

УДК 004.415.2.045 (076.5)

В.В. Дудник

Метод доменного аналізу в інженерії програмного забезпечення

Національний авіаційний університет

Кафедра інженерії програмного забезпечення

Науковий керівник – Сидоров М.О., д-т.н., професор

Формальне представлення бізнес процесів використовуючи нотацію RTPA (Real-Rime Process Algebra), яка призначена для формалізації і опису бізнес-процесів. На основі формального представлення створюється графічне представлення бізнес процесів використовуючи IDEF0 нотацію. В такий спосіб була підвищена ефективність методології взаємодії в рамках «аналітик-спеціаліст». Цей інструмент повинен був забезпечити групову роботу над створенням моделі, з безпосередньою участю всіх аналітиків і спеціалістів, які беруть участь в рамках проекту. У зв'язку з тим, що в рамках проведення бізнес аналізу, беруть участь аналітики та спеціалісти, аналізуючи надані данні та створюючи вихідну документацію, цей процес трудомісткий. Інструмент повинен полегшити роботи бізнес аналітиків на стадіях аналізу та створення вихідної документації.

Формальное представление бизнес процессов используя нотацию RTPA (Real-Rime Process Algebra), которая предназначена для формализации и описания бизнес-процессов. На основе формального представления создается графическое представление бизнес процессов используя IDEF0 нотацию. Таким образом была повышена эффективность методологии взаимодействия в рамках «аналитик-специалист». Этот инструмент должен был обеспечить групповую работу над созданием модели, с непосредственным участием всех аналитиков и специалистов, участвующих в рамках проекта. В связи с тем, что в рамках проведения бизнес анализа, принимают участие аналитики и специалисты, анализируя предоставленные данные и создавая исходную документацию, этот процесс трудоемкий. Инструмент должен облегчить работы бизнес аналитиков на стадиях анализа и создания исходной документации.

Formal representation of business processes using the notation of RTPA (Real-Rime Process Algebra), which is designed to formalize and describe business processes. Based on the formal representation creates a graphical representation of business processes using the IDEF0 notation. In this way the methodology was improved efficiency of interaction within the "analyst-specialist." This tool should provide group work to create models, with direct participation of all the analysts and experts involved in the project. Due to the fact that in the framework of business analysis, participating analysts and experts, analyzing available data and creating original documentation, the process is laborious. The tool should facilitate the work of business analysts on the stages of analysis and creation of original documentation.

Ключові слова: доменний аналіз, повторне використання, доказове формування

1. Доменний аналіз

Програмне забезпечення повторного використання може бути поліпшено шляхом виявлення об'єктів та операцій для класу подібних систем, тобто для певного домену. Приклади доменів: бронювання авіаквитків системи, інструменти розробки програмного забезпечення, призначених для користувача інтерфейсів і фінансових додатків. Сфера області може бути вибрана довільно, широко, наприклад, банківська справа, або вузько, як просте редагування тексту. Зазвичай широкі області будуються на

декількох вузьких областях. Наприклад, домен призначений для інтерфейсу користувача можна розглядатися як область в системах бронювання авіакомпаній домену.

Загальні об'єкти та операції, ймовірно існують в декількох областях і, отже, є кандидатами в повторно використовні компоненти. Знайомство з областю, а також з її загальними і змінними частининами дає можливість для подальшого використання при розробці нових систем у цій галузі.

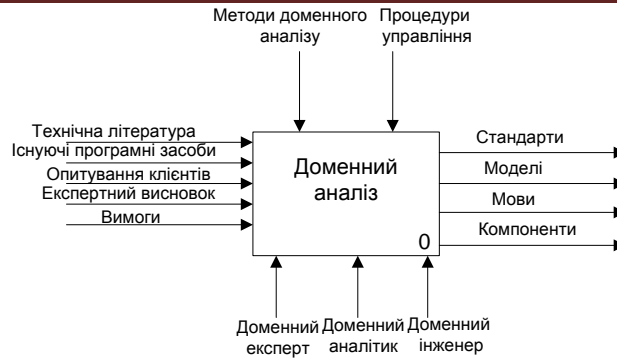


Рис.1. Контекст аналізу предметної області

Доменний аналіз підкреслює важливість проектування, а не код. Це робиться шляхом отримання архітектур з загальної моделі або спеціалізованої мови, які істотно збільшили потужність процесу розробки програмного забезпечення в конкретній області. Доменний аналіз можна розглядати як безперервний процес створення і підтримки інфраструктури в повторному використанні певного домену. Мал.1 показує, входи, виходи, управління і механізми аналізу предметної області. Вертикальний домен ідентифікує певний клас систем. Горизонтальний домен містить загальні частини програмного забезпечення, які використовуються в декількох вертикальних доменах. Прикладами горизонтального повторного використання математичних бібліотек функцій, контейнерних класів.

Доменний аналіз ідентифікує об'єкти і операції класу подібних систем в тій чи іншій предметній області. Типові види діяльності в галузі інженерних аналізу предметної області, розробка архітектури, багаторазове використання компонентів створення, відновлення і компонентів управління.

