

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістра

(освітній ступінь)

на тему: **Методи і засоби автоматизованої генерації документації у процесі вдосконалення комп'ютерних систем**

Виконав: студент (ка) 2 курсу, групи СІд-2
спеціальності 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

| | | |
|-------------------|-------|--|
| | <hr/> | Тимощук М.В. (прізвище та ініціали) |
| Керівник | <hr/> | Луцків А.М. (прізвище та ініціали) |
| Нормоконтроль | <hr/> | Луцик Н.С. (прізвище та ініціали) |
| Завідувач кафедри | <hr/> | Осухівська Г.М. (прізвище та ініціали) |
| Рецензент | <hr/> | Гащин Н.Б. (прізвище та ініціали) |

Тернопіль
2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

ЗАТВЕРДЖУЮ

Завідувач кафедри Осухівська Г.М.

«_____» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Тимошук Марині Валеріївні
(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) Методи і засоби автоматизованої генерації документації у процесі вдосконалення комп'ютерних систем

Керівник проекту (роботи) Луцків Андрій Мирославович, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «14» грудня 2022 року №4/7-1011

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Методи формування документів, технології вдосконалення комп'ютерних систем, компоненти комп'ютерних систем, типи програмного забезпечення

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз підходів до формування документації при побудові комп'ютерних систем

2. Метод генерації документації при вдосконаленні комп'ютерних систем

3. Проектування архітектури інструменту автоматизованої генерації документації при вдосконаленні комп'ютерних систем 4. Охорона праці та безпека в надзвичайних ситуаціях.

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Актуальність і мета дослідження. 2. Задачі дослідження, об'єкт і предмет, наукова новизна і практична цінність дослідження. 3. Роль і місце документування в процесі вдосконалення комп'ютерних систем 4. Класифікація документів. 5. Логічна структура документів

6. Модельно-керована розробка документації. 7. Метод автоматизованої генерації документів

8. Висновки

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| <i>Охорона праці та безпека в надзвичайних ситуаціях</i> | <i>Осухівська Г.М.</i> | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|-------|--|--------------------------------|-----------------|
| 1. | <i>Аналіз підходів до формування документації при побудові комп'ютерних систем</i> | | <i>виконано</i> |
| 2. | <i>Метод генерації документації при вдосконаленні комп'ютерних систем</i> | | <i>виконано</i> |
| 3. | <i>Проектування архітектури інструменту автоматизованої генерації документації при вдосконаленні комп'ютерних систем</i> | | <i>виконано</i> |
| 4. | <i>Охорона праці та безпека в надзвичайних ситуаціях</i> | | <i>виконано</i> |
| 5. | <i>Оформлення пояснювальної записки</i> | | <i>виконано</i> |
| 6. | <i>Оформлення графічного матеріалу</i> | | <i>виконано</i> |
| 7. | <i>Попередній захист кваліфікаційної роботи магістра</i> | | <i>виконано</i> |
| 8. | <i>Захист кваліфікаційної роботи магістра</i> | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент _____

(підпис)

Тимощук М.В.

(прізвище та ініціали)

Керівник проекту (роботи) _____

(підпис)

Луцків А.М.

(прізвище та ініціали)

АНОТАЦІЯ

Методи і засоби автоматизованої генерації документації у процесі вдосконалення комп'ютерних систем // Кваліфікаційна робота магістра // Тимошук Марина Валеріївна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, група СІд-2 // Тернопіль, 2022 // с. – 87, рис. – 34 , табл. – 9, аркушів А1 – 8 , додат. – 1, бібліогр. – 21.

Ключові слова: документація, генерація, вдосконалення, автоматизація, комп'ютерна система.

Метою кваліфікаційної роботи є дослідження методів і засобів автоматизованої генерації документації в процесі вдосконалення комп'ютерних систем.

У роботі проаналізовано наукові публікації і практичні рекомендації щодо автоматизації процесу генерування документації при вдосконаленні комп'ютерних систем з використанням методів прямої і зворотної інженерії. Окрім цього досліджено існуючі технології автоматизованого формування структури документів та залежностей між ними, що дало змогу класифікувати їх за ознаками атрибутів, які є спільними для усіх технологій, спільними для конкретного типу документації і відповідно конкретного документу конкретної технології при вдосконаленні комп'ютерних систем

Формалізовано та обгрунтовано моделі для представлення документації програмного забезпечення у процесі вдосконалення комп'ютерних систем, що дало змогу побудувати ланцюжок перетворення однієї моделі документу в іншу. Розроблено метод автоматизованої генерації документів, який враховує вимоги технологій вдосконалення комп'ютерних систем і дає змогу формувати набір документації для майбутнього оновленого програмного забезпечення.

ABSTRACT

Methods and means of documentation automated generation in the process of improving computer systems // Master thesis// Tymoshchuk Maryna Valeriivna // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, group CId -2 // Ternopil, 2022 // p. - 87, fig. - 34 Table. -9, Sheets A1 - 8, Add. - 5, Ref. - 21.

Keywords: documentation, generation, improving, automation, computer system.

The purpose of the qualification work is to research the methods and means of automated generation of documentation in the process of improving computer systems.

The paper analyzes scientific publications and practical recommendations regarding the automation of the documentation generation process when improving computer systems using forward and reverse engineering methods. In addition, the existing technologies for the automated formation of the structure of documents and dependencies between them were investigated, which made it possible to classify them according to attributes that are common to all technologies, common to a specific type of documentation and, accordingly, to a specific document of a specific technology when improving computer systems

Models for presenting software documentation in the process of improving computer systems were formalized and substantiated, which made it possible to build a chain of transformation of one document model into another.

A method of automated document generation has been developed, which takes into account the requirements of technologies for improving computer systems and makes it possible to form a set of documentation for future updated software.

ЗМІСТ

| | |
|--|----|
| ВСТУП | 8 |
| РОЗДІЛ 1 АНАЛІЗ ПІДХОДІВ ДО ФОРМУВАННЯ ДОКУМЕНТАЦІЇ ПРИ ПОБУДОВІ КОМП'ЮТЕРНИХ СИСТЕМ | 12 |
| 1.1. Процес документування як допоміжний процес життєвого циклу | 12 |
| 1.2. Аналіз специфіки понять документ, документування та документація у процесі вдосконалення комп'ютерних систем..... | 14 |
| 1.3. Методи генерації документації при вдосконаленні комп'ютерних систем.. | 22 |
| 1.3.1. Аналіз існуючих методів генерування документації | 23 |
| 1.3.2. Аналіз генерації документації в процесі реінженерії ПЗ комп'ютерних систем | 27 |
| 1.4. Висновки до розділу | 28 |
| РОЗДІЛ 2 МЕТОД ГЕНЕРАЦІЇ ДОКУМЕНТАЦІЇ ПРИ ВДОСКОНАЛЕННІ КОМП'ЮТЕРНИХ СИСТЕМ..... | 29 |
| 2.1. Розробка на основі моделей..... | 29 |
| 2.2. Моделі документів ПЗ в процесі вдосконалення комп'ютерних систем | 30 |
| 2.2.1. Формалізація моделі документів при вдосконаленні комп'ютерних систем | 31 |
| 2.2.2. Метамоделі документації опису технологій вдосконалення комп'ютерних систем | 35 |
| 2.2.3. Модель представлення документації використовуваних технологій вдосконалення комп'ютерних систем..... | 36 |
| 2.3. Метод генерації документації при вдосконаленні і модернізації комп'ютерних систем | 37 |
| 2.3.1. Обґрунтування та опис методу..... | 38 |
| 2.3.2. Реалізація методу базованого на моделях | 41 |
| 2.4. Висновки до розділу | 50 |

