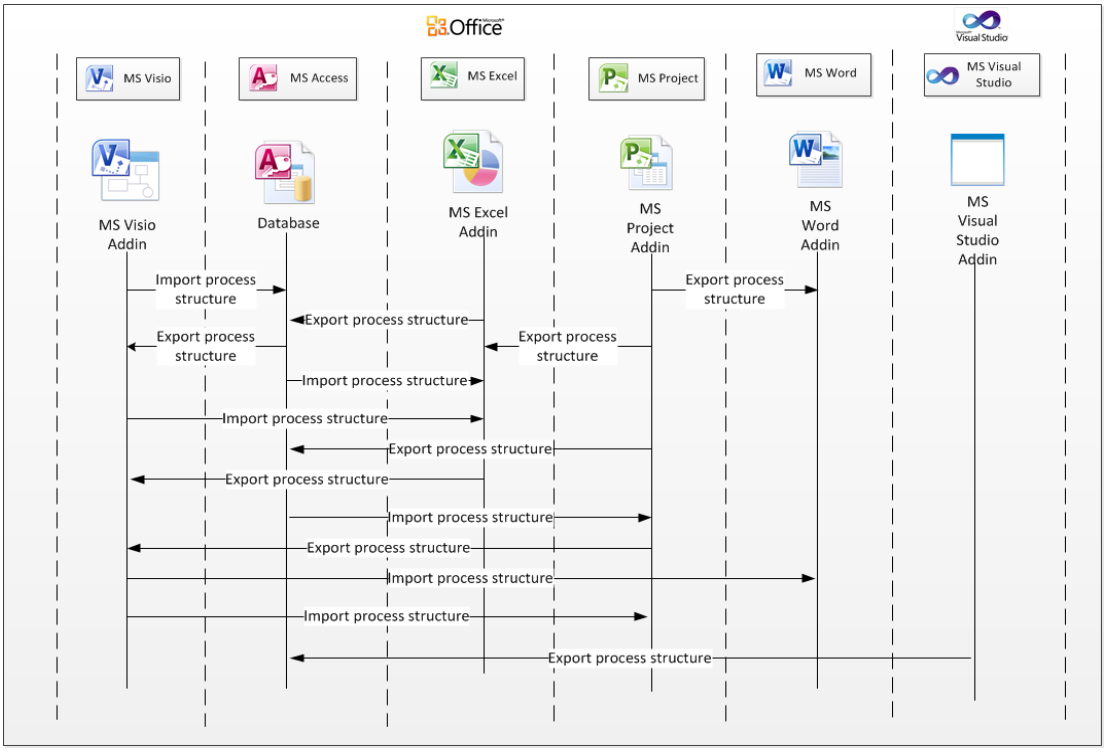

Рис. 2. Діаграми залежностей на компонентах MS Office 2010

Розглядається приклад застосування засобів для створення компонентів програмного забезпечення авіаційних тренажерів [7,8]. Досвід реінженерії програмного забезпечення в різних доменах показує, що значна частина труднощів, які виникають при реалізації процесів зворотної інженерії, і які обмежують або роблять повністю неможливою автоматизацію процесів пов'язана з тим, що, як правило, при розробці успадкованого програмного забезпечення розробники враховуючи особливості обладнання, для реалізації відомих моделей (з метою досягнення економічності або ефективності) часто застосовують власні прийоми. Через відсутність вимог документація, має низьку якість і, як правило не збігається з програмним забезпеченням.

В цілому ці особливості пов'язані з культурою розробки, існуючої в момент створення успадкованого програмного забезпечення і широко відомі. Однак, при реалізації процесів зворотної інженерії розпізнавання цих та інших особливостей у кожному конкретному випадку вимагає значних зусиль. Відсутність будь-якої класифікації цих особливостей ускладнює і робить затратною зворотню інженерію програмного забезпечення.

Шляхом доменного аналізу, на основі досвіду зворотної інженерії програмного забезпечення авіаційного тренажера, виявлено та описано ряд схем, які враховують вказані особливості, і можуть служити основою для проведення реінженерії програмного забезпечення подібного розглядався. Виявлено чотири схеми виконання реінженерії, які описують наступним сценарієм: $S : L \xrightarrow{c} N$, де S - позначення схеми реінженерії; L - успадковане програмне забезпечення; N - нове програмне забезпечення; С-умови, від яких залежить застосування схеми (табл.).