1.1. Етапи доменного аналізу

Доменний аналіз можна розглядати як збір досвіду і знань експертів. Домен знань містить-

ся інформація про те, які проблеми в одній області, розглядаються в програмних системах. Досвід і знання накопичуються, поки не досягнуть поріг, при якому абстракції можуть бути утворені і підготовлені до повторного використання. Переваги аналізу предметної області можна резюмувати наступним чином:

- повторне використання знань в предметній області;
- повторне використання компонентів в певному контексті, тобто предметно-орієнтованих компонентів;
- предметно-орієнтовані моделі для класифікації, зберігання та пошуку компонентів;
- рамки для інструментів і систем синтезу з компонентів багаторазового використання;
- крупнозернистих повторного використання в продуктах (в тому ж домені);
- визначення компонентів багаторазового використання програмного забезпечення.

Повторного використання знань в предметній області набуває все більшого значення.

Результати аналізу предметної області може бути використані не тільки для повторного використання, а й для навчання нових співробітників, надання їм загальної структури та функціонування систем в конкретній предметній області

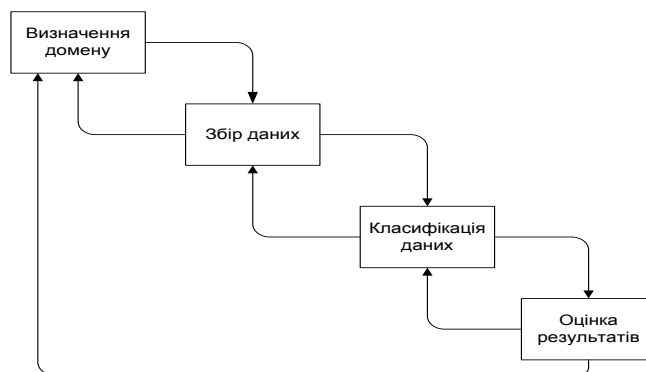


Рис.1.1. Етапи аналізу предметної області

2. Формалізоване представлення доменного аналізу

Розробка програмного забезпечення є унікальною дисципліною, в якій об'єкти їхнього дослідження вимагають нових форм математичного представлення, моделювання, опису, розробки, впровадження та супроводу програмних систем.

RTPA (Real-time process algebra) розглядається як цілісна система, яка для формального представлення інженерного рішення в програмному забезпеченні специфікації системи, уточнення і реалізації в режимі реального часу.

Тип системи, процес позначення, процес відносин, процес складання правил RTPA описані. Специфікація системи та вдосконалення методології RTPA і тематичних досліджень з проблем реального світу, які демонструють описовою силою RTPA як потужну систему інженерних позначення програмного забезпечення.

2.1 Real-time process algebra (RTPA)

Перший крок формалізованого представлення аналізу предметної області, це описання процесів доменного аналізу з використанням RTPA (Real-Time Процес алгебри) нотації.

Шаблон для цього опису, ви можете побачити в загальний характер для опису

(RTPA) являє собою набір формальних визначень та правил для опису алгебраїчних і в режимі реального часу відносини програмних процесів.

Процес є абстрактна модель програмного забезпечення, яке представляє перехід процедури системи з одного стану в інший, змінюючи значення його входів $\{I\}$, виходів $\{O\}$. RTPA розроблений як когерентна алгебраїчна системи інженерних позначень програмного забезпечення і є формальну методологію для вирішення інженерних проблем.

RTPA може бути використана для опису фізичних і логічних моделей системи. Системний процес R включає в себе алгебраїчні і реляційні операції:

$$R = \{ \rightarrow, \curvearrowright, |, | \dots | \dots, * R, R +, R, ||, \text{ff}, |||, \gg, \dots \}$$

Послідовність, позначається: $P \rightarrow Q$
Стрибок, позначається: $P \curvearrowright Q$

S	Name of process
N	Number of process
I	Input
O	Output
R	Require
C	Control
\curvearrowright	Jump
\rightarrow	Sequence

2.2 Опис процесів доменного аналізу використовуючи RTPA.

Static Behaviors (Process Schema $S = PN \ N \ || \ \text{Process ID } S \ (\{I\}; \{O\}; \{R\}; \{C\}) \ || \ \{\text{Detailed Processes}\}$)

Dynamic Behaviors Process Dispatch $\triangleq \S \rightarrow (@Event1S \rightarrow \{\text{ProcessSet1}\} \ | \ @Event2S \rightarrow \{\text{ProcessSet2}\} \ \dots \ | \ @Event \ n \ S \rightarrow \{\text{ProcessSet } n \}) \rightarrow \S$

Вхідні дані

- I1 Організаційна інформація
- I2 Системні артіфакти
- O1 Визначення домену
- O2 Доменна модель
- O3 Бази активів
- R1 Організаційний контекст
- R2 Проектні обмеження
- DP1 План домену
- DP2 Модель домену
- DP3 Бази активів інженерії