| | | | |
|---|------------------------------------|--------------------------|----|
| РОЗДІЛ 3 ПРОЕКТУВАННЯ | АРХІТЕКТУРИ | ІНСТРУМЕНТУ | |
| АВТОМАТИЗОВАНОЇ ГЕНЕРАЦІЇ ДОКУМЕНТАЦІЇ ПРИ | ВДОСКОНАЛЕННІ | КОМП'ЮТЕРНИХ СИСТЕМ..... | 51 |
| 3.1. Концептуальна архітектура інструменту | автоматизованої генерації | документів..... | 51 |
| 3.2. Характеристика компонентів архітектури | засобу автоматизованої генерації | документів..... | 54 |
| 3.2.1. Підсистема управління генерацією | документів..... | | 55 |
| 3.2.2. Редактор метаописів при автоматичній | генерації документації..... | | 56 |
| 3.2.3. Менеджер представлень..... | | | 60 |
| 3.2.4. Генератор документів..... | | | 64 |
| 3.2.5. Довідники у системі автоматичної | генерації документації..... | | 67 |
| 3.3. Висновки до розділу..... | | | 68 |
| РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В | НАДЗВИЧАЙНИХ СИТУАЦІЯХ | | 69 |
| 4.1. Охорона праці..... | | | 69 |
| 4.2. Здоровий спосіб життя людини та його | вплив на професійну діяльність.... | | 72 |
| ВИСНОВКИ..... | | | 77 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | | | 79 |
| Додаток А Тези конференції..... | | | 81 |

ВСТУП

Актуальність теми магістерської роботи. Технологічність сфери ІТ забезпечує впровадження нових підходів для вирішення та підвищення ефективності процесів автоматизації як для окремих компаній, так і для звичайних користувачів. Спостерігаючи стрімкий ріст та використання хмарних сховищ і сервісів, зниження порогу входження розробників як програмного, так і апаратного забезпечення, розвиток методів і засобів штучного інтелекту, інтернету речей і DevOps технологій можна зробити висновок, що технології виробництва комп'ютерних та інформаційних систем стали досконалыми. Проте, незважаючи на такий стан речей, відкритими залишаються багато питань, які пов'язані з переходом із застарілих технологій на більш сучасні. При цьому актуальними задачами є забезпечення документованості програмного забезпечення у процесі його міграції з однієї технології на іншу.

Створювати документацію з нуля за наявності існуючої у процесі вдосконалення чи оновлення комп'ютерних систем є хибним шляхом, оскільки вимагає значних трудозатрат та фінансових витрат. Тому доцільною та актуальною задачею у сфері інформаційних технологій є розробка методів та інструментів автоматичної генерації документації хоча б окремих компонентів комп'ютерних систем при їх модернізації.

При цьому необхідно вирішувати ряд підзадач, як наприклад, аналіз вимог щодо вдосконалення системи, їх порівняння з властивостями існуючого програмного забезпечення, використовуваними технологіями реалізації та ін.

Важливим аспектом при міграції чи вдосконаленні комп'ютерних систем є створення документів на різних етапах виконання процесів і формування представлень (подання) документів. В якості інструменту для документованого представлення результатів виконання етапів вдосконалення комп'ютерної системи можна розглядати нотації мови UML.

Зафіксовані таким чином документи потрібно брати до уваги при реалізації майбутнього програмного забезпечення.

Такий підхід до вдосконалення комп'ютерних систем, особливо до програмної складової, передбачає розвиток методів і засобів зворотної інженерії і побудову зв'язків з прямими технологіями реалізації при документуванні результатів. Автоматизація процесу формування документації ПЗ досліджується багатьма вченими, як закордонними (М. Freeman, Т. Biggerstaff, V. Rajlich), так і вітчизняними (О. Авраменко, М. Сидоров, В. Хоменко).

Результати їхніх досліджень є доволі вагомим внесок у розв'язання проблеми автоматизації процесу створення документації для наслідуваного програмного забезпечення, однак все ж залишається не до кінця дослідженими методи і засоби автоматичної генерації документації при вдосконаленні комп'ютерних систем.

Метою роботи є дослідження методів і засобів автоматизованої генерації документації програмного забезпечення у процесі вдосконалення комп'ютерних систем.

Для досягнення зазначеної мети у роботі сформульовано наступні **задачі**:

- аналіз наукових публікацій і практик автоматизованої генерації документації при вдосконаленні або модернізації комп'ютерних систем;
- аналіз існуючих методів і технологій створення документації програмного забезпечення;
- побудова моделей опису документації з врахуванням особливостей технологій як прямої, так і зворотної інженерії;
- розробка методу автоматизованої генерації документації з врахуванням аспектів конкретних технологій реалізації і етапу вдосконалення ПЗ;
- побудова архітектури інструментального засобу автоматизованої генерації документації ПЗ.

Об'єкт дослідження – процеси автоматизованої генерації документації комп'ютерних систем.

Предмет дослідження – моделі, методи і засоби генерації документів програмного забезпечення.

Наукова новизна одержаних результатів:

– уперше формалізовано та обґрунтовано моделі для представлення документації програмного забезпечення у процесі вдосконалення комп’ютерних систем, що дало змогу побудувати ланцюжок перетворення однієї моделі документу в іншу на трьох рівнях абстракції і врахувати властивості використовуваних технологій реалізації компонентів комп’ютерних систем.

– набув подальшого розвитку метод автоматизованої генерації документів, який враховує вимоги технологій вдосконалення комп’ютерних систем і дає змогу формувати набір документації для майбутнього оновленого програмного забезпечення та набори його представлення, що керуються параметрами конкретної технології й етапом початку проведення відновлення документації.

Методи дослідження. Для виконання задач дипломної роботи магістра використано наступні методи:

- абстракції та узагальнення – при аналізі документації технологій розробки програмного забезпечення;
- формалізація і моделювання – при розробці моделей документів ПЗ;
- аналіз і проектування – при розробці засобів, які реалізують автоматизовану генерацію документації ПЗ КС при їх модернізації;
- експеримент – при апробації запропонованих методів і засобів;
- вимірювання – при аналізі результатів експериментального і практичного застосування методу та інструментальних засобів.

Практичне значення одержаних результатів. Практичне значення одержаних результатів полягає в імплементації архітектури інструментального засобу автоматизованої генерації документації у процесі вдосконалення комп’ютерних систем.

Публікації. Публікації. Результати кваліфікаційної роботи апробовані на X науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року) як тези конференцій.

1. Луцків А.М., Тимошук М.В. Важливість документування в процесі вдосконалення комп'ютерних систем. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 71.

2. Луцків А.М., Тимошук М.В. Моделі представлення документації програмного забезпечення при вдосконаленні комп'ютерних систем. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 72.

Структура роботи. Кваліфікаційна робота містить розрахунково-пояснювальну записку та графічний матеріал. До складу записки входить вступу, 4 розділи, загальні висновки, список використаних джерел і додатки. Обсяг роботи: розрахунково-пояснювальна записка – 92 арк. формату А4, графічна частина – 8 аркушів формату А1.

РОЗДІЛ 1

АНАЛІЗ ПІДХОДІВ ДО ФОРМУВАННЯ ДОКУМЕНТАЦІЇ ПРИ ПОБУДОВІ КОМП'ЮТЕРНИХ СИСТЕМ

1.1. Процес документування як допоміжний процес життєвого циклу

Процес формування документації на етапах життєвого циклу комп'ютерних систем передбачає фіксацію важливої інформації та результатів виконання кожного з основних процесів. Процес генерування документації, зазвичай, відбувається разом з іншими процесами реалізації КС (рис. 1.1).