Таблиця 1

Схеми реінженерії програмного забезпечення

№	Назва схеми (S)	Умова застосування ©	Можливий характер реалізації
1	«Код-код»	Простий, лінійний код (послідовність операторів присвоювання).	Автоматично
2	«Код-алгоритм-код»	Код реалізує відому модель, алгоритм реалізації відомий.	Автоматично
3	«Код-модель-алгоритм-код»	Код реалізує невідому модель(розробника), алгоритм незрозумілий, вимагає перевірки, опису моделі та алгоритму немає.	Напівавтоматично
4	«Код-модель-технічний опис-алгоритм-код»	Код реалізує невідому модель (розробника), алгоритм незрозумілий, вимагає перевірки, опис моделі і алгоритму є.	Напівавтоматично

Створено інструмент «Екстрактор - абстрактор» – зворотної інженерії, який призначений для аналізу вихідного коду на мові програмування SYPS платформи Robotron, і побудови його високорівневого алгоритмічного уявлення. Інструмент був спеціально створений для зворотної інженерії, і автоматизації процесу аналізу вихідних кодів на мові програмування SYPS і спрощення роботи програмістів при міграції коду на такі мови високого рівня як С, С++. Для розробки інструменту була обрана платформа Microsoft.Net і мова програмування С # (рис 3).

Рис 3. Загальна схема інструменту

Якщо алгоритм вихідного коду успадкованого програмного забезпечення простий, має лінійний характер (послідовність операторів присвоювання) і його розуміння не викликає сумнівів, то реінженерія може виконуватися за схемою «код - в - код».

За цією схемою була виконана реінженерії програми блоку «Початкові умови», а також програми розрахунків окремих параметрів блоку «Навігаційна система». Наприклад, програма блоку "Початкові умови" вирішує наступні завдання:

- встановлення початкових значень всім змінним програм моделювання динаміки польоту, силової установки, навігаційної системи;
- розрахунок кількості палива в паливних системах літака, початкової ваги літака, положення радіомаяків, висоти і координат аеродромів, включаючи дані отримані від пульта інструктора.

Використання сучасного обчислювача дозволило значно спростити програми блоку "Початкові умови" шляхом перевизначення функції дискретизації часу, обміну даними з пультом інструктора, кабіною екіпажу та іншими блоками. Зпрощення програми блока дозволило використовувати схему реінженерії «код - в - код». Наприклад, функція налаштування навчального значення всім змінним програм реалізована послідовністю команд SYPS і відповідною програмою в мові С:

Таблиця 2

Адреса пам'яті	**Код операції**	**Мітка**	**Мнемоніка операції**	**Операнд**
01410	01000	VNGP	KOP	
01411	00020332		LDA	LG13
01412	00030111		UND	MASK+1
01413	100040		UAR	
01414	00011420		SUN	*+4
01415	00020152		LDA	DM4
01416	00040766		SPA	DM
01417	00011422		SUN	*+3
01420	00020153		LDA	DM4+1
01421	00040766		SPA	DM
01422	00020423		LDA	PSL1
01423	10100312		SUS	<' 312>
01424	00040744		SPA	CPL1
01425	00020423		LDA	PSL1
01426	10100313		SUS	<' 313>
01427	00040741		SPA	SPL1
01430	00020424		LDA	PSL2
01431	10100312		SUS	<' 312>
01432	00040745		SPA	CPL2
01433	00020424		LDA	PSL2
01434	10100313		SUS	<' 313>
01435	00040742		SPA	SPL2

```
if ( fD72 )
        DM  = 0.0288f;
else
        DM  = 0.016f;
  CPL1  = (float)cos( degree2rad( PSL1, 180.0f ));
  CPL2  = (float)cos( degree2rad( PSL2, 180.0f ));
  SPL1  = (float)sin( degree2rad( PSL1, 180.0f ));
  SPL2  = (float)sin( degree2rad( PSL2, 180.0f ));
```

Висновки. Згідно методу побудови засобів, що автоматизують процеси доменного аналізу представленого розширення RTPA, яке демонструється на прикладі розробки програмного забезпечення авіаційного тренажера. Метод може використовуватися для побудови засобів, які автоматизують інші процеси доменної інженерії.