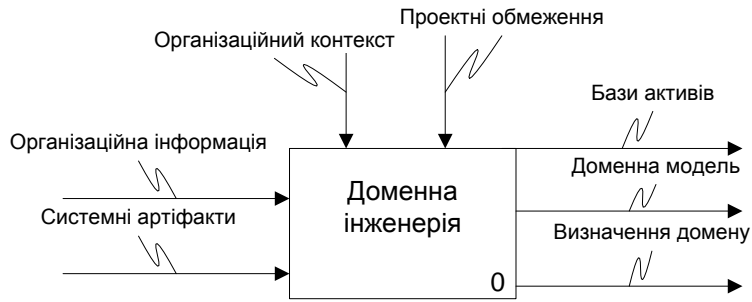


Рис. 2.1. Доменна інженерія

Static Behaviors

(Process Schema DOMAIN ENGINEERING = PN 0 || { ProcessID: DOMAIN ENGEERING ({I:I1,I2};{O:O1,O2,O3};{R:R1,R2}) } || { DetailedProcesses: DP1, DP2, DP3 })

Dynamic Behaviors

Process Dispatch \triangleq § \rightarrow (@Event1 DOMAIN ENGINEERING \rightarrow {PN0:DP1,DP2,DP3} \rightarrow §

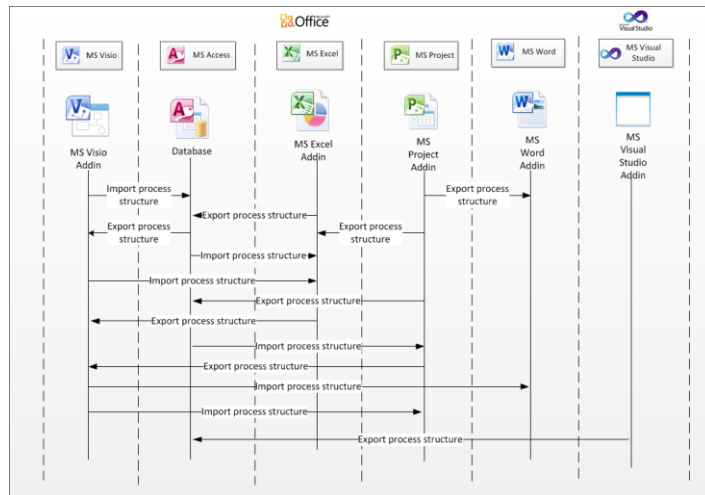


Рис.3.1. Залежність компонентів офісу

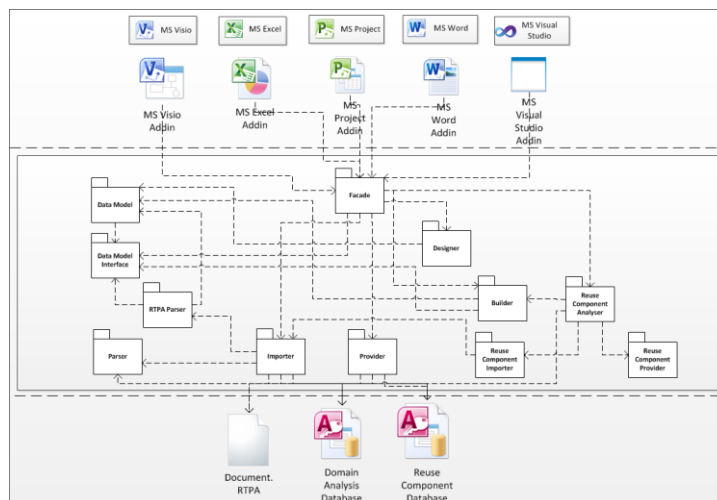


Рис.3.2. Багаторівнева архітектура

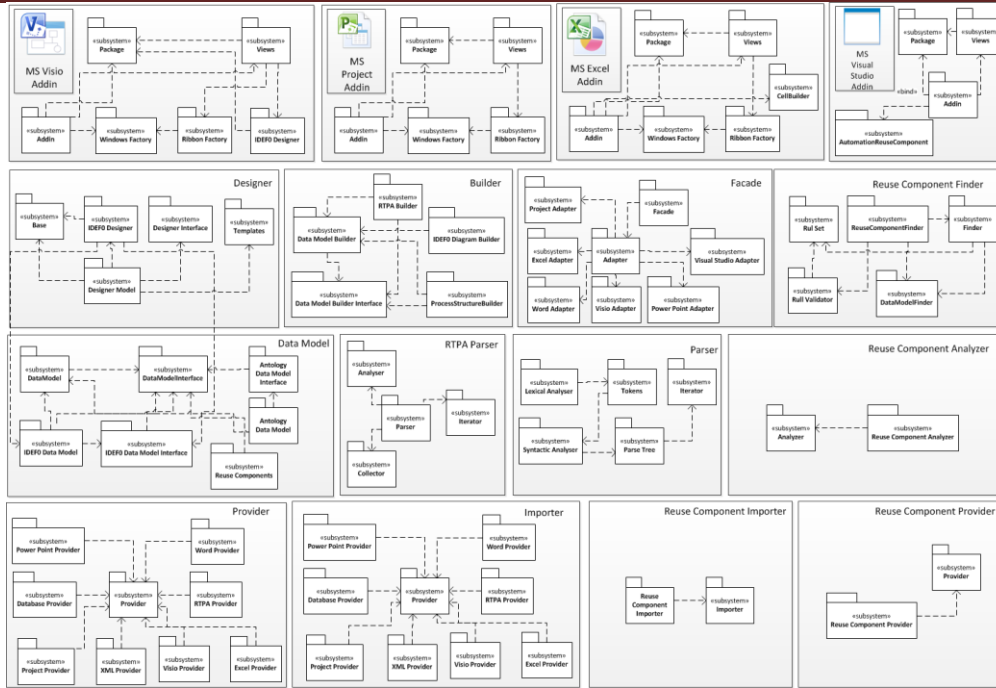


Рис.3.3. Діаграма компонентів

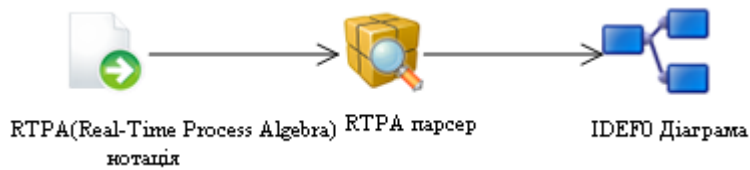


Рис.3.4. Імпорт нотації в діаграму

3.2 Опис функціональних можливостей програмного комплексу

Для запуску програми необхідно вибрати і запустити файл `distribution.exe`, який знаходиться у теці в яку було розпаковано архів `install.exe`.

Після запуску основного файлу програмного забезпечення відкривається головне вікно програми рис.3.5

рис. 3. Інтерфейс програмного продукту MS Office 2010 містить дві вкладки “Domain Analysis”, “Reuse Component Automation Finder”.

Вкладка “Domain Analysis” містить в собі меню: “Notation”, “Access Workflow, Views, Export”.

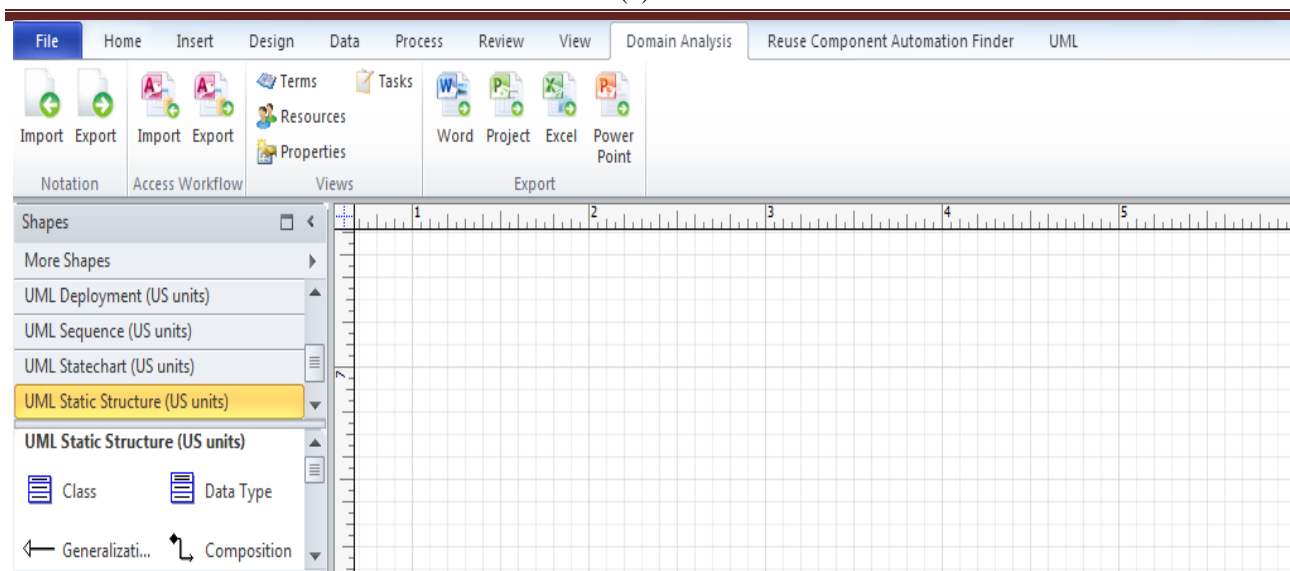


Рис. 3.5 Вікно програми