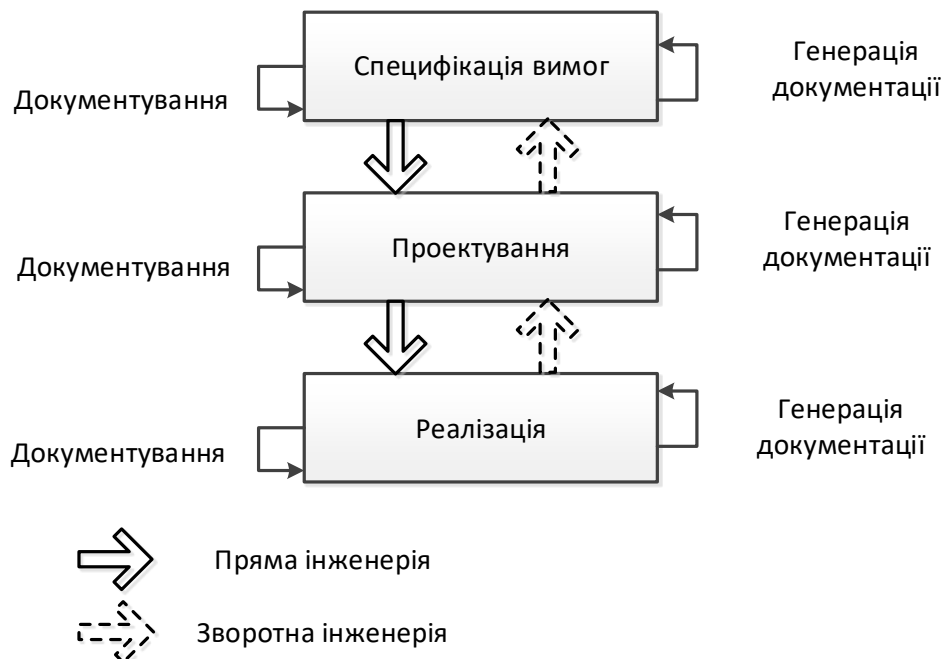


Рис. 1.1. Документування в процесі прямої і зворотної інженерії

В контексті зворотної інженерії застосовується синтез документації КС (рис. 1.1). Синтез документації в процесі зворотної інженерії практикується при вдосконаленні КС або при реінженерії наслідуваного ПЗ КС, в тому випадку коли документація програмного продукту загублена, неповна, неточна або погано структурована.

В якості результату синтезу документації у процесі вдосконалення комп'ютерних систем виступають альтернативні форми подання ПЗ. Оскільки, при цьому, формування документації застосовується тільки до фази реалізації,

то розглядаються різні форми предсавлення вихідного коду: лістинги, списки перехресних посилань, графічні представлення у вигляді діаграми потоків даних, структур даних і потоків управління. Генерування документації розглядається в контексті реінженерії ПЗ і визначається як процес, який забезпечує створення документації для існуючого програмного забезпечення. Окрім того, формування документації програмного забезпечення комп'ютерних систем визначається як процес аналізу ПЗ, результатом якого є документація, представлена в різних формах, включаючи документацію для користувачів і реформатування лістингів вихідних кодів.

При генеруванні документації фіксується інформація, добута з існуючого вихідного коду за допомогою зворотної інженерії, яка включає діаграми потоків даних, схеми потоків управління. У ряді робіт розглядається інкрементний підхід до створення документації на основі веб-технологій, яке визначається як процес генерації гіпертекстових нотацій до вихідного коду ПЗ. Практичне застосування генерування документації за допомогою засобу ARIS Redocumentation Scout (RDS) SAP призначене для внесення зміни у документацію бізнес-процесів, підтримуваних ПЗ компанії SAP, і забезпечує пошук, збір і обробку інформації по недокументованих бізнес-процесах, що реалізуються в SAP-системі. Аналіз сучасного вживання поняття генерації документації розробниками показує, що він розглядається як документування в процесі удосконалення або супроводу комп'ютерної системи.

Порівнюючи процеси документування та генерації документації в процесі вдосконалення комп'ютерних систем, можна сказати, що загальним для цих процесів є їх результат – документація і, відповідно, роботи, пов'язані з плануванням, проектуванням і випуском документації.

Основна відмінність процесів полягає в тому, що документування реалізується в прямій інженерії, а генерація документації – у зворотній (рис. 1.1). Об'єктом документування є розроблювальне або супроводжуване ПЗ КС. Об'єктом синтезу супровідної документації – удосконалене ПЗ, документація якого повністю або частково втрачена або містить багато неточностей, застаріла

або погано структурована. Для отримання інформації про об'єкт генерування документації використовуються засоби зворотної інженерії.

Таким чином, генерацію документації у процесі вдосконалення комп'ютерних систем можна визначити як процес створення документації, що відповідає встановленим вимогам або стандартам, з використанням результатів зворотної інженерії.

1.2. Аналіз специфіки понять документ, документування та документація у процесі вдосконалення комп'ютерних систем

Документ (від лат. Documentum зразок, свідоцтво, доказ) – це матеріальний об'єкт, що містить інформацію у зафіксованому вигляді, спеціально призначений для її передачі в часі і просторі.

Документи класифікують за формою подання, змістом і типом матеріального носія (рис. 1.2). За формою подання документи поділяються на текстові (книги, журнали, рукописи), графічні або образотворчі (креслення, схеми, графіки, плани, карти, діаграми) і аудіовізуальні (звукозаписи, кінофільми, діапозитиви). За типом матеріального носія розрізняють документи на паперовому, магнітному носії, фото-і відеоплівці.



Рис. 1.2. Класифікація документів

Узагальнена схема створення, обробки та використання документу представлена на рис. 1.3. Автор документу – це особа, яка проектує та розробляє документ. Автор проводить вивчення шляхом безпосереднього сприйняття об'єкта документування, збір інформації про об'єкт з допомогою технічних засобів, аналітичне дослідження, в результаті чого формує представлення, яке складає інформаційну чи змістову основу документа.

Створена інформація редагується, коригується і форматується для отримання необхідних властивостей документа, потім переноситься на матеріальний носій. У ролі автора зазвичай виступає людина. Однак створення документу може здійснюватися і без участі людини. Наприклад, створення сейсмограм відбувається автоматично; процес обробки штрих-кодів здійснюється без людської участі; більшість інформації в контурах автоматичного управління створюється і використовується без участі людини; звіти про події, які відбулися за певний період, формуються автоматично системою управління технологічним процесом.

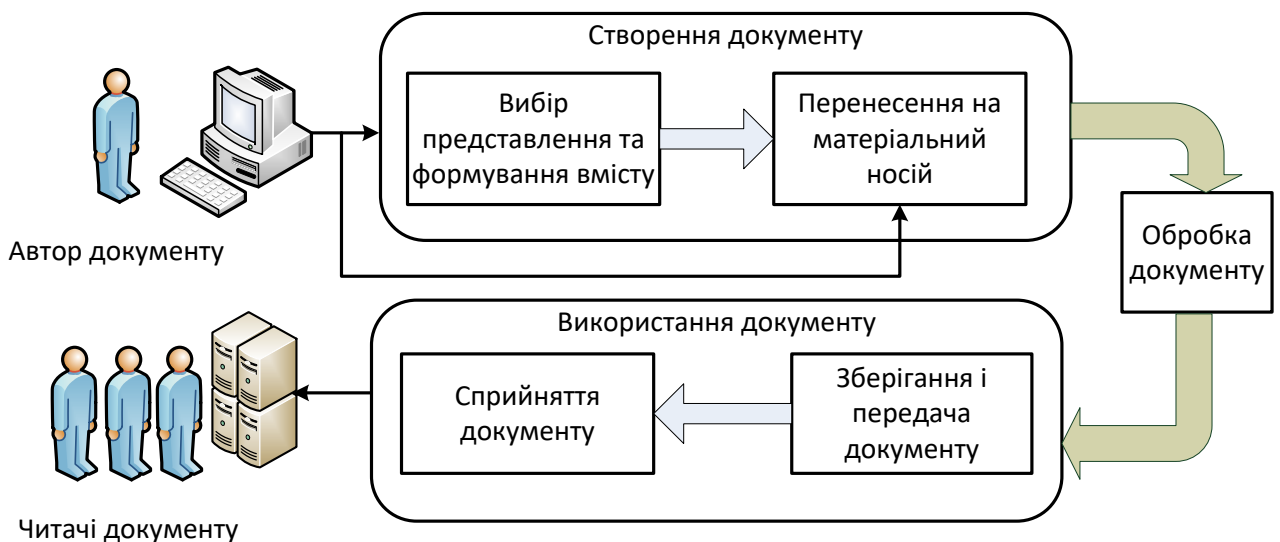


Рис. 1.3. Схема створення, опрацювання та використання документів

Створений документ піддається опрацюванню, яке може бути двох видів [5]:

- семантичне, пов'язане з вмістом документу (переклад, реферування, анотування);
- несемантичне, не пов'язану з вмістом документа (копіювання, тиражування, перетворення в іншу форму подання).