ЛІТЕРАТУРА

1. Сидоров Н.А. Восстановление, повторное использование и переработка программного обеспечения. I. // УСим – К.- 1998.- №3.- С.74-83.
2. Wartik S., Prieto-Diaz R. Criteria for comparing reuse-oriented domain analysis approaches.- Software productivity consortium, 1991. Hp. 31-67.
3. Dudnyk V., Ryabokin U., Mendzebrovskiy I. Method of domain analysis to information support of aircraft.- National Aviation University, Ukraine.- 2012.- p.p. 1.10.1-1.10.19
4. Сидоров Н.А., Мендзебровский И.Б., Рябоконь Ю.Н. Метод построения средств доменного анализа на основе формальных спецификаций в RTPA. Вестник Восточноукраинского национального университета им . Даля.- №4.- 2012.- С. 52-57.
5. Wang Y. Software engineering foundations: a software science perspective.- Auerbach Publications, 2008, p. 1200.
6. STARS. Organization Domain Modeling (ODM) Guidebook Version 2.0. STAR Technical Report D613-55159, STARS Technology Center, Arlington VA, July 1996.
7. Sidorov N., Khomenko V., Nedovodeev V. Reengineering of the air simulators legacy software.- Proc. Of the NAU.- №2.- 2008.- P. 28-35.
8. Сидоров Н.А., Недоводеев В.Т., Сердюк И.П., Хоменко В.А., Сидоров Е.Н. Реинженерия наследуемого программного обеспечения информационно – моделирующих тренажерных комплексов. УС и М.- 4. - 2008. - с.68-75.

Надійшла: 5.10.2012 р.

Рецензент: д.т.н., професор Петров О.С,

УДК 004.942.001.57 **Молодецька К.В.**

МОДЕЛЬ ТЕПЛОВОГО ПОЛЯ СМАРТ-КАРТИ МОДУЛЯ АУТЕНТИФІКАЦІЇ В КОМП'ЮТЕРИЗОВАНИХ СИСТЕМАХ

Запропоновано аналітичну модель теплового поля модуля смарт-карти пристрою аутентифікації, побудовану із використанням методу на основі алгебричних властивостей диференціальних спектрів. Виконано порівняння отриманих результатів із іншими відомими методами.

Ключові слова: захист інформації, теплове поле, модуль аутентифікації, диференціальні перетворення.

На сьогодні відома велика кількість різнопланових загроз безпеки інформації в комп'ютеризованих системах [1–2]. В останні роки на основі сучасних методів здійснюються нові види криптоаналітичних атак, метою яких є визначення деталей виконання криптографічних перетворень. До таких методів відносять атаки на комп'ютеризовані системи за допомогою вимірювання рівня енергоспоживання, випромінювання та використання інших побічних каналів. Такі атаки можуть здійснюватися проти різних алгоритмів шифрування з відкритим ключем в пристроях аутентифікації, що забезпечені системами протидії вторгненню [1]. Як приклад атаки на смарт-карту розглянемо випадок, коли порушник має на меті несанкціоноване отримання секретних ключів, які знаходяться всередині модуля, захищеного від вторгнення. Порушник не здатний до здійснення криптоаналізу алгоритмів чи протоколів і не може зламати систему протидії вторгненню.

Альтернативними варіантами атаки є таймінг-аналіз, який полягає у визначенні часу, необхідного модулю для виконання певних операцій і подальшого встановлення інформації про ключ, або вимірювання кількості тепла, що виділяється у модулі, та визначення місцезнаходження джерела випромінювання. Атаки на основі вимірювання випромінюваної теплової енергії були використані для розкриття секретів усіх карт, що є на ринку [1–2]. Теплове поле такого модуля описується диференціальним рівнянням з частинними похідними з початковими і граничними умовами. Моделювання теплових полів виконується