Підменю «Notation» (рис. 3.5) містить в собі дві функції зчитування даних та запису даних в текстовий файл вигляду. Натисканням на кнопку «Import» відкривається діалогове вікно пошуку файлу на диску. Після вибору відповідного файлу і натискання на кнопку «Open» знизу виводиться основна інформація, що характеризує вибраний файл. Натисканням на «Open» запускається процес зчитування вхідної інформації з вибраного текстового файлу до внутрішньої пам'яті програми. Натисканням на кнопку «Export» відкривається діалогове вікно збереження файлу на диску. Після вибору місця на диску для збереження файлу і натискання на кнопку «Save» файл зберігається на диску. (рис. 3.6)

Підменю «Access Workflow» (рис. 3.6) містить в собі дві функції зчитування даних та запису даних в базу даних на MS Access 2010 вигляду. Натисканням на кнопку «Import» відкривається діалогове вікно пошуку файлу на диску. Після вибору відповідного файлу і натискання на кнопку «Open» знизу виводиться основна інформація, що характеризує вибраний файл. Натисканням на «Open» запускається процес зчитування вхідної інформації з вибраного файлу бази даних на MS Access 2010 до внутрішньої пам'яті програми. Натисканням на кнопку «Export» відкривається діалогове вікно збереження файлу на диску. Після вибору місця на диску для збереження файлу і натискання на кнопку «Save» файл нової бази зберігається на диску. (рис. 3.6).