Документ може зберігатися або передаватися шляхом тиражування копій (друкованих або електронних) або поширенням через комп'ютерну мережу. Наприклад, документ ПЗ КС може зберігатися автономно – окремо від ПЗ (наприклад, видрукувані рекомендації) або бути вбудованим до складу ПЗ – технічно виконаним, як його складова частина (наприклад, довідкова система). Читачами документу є люди або технічні пристрої, для яких адресовано документ. При використанні документу його читачі сприймають інформацію з матеріального носія і формують представлення про об'єкт документування, необхідне їм виконання задач, пов'язаних з ним.

У галузевих стандартах приводиться кілька визначень документу ПЗ. Відмінності у визначенні документу ПЗ обумовлюються областю застосування стандарту, з точки зору якої розглядається визначення. У стандарті, який регламентує процеси управління програмними проектами, документ розглядається як сукупність даних, записаних на будь-якому носії інформації, які, як правило, володіють постійністю і можуть бути прочитані людиною чи машиною.

Стандарт специфікує процес розробки користувацької документації, визначає документ як цільову інформацію, розроблену для певного кола користувачів, представлену на носії інформації (наприклад, книга, диск, відео) в установленому форматі. Стандарт, який застосовується при специфікації змісту документації автоматизованих систем (АС), визначає документ як унікально ідентифіковану одиницю інформації, призначену для використання людиною, наприклад, звіт, специфікація, інструкція або книга. Документ у процесі вдосконалення комп'ютерних систем можна представити у вигляді трьох компонентів $d=(C,S,P)$, де C – контент, S – структура, P – вид представлення

документу. Ці параметри по різному пов'язані з характеристиками документу – інформацією і фізичним носієм, що проілюстровано на рис. 1.4.

Зміст (контент) документу відображає яким чином розуміти об'єкт документування, його атрибути, застосування та ін. Також наповнення документу можна розглядати у вигляді сукупності подання, що відображає дані про його структурні елементи та документ загалом.

Прикладами подання програмного забезпечення комп'ютерних систем може бути множина вимог, use case діаграми, діаграми послідовностей та об'єктів, ER-діаграма бази даних та ін.

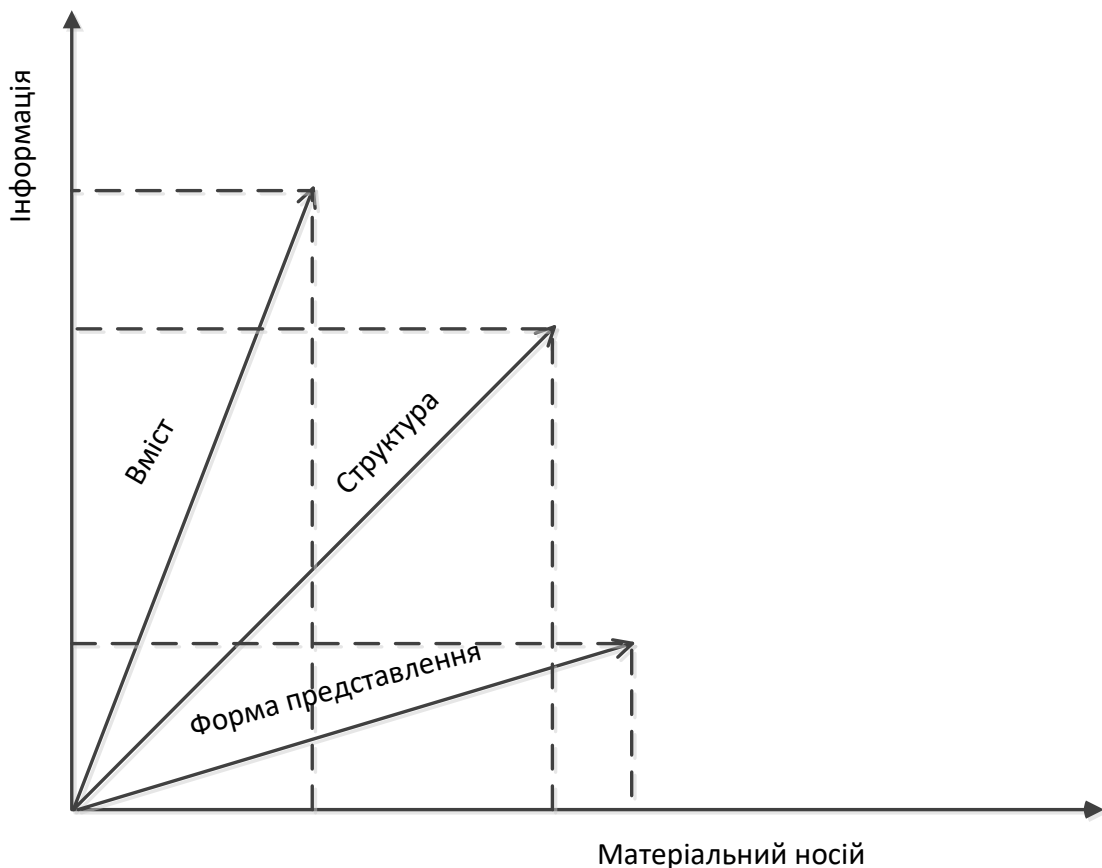


Рис. 1.4. Коомпненти документу при вдосконаленні комп'ютерних систем

Структура документа має відношення до інформації, яка в ньому знаходиться та до носія, на якому він зберігається. Інформація впливає на структурованість вмісту документу, а носій – на службову частину документу, наприклад, пов'язану з пошуковими і навігаційними можливостями у документі.

Зазвичай, будову документа представляють як сукупність елементарних компонентів і зв'язків між ними.

Мета процесу структуризації документа полягає у відображенні логіки організації інформації, а також спрощення знаходження потрібних даних та переходу по частинах документа. Елементарні компоненти документа, які відображають його структуру, пропонується групувати за змістовими ознаками, що визначаються наповненістю, ідентифікаційними ознаками та службовими для забезпечення зручного переходу між розділами документа, як показано на рис. 1.5.

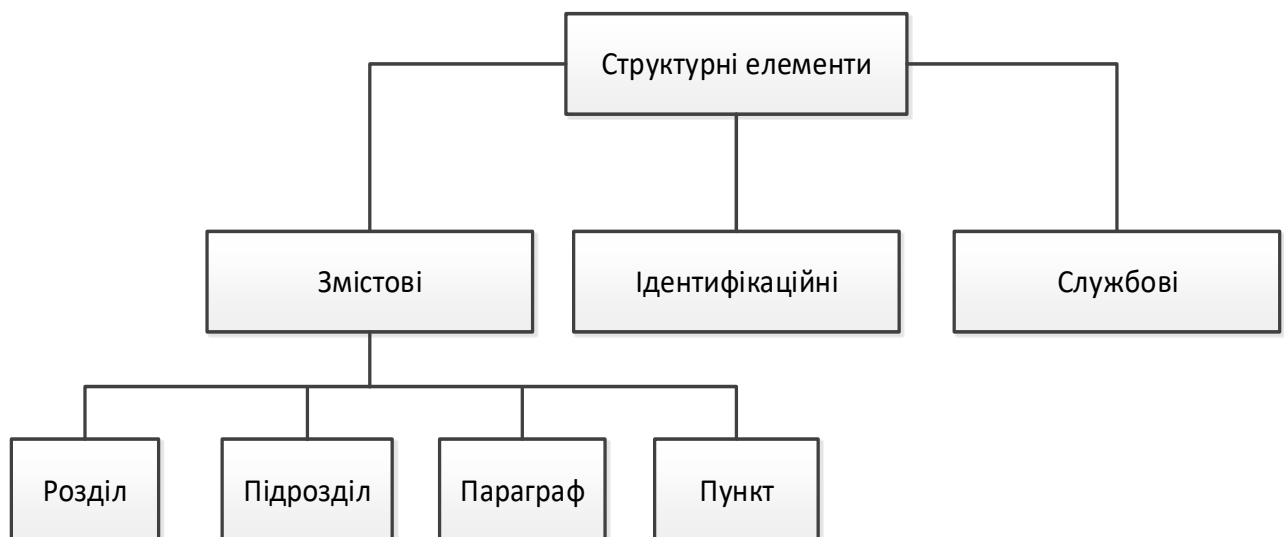


Рис. 1.5. Класифікація структурних елементів документа

Змістовні структурні елементи відображають організацію інформації, що міститься в документі, у вигляді логічно закінчених частин, що утворюють ієрархічну структуру (рис.1.6).

Склад змістовних структурних елементів і структура документа встановлюються видами документів і профілями. Види документів визначено у стандартах. Наприклад, в комплексі стандартів 9-й серії ГОСТ Єдиної системи програмної документації (ЕСПД) структура кожного виду документа описується як перерахування назв необхідних розділів і описів, що вказують, які відомості

потрібно включити в розділ. Профіль представляє собою шаблон, який налаштовується і визначає склад документації і структуру документів (рис. 1.7).

Склад – це перелік документів, а структура – сукупність інформаційних пунктів. Інформаційний пункт – це мінімальна структурна одиниця документу, що містить відомості про ПЗ. Один інформаційний пункт може входити в більш ніж один документ або бути взагалі відсутнім. Стандарт наводить перелік всіх інформаційних пунктів, які можуть входити в той чи інший документ.

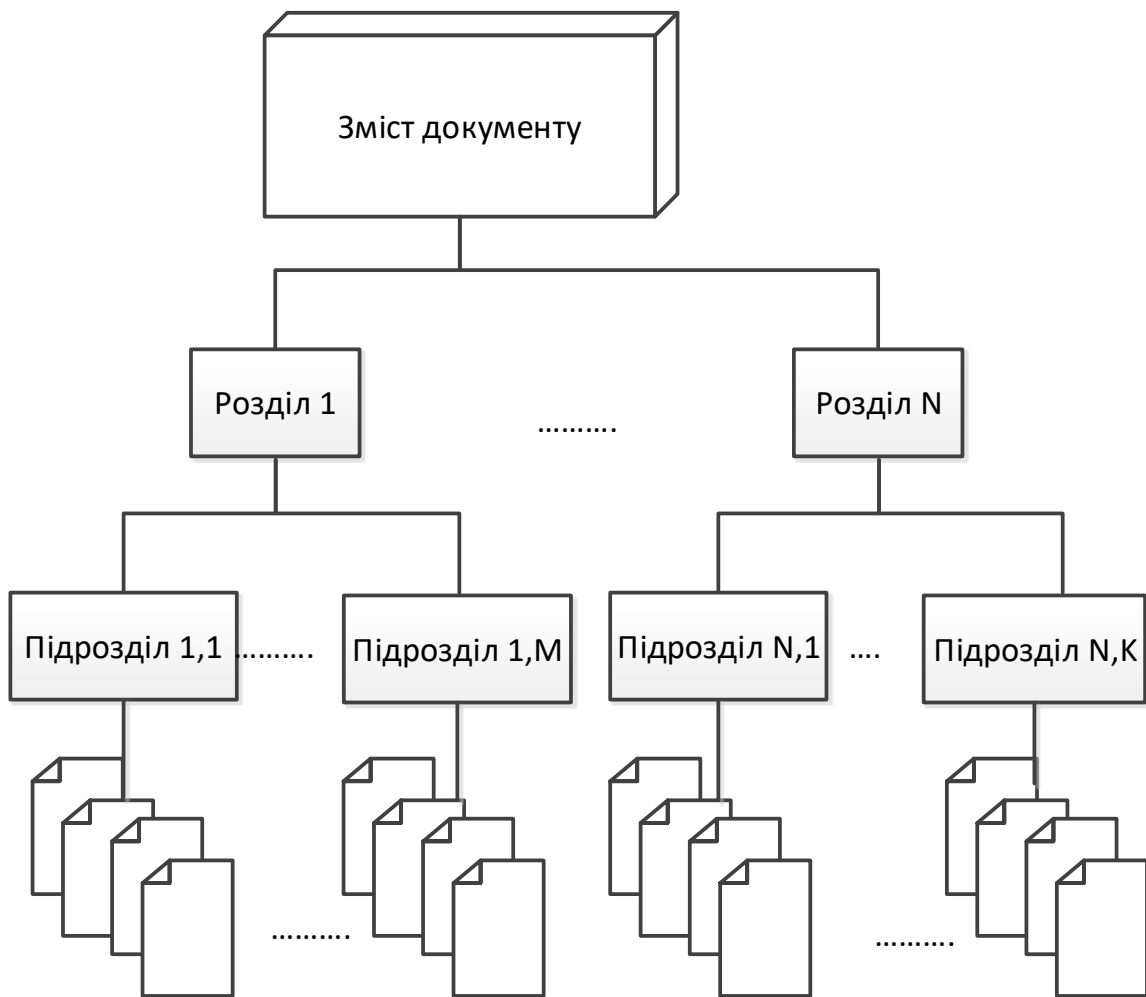


Рис.1.6. Логічна структура документу

Підгрупа
інформаційного
пункту

| Інформаційний пункт | | | Документ-1 | Документ-2 |
|---------------------|-------------------|--------|------------|------------|
| Номер | Назва | Рівень | | |
| α |α | α | α | α |
| α | 234 Компоненти ПЗ | | | |
| 2341 | Організація ПЗ | 1 | α | С |
| | | 2 | α | α |
| | | 3 | Е | α |
| 2342 | Організація даних | 1 | α | α |
| | | 2 | Е | α |
| | | 3 | С | α |
| α | α | α | α | α |

Складова інформаційного пункту

Рівень інформаційного пункту

Рівень обов'язковості входження

Рис. 1.7. Приклад профілю документації стандарту

Ідентифікаційні структурні елементи документа (рис. 1.5) використовуються для унікальної ідентифікації документа та управління нумерацією. Інформація, необхідна для ідентифікації документа – це заголовок документа, його реєстраційний номер, номер версії документа, дата випуску виправлення, автор, вказівка повноважень (або клас конфіденційності), захисний і авторський знаки, назва випускаючої організації. Ідентифікаційні елементи виносяться на титульний аркуш друкованого документа або на початкову сторінку on-line документа.

Службові структурні елементи призначені для забезпечення читабельності і зручності використання документа. До таких елементів належать такі:

- зміст з назвами розділів і номерами сторінок;
- список ілюстрацій з номерами малюнків і їх назвами;
- вступ, який описує мету документа і коротко його зміст;
- інформація по використанню документа;
- глосарій, в якому наведено визначення спеціальних термінів, що використовуються в документі;

- список джерел інформації у вигляді посилань або документів;
- покажчик у вигляді списку ключових термінів та сторінок, що містять посилання на ці терміни.