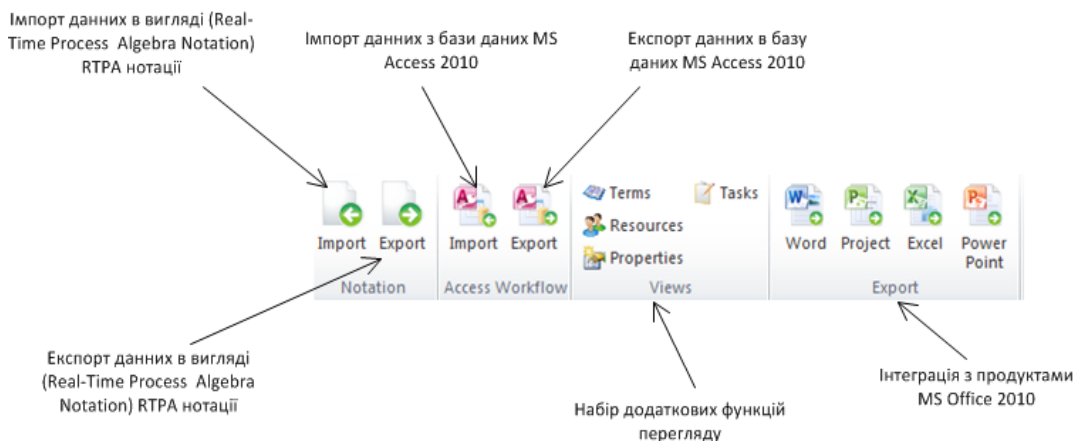


Рис. 3.6 Опис можливостей

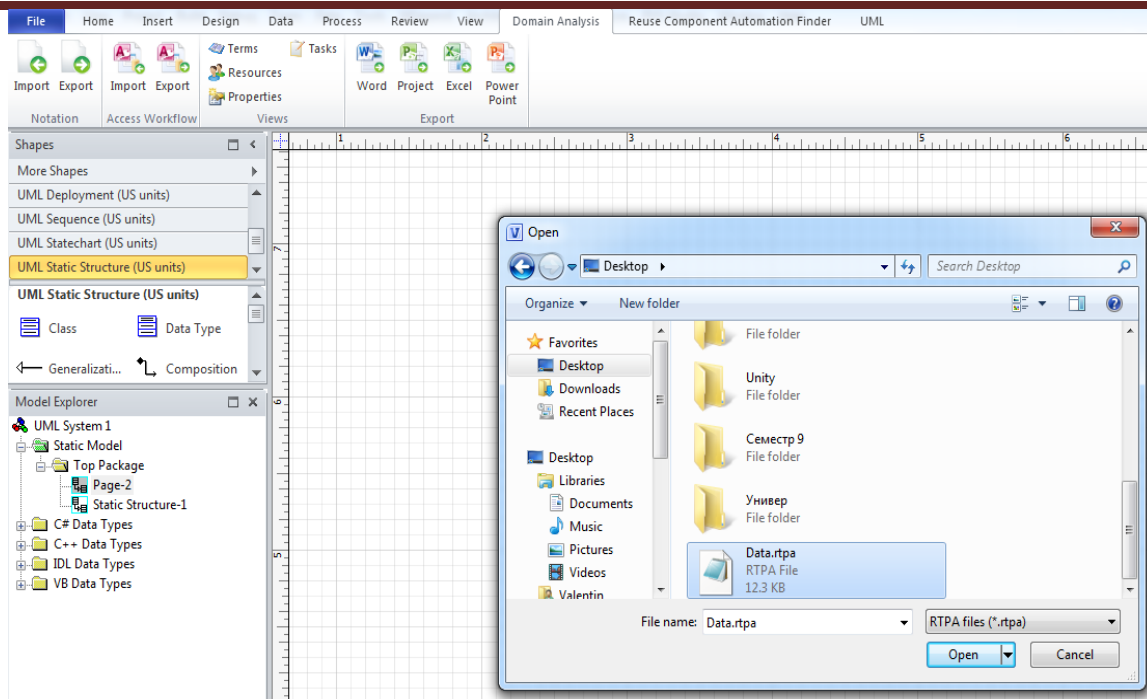


Рис. 3.7 Імпорт даних в текстову представленні

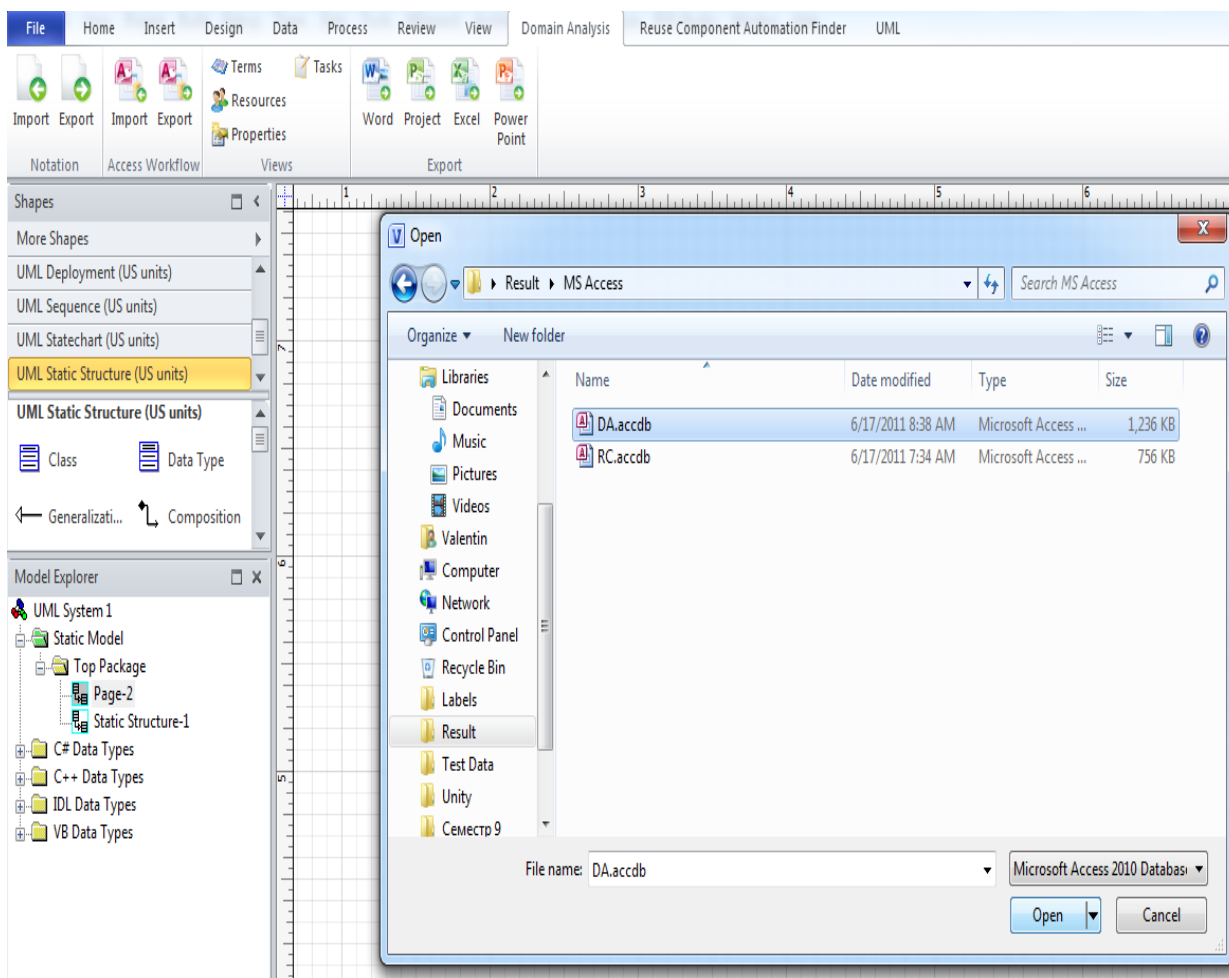


Рис. 3.8 Імпорт даних з бази даних MS Access 2010

Підменю «Views» (рис. 3.6) дає можливість працювати зі словником термінів “Terms” (рис.3.9), які використовуються в роботі з програмою, дає можливість переглянути задачі та ресурси, які необхідні для виконання процесів доменного аналізу “Resources”, ”Tasks”. Підменю «Export» (рис.3.5) дає можливість експортувати данні, які використовуються в MS Visio

2010 до MS Project 2010 (рис. 3.7) для управління часом виконання та контролю за використанням ресурсів, також є можливість представити цю інформацію у вигляді текстових даних (MS Word) або статистичних (MS Excel) використовуючи при цьому продукти MS Office 2010.