В екранних документах використовуються додаткові службові структурні елементи, що забезпечують пошук спеціальних термінів і можливість навігації, яка дозволяє читачеві знайти своє поточне положення і переміщатися по документу. Форма подання документа визначається правилами, сформованими шляхом встановлення значень для стандартизованих або корпоративних елементів оформлення документа. До них відносять правила форматування тексту, заголовків, таблиць, використання колонтитулів, розмір відступів, оформлення списків, графічного матеріалу. Крім того, певні елементи оформлення відносяться тільки до документів на паперовому носії, наприклад, розмір і якість паперу, правила зшивання, а інші висуваються тільки до електронних довідок та on-line документів, наприклад, підсвічування при використанні кольору.

Документація – це сукупність документів, присвячених деякому питанню, явищу, процесу, особі, установі. Документація ПЗ – це один і більше пов'язаних документів ПЗ. Таким чином, документацію ПЗ D можна представити як непорожню множину, елементами якої є документи: $D = \{ D_1, D_2 \dots D_n \}$.

Документація ПЗ забезпечує наступне:

- відстеження термінів та стану проекту при управлінні розробкою ПЗ;
- узгоджує взаємодія учасників проекту;
- забезпечує якість ПЗ;
- розуміння і правильне використання ПЗ.

Структуру документа ПЗ, відповідно до стандарту, складають титульна, інформаційна та основна частини. Зовнішнє представлення документів встановлюється у вигляді загальних вимог до оформленню документів ПЗ для будь-якого способу виконання документів на різних матеріальних носіях. Стандарт встановлює види програмних документів, для кожного з яких описується їх зміст.

Стандарти ГОСТ 34 мають статус міждержавних для країн СНД. Об'єктом стандартизації цього комплексу стандартів є автоматизовані системи (АС) різного призначення і всі види компонентів, включаючи ПЗ і БД. Основна увага комплексу приділяється змісту документів кожного етапу ЖЦ.

Деякі вітчизняні підприємства керуються стандартом MIL-STD-49. Це стандарт Міністерства оборони США, регламентує розробку ПЗ, надає список документів, які супроводжують розробку ПЗ і детально описує їх зміст. Стандарт не зареєстрований в Україні.

Існують міжнародні стандарти ISO/IEC, регламентують склад документації ПЗ і керування процесом проектування (рис. 1.8).

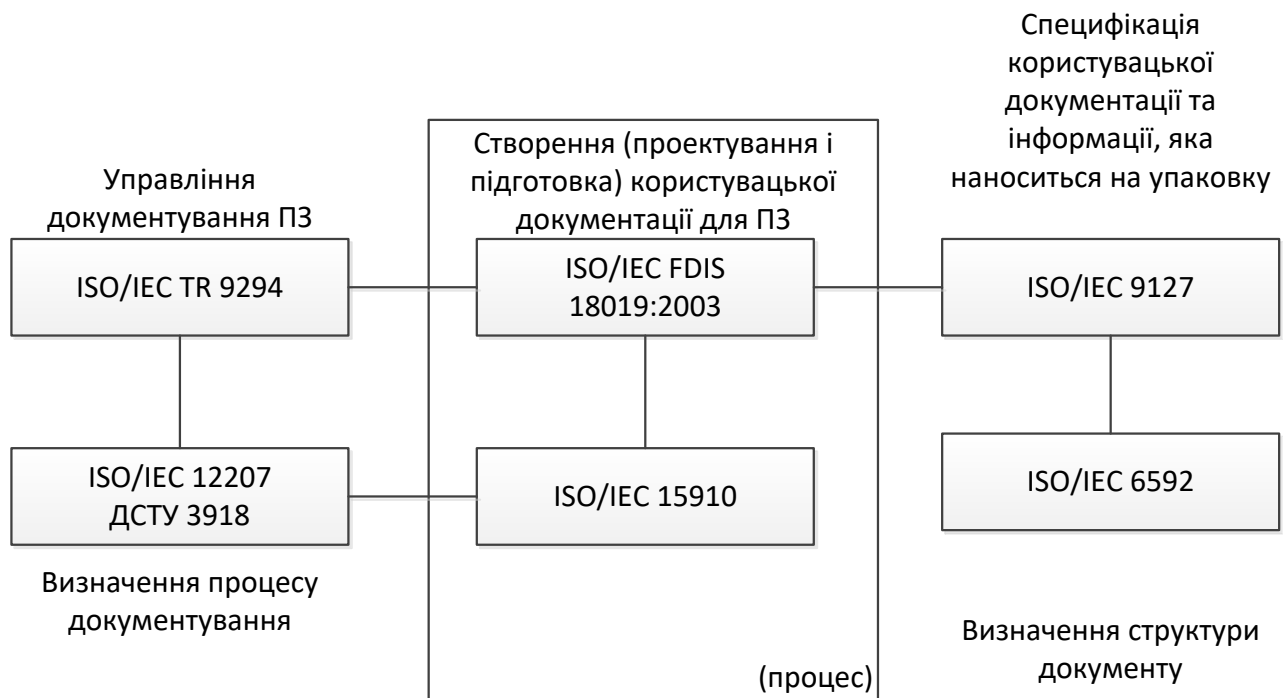


Рис. 1.8. Стандарти ISO/IEC, які регламентують документацію ПЗ

1.3. Методи генерації документації при вдосконаленні комп'ютерних систем

Для постановки задачі розробки методу генерування документації при вдосконаленні комп'ютерних систем варто розглянути існуючі методи та оцінити їх з точки зору застосування в процесі реінженерії.

1.3.1. Аналіз існуючих методів генерування документації

Аналіз робіт в області зворотної інженерії та реінженерії виявив два існуючих на сьогоднішній день методи генерування документації: структурна та інкрементна генерація на основі веб-технологій. Структурна генерація ПЗ при вдосконаленні комп'ютерних систем вирішує задачу відновлення архітектурних аспектів програмного забезпечення комп'ютерних систем з використанням зворотної інженерії. В основу методу покладено принципи віртуальних розшарувань підсистем (virtual subsystem stratifications - VSS) і моделей програмних взаємозв'язків (software interconnection models - SIM). Принцип VSS означає поділ на модулі логічних структур визначених користувачем артефактів (підсистем), потім декомпозиція їх на групи та / або формування цих структур у шари. Розділення на модулі і шари породжує двовимірне представлення структури ПЗ із залежностями всередині та між шарами. Третій вимір необхідний для відображення різних представлень. Представлення ПЗ у вигляді куба формує повний набір документації ПЗ. Для підтримки декількох релізів вводиться четвертий вимір, що показує рух «куба» у часі (рис. 1.9).

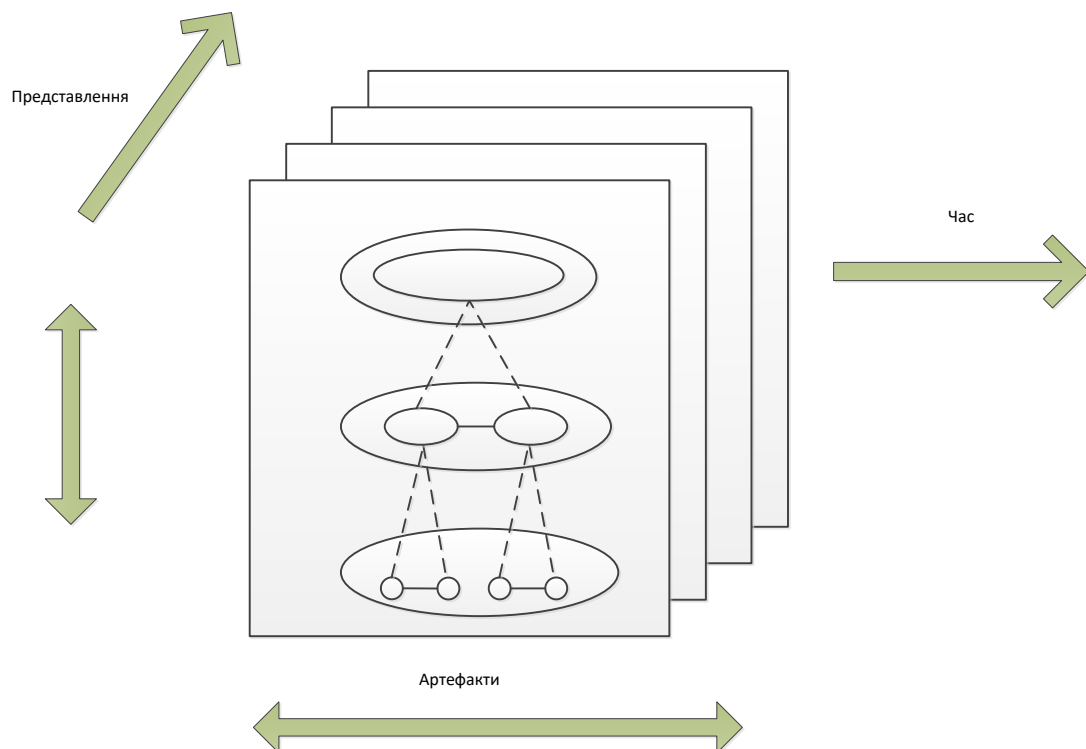


Рис. 1.9. Принцип VSS

SIM – це формальні моделі, що використовуються для опису відношень між об'єктами в програмній системі у вигляді набору кортежів: $SIM = \{ \{ \text{об'єкти} \}, \{ \text{відношення} \} \}$. Набір кортежів зображається графом з об'єктами в його вузлах і відношенням у вигляді дуг між вузлами. Залежності всередині шару моделюються відношеннями у синтаксичній моделі взаємозв'язків.

Наприклад, SIM всередині шару = ($\{ \text{артефакт} \}, \{ \text{залежить-від} \}$). Залежності між шарами визначають те, як один шар пов'язаний з іншим. Відношення, які зв'язують шари – це композиція, наслідування, вкладеність. Наприклад, SIM між шарами = ($\{ \text{підсистема} \}, \{ \text{є-частиною} \}$).

Структурний синтез реалізується двома фазами. У першій фазі, яка виконується автоматично, проводиться синтаксичний аналіз вихідного коду, визначення артефактів і збереження їх в репозиторії. У результаті породжується неструктурований граф потоку ресурсів, який потім використовується для представлення структурних залежностей, таких як виклики функцій і доступ до даних.

Друга фаза виконується експертами по розпізнаванню шаблонів і технології композиції підсистем. Внаслідок цього, генеруються багаторівневі ієрархії для високорівневих абстракцій. Наприклад, аналітик може об'єднати функції в підсистему відповідно до бізнес-правил або загальноприйнятих принципів модульності ПЗ, забезпечуючи різні альтернативні ракурси, необхідні для обслуговування ПЗ. При цьому композиція підсистем є рекурсивним процесом групування блоків (наприклад, типів даних, процедур) в складні підсистеми. Після дослідження структури підсистем, аналізу побудованих ієрархій підсистем, ідентифікації інтерфейсів, виділення центральних і периферійних компонентів експерт приступає до синтезу представлень. Ці представлення становлять основу документації ПЗ (рис. 1.10).

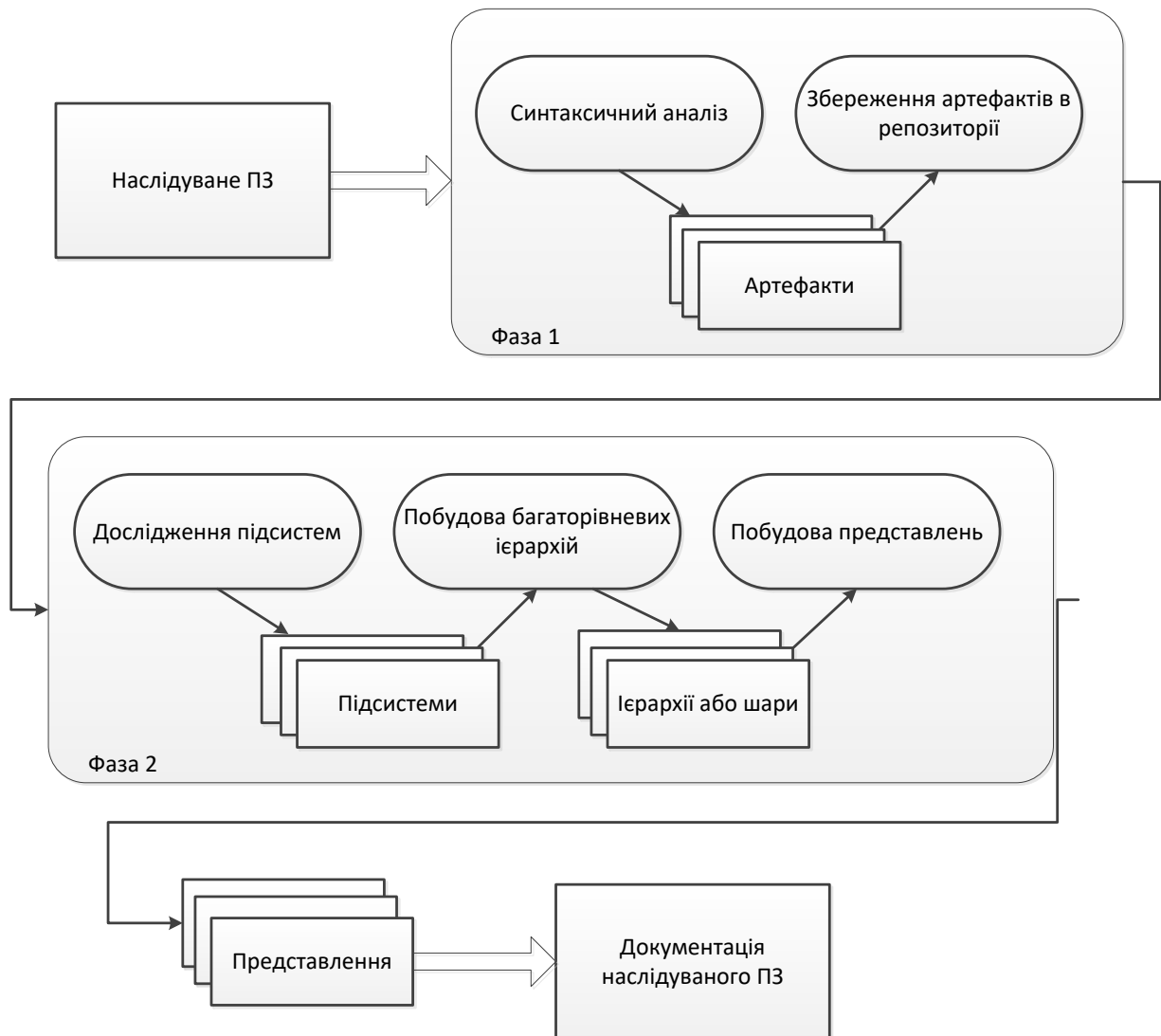


Рис. 1.10. Фази структурного синтезу супровідної документації

Для реалізації структурного синтезу супровідної документації розроблений спеціальний інструмент Rigi [17], що складається з таких компонентів:

- система синтаксичного аналізу, яка підтримує мови C і COBOL;
- репозиторій для збереження інформації, добутої з вихідного коду;
- віконний графічний редактор для управління представленнями програми.

Інкрементний синтез супровідної документації з використанням веб-технологій полягає у використанні гіпертекстової нотації PAS (Partitioned Annotations of Software) для опису вихідного коду. В гіпертексті вихідний код і документація представлені як перехресні посилання з коду в документацію і між

компонентами в кодї. Для перегляду використовуються стандартні web-браузери.