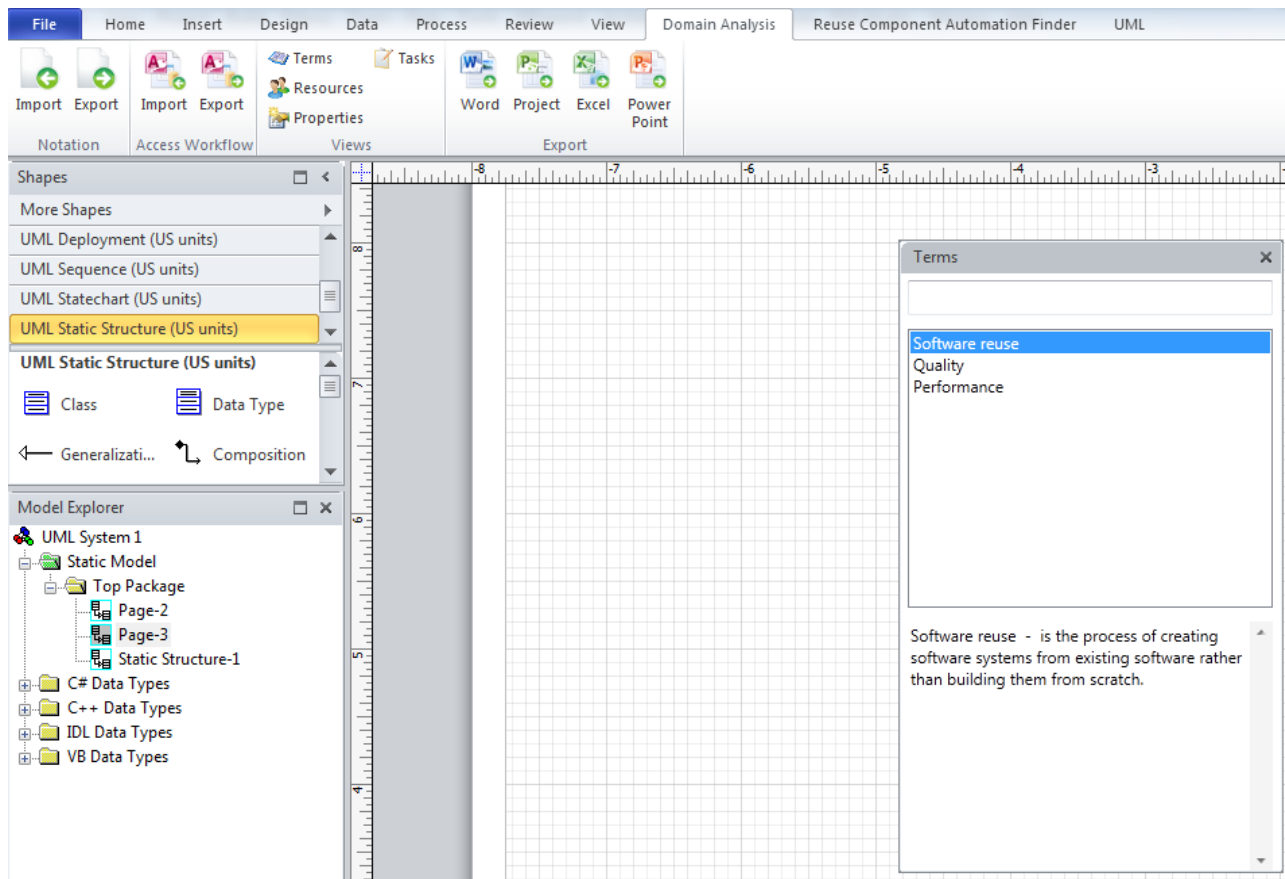


Рис. 3.9 Словник термінів

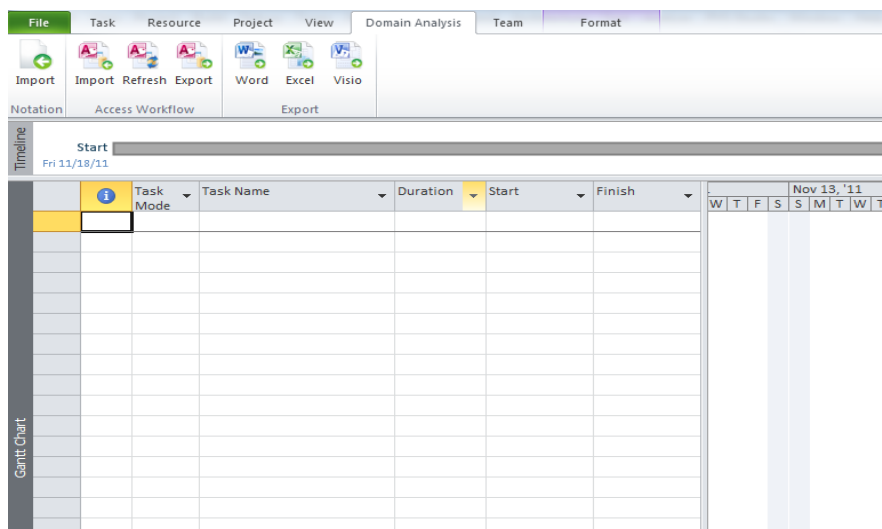


Рис. 3.10 Інтеграція з продуктом MS Project 2010

Підменю «Notation» (рис. 3.10) містить в собі дві функції зчитування даних з текстового файлу вигляду. Натисканням на кнопку «Import» відкривається діалогове вікно пошуку файлу на диску. Після вибору відповідного файлу і натискання на кнопку «Open» знизу виводиться основна інформація, що характеризує вибраний файл. Натисканням на «Open» запускається процес зчитування вхідної інформації з вибраного текстового файлу до внутрішньої пам'яті програми. (рис. 3.8)

Підменю «Access Workflow» (рис. 3.7) містить в собі дві функції зчитування даних та запису даних в базу даних на MS Access 2010

вигляду рис. 3.6. Натисканням на кнопку «Import» відкривається діалогове вікно пошуку файлу на диску. Після вибору відповідного файлу і натискання на кнопку «Open» знизу виводиться основна інформація, що характеризує вибраний файл. Натисканням на «Open» запускається процес зчитування вхідної інформації з вибраного файлу бази даних на MS Access 2010 до внутрішньої пам'яті програми. Натисканням на кнопку «Export» відкривається діалогове вікно збереження файлу на диску. Після вибору місця на диску для збереження файлу і натискання на кнопку «Save» файл нової бази зберігається на диску. (рис. 3.7).



Рис.3.11 Опис можливостей

Підменю «Export» (рис.3.11) дає можливість експортувати данні, які використовуються в MS Visio 2010 до MS Visio 2010 (рис. 3.9) для представлення інформації в графічному вигляді,

також є можливість представити цю інформацію у вигляді текстових даних (MS Word) або статистичних (MS Excel) використовуючи при цьому продукти MS Office 2010.

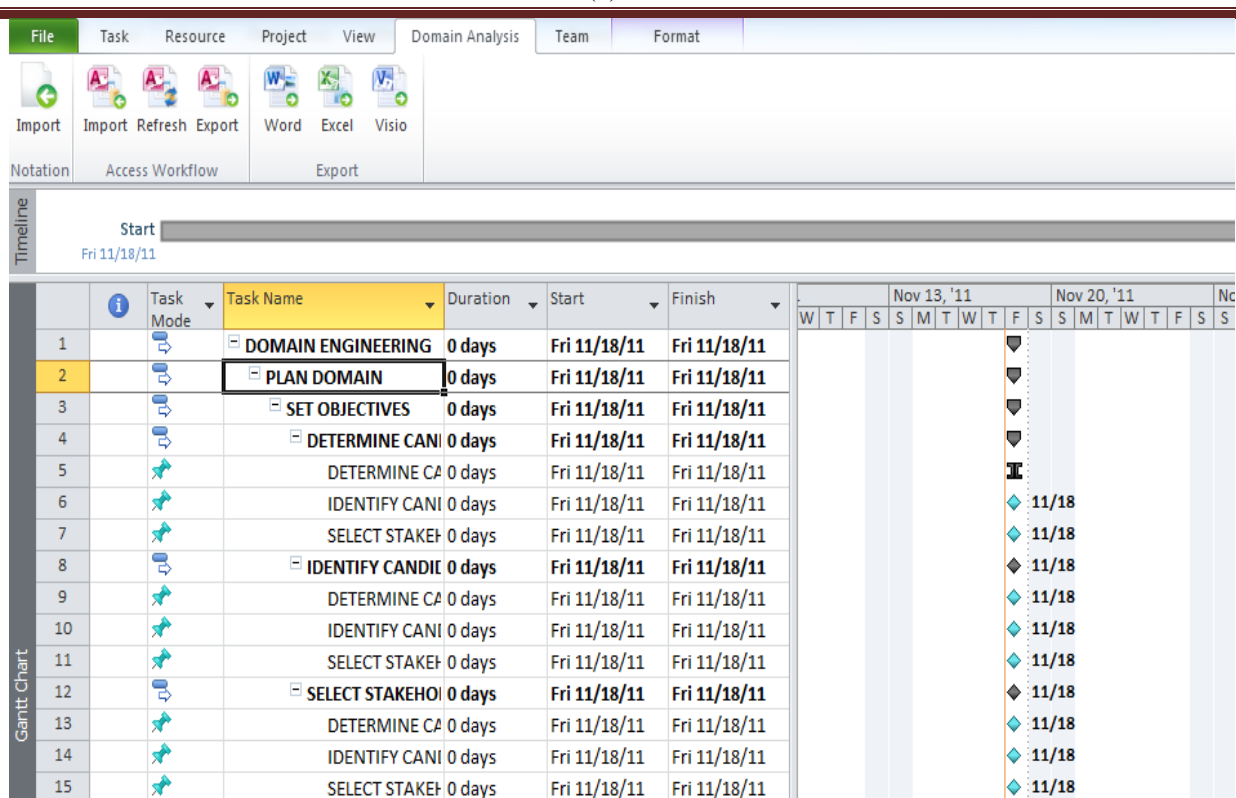


Рис.3.12 Імпорт даних (інтеграція з MS Visio 2010)

Література:

1. Wartik, S., R. Prieto-Diaz. "Criteria for Comparing Domain Analysis Approaches." ss. 24-48 International Journal of Software Engineering and Knowledge Engineering, September 1992. pp 24-57
2. Simos, M. "The Growing of an Organon: A Hybrid Knowledge-Based Technology and Methodology for Software Reuse." In Domain Analysis and Software Systems Modeling, R. Prieto-Diaz and G. Arango, ed., IEEE Computer Society Press, 1991. pp. 43-72
3. Simos, M., "Navigating Through Soundspace: Modeling the Sound Domain At Real World." In

Proceedings of the 4th Annual Workshop on Software Reuse, Herndon VA, November 1991. pp. 15-38

4. Simos, M., "Juggling in Free Fall: Uncertainty Management Aspects of Domain Analysis Methods." In Proceedings of the 5th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer-Verlag, July 1994. pp. 20-63

Відомості про автора



Дуднік Валентин – студент 4 курсу факультету комп'ютерних наук Національного авіаційного університету, кафедра «Інженерія програмного забезпечення». Наукові інтереси – доменний аналіз, технології програмного забезпечення

E-mail: Valentin.Dudnik@hotmail.com



Сидоров Николай Александрович, д.т.н., проф., декан факультета комп'ютерних наук, завідувач кафедри інженерії програмного забезпечення Національного авіаційного університету, наукові інтереси – інженерія програмного забезпечення.

E-mail: nikolay.sidorov@livenau.net