Засіб синтезу супровідної документації на основі PAS є гіпертекстовим блокнотом, в який розробник може записувати інформацію, що стосується ПЗ. Кожен компонент вихідного коду (клас, функція, залежність чи аргументи функції) супроводжується анотацією, яка його пояснює. Анотація поділена на розділи, що описують компоненти на різних рівнях абстракції або з різних точок зору.

Всі компоненти в рамках одного проекту спільно використовують ідентичні розділи. Вкладеність всередині компонентів коду відображаються і на вкладеності в їх анотаціях – анотації функцій-членів класів вбудовуються в анотації класів, а анотації аргументів функцій – в анотації функцій.

Анотації представлені в обов'язкових розділах, які завжди генеруються, і вибіркового розділах, які формуються на вимогу користувача. До обов'язкових належать розділи опису та залежностей для класів, описи і алгоритми для функцій-членів класів, описи області для аргументів функцій. Вибіркові розділи варіюються і засновані на потребах самого проекту. Архітектура PAS складається з кореневого файлу, списку всіх класів і гіпертекстових зв'язків з їх анотаціями. Анотації класів мають вказівники на код класу. При пошуку за допомогою PAS, користувач рухається між кодом і анотаціями, використовуючи «прямий» і «зворотній» напрям. Оскільки, код є «листочком» дерева анотацій, він не має вбудованих гіпертекстових посилань. Код та анотації оновлюються незалежно. У фазі планування учасники програмного проекту ідентифікують ті частини ПЗ, в які вноситимуться зміни. Виконуючи зміни і верифікуючи їх правильність у фазі синтезу супровідної документації, для розуміння програми зроблені зміни записуються у відповідні розділи PAS. Анотації витягуються з коментарів у кодї і створюються вручну. Для витягання використовується генератор анотацій – Hypertext Management System (HMS), робота якого заснована на програмному аналізі для виділення програмних компонентів, їх залежностей і збереження цієї інформації в БД.

1.3.2. Аналіз генерації документації в процесі реінженерії ПЗ комп'ютерних систем

Продовження використання наслідуваного ПЗ здійснюється шляхом застосування його в колишньому призначенні (після відновлення) або шляхом застосування його або назв компонентів за іншим призначенням (після переробки або повторним використанням). При вдосконаленні комп'ютерних систем реінженерія ПЗ є одним із шляхів трансформації ПЗ.

Аналіз існуючих методів показує, що вони дозволяють вирішити задачу синтезу супровідної документації, однак не задовольняють в достатній мірі вимоги до документації, необхідної при проведенні реінженерії. Метод структурного синтезу спрямований на документування тільки архітектурних аспектів наслідуваного ПЗ. Результат такого синтезу сформульований неоднозначно: з одного боку він є сукупністю представлень, а з іншого – документацією ПЗ. В методі не розглядаються склад документації та вимоги до документів. Автоматизація методу зачіпає тільки задачу добування інформації з наслідуваного ПЗ і роботу експертів з аналізу інформації.

Метод інкрементного синтезу супровідної документації дозволяє документувати об'єктно-орієнтований код, створюючи до нього електронні текстові анотації заданої структури. Структура документів збігається зі структурою анотацій, а їх вміст формується автоматично шляхом добування з коментарів до коду або заповнюється програмістом вручну.

Метод дозволяє отримувати електронні документи тільки у вигляді анотацій для фази реалізації і найбільше застосовується при супроводі ПЗ.

Розглянуті методи орієнтовані на вирішення задачі документування тільки деяких особливостей ПЗ при вдосконаленні комп'ютерних систем і не беруть до уваги аспектів застосування результатів генерації документів під час реінженерії, зокрема, вимог технологій реалізації прямої інженерії.

1.4. Висновки до розділу

1. Проаналізовано наукові публікації і практичні рекомендації щодо автоматизації процесу генерування документації при вдосконаленні комп'ютерних систем з використанням методів прямої і зворотної інженерії, що дало змогу підтвердити їхню важливість та актуальність імплементації на стадіях життєвого циклу, а також встановити структуру і зміст документів на кожному з них.

2. Досліджено існуючі технології автоматизованого формування структури документів та залежностей між ними, що дало змогу класифікувати їх за ознаками атрибутів, які є спільними для усіх технологій, спільними для конкретного типу документації і відповідно конкретного документу конкретної технології при вдосконаленні комп'ютерних систем.

3. Проведено аналіз існуючих методів генерації документів, що орієнтовані на використання структурного або інкрементного підходів та веб-технологій, у результаті якого підтверджено той факт, що їх застосування у процесі зворотної інженерії не забезпечує можливості одержання документації, яка б задовольняла встановленим вимогам.

РОЗДІЛ 2

МЕТОД ГЕНЕРАЦІЇ ДОКУМЕНТАЦІЇ ПРИ ВДОСКОНАЛЕННІ КОМП'ЮТЕРНИХ СИСТЕМ

2.1. Розробка на основі моделей

Модель в загальному випадку – це представлення деякого об'єкта або сукупності об'єктів (оригіналу) [13]. Наприклад, математичні рівняння є аналітичними моделями. Модель представляє форму оригіналу, але відрізняється від нього за масштабом, реалізацією або поведінкою. Моделі застосовуються для прогнозування поведінки або властивостей оригіналу і описуються за допомогою набору позначень. Під моделлю ПЗ розуміється специфікація системи та її характеристики, і виконується з певною метою [18]. Концепція створення моделей як основи для розуміння, проектування, конструювання, впровадження та експлуатації, обслуговування та модифікації систем реалізована у вигляді підходу Model Driven Development. Суть модельно-керованої розробки передбачає трансформацію, пов'язаних одна з одною моделей, які описують одну систему на різних рівнях абстракції, що в результаті приводить до написання коду [14, 19] (рис.2.1).

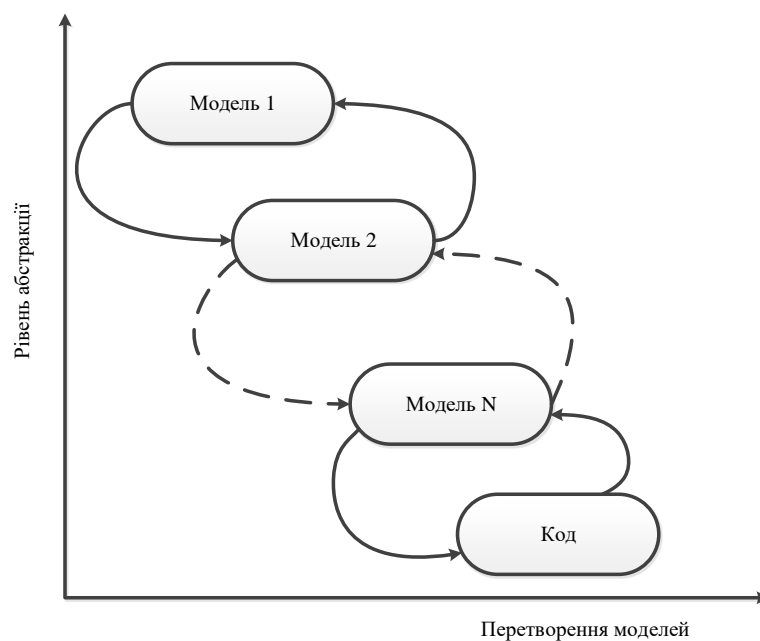


Рис. 2.1. Підхід MDD при розробці ПЗ

MDA базується на таких стандартах:

- уніфікована мова моделювання (UML) [16];
- стандарт для обміну даними між банками даних, системами підтримки прийняття рішень і технологіями порталів (Common Warehouse Metamodel CWM) [18];
- загальна абстрактна мова для опису метамodelей, основа для CWM і UML-метамodelей (Meta-Object Facility - MOF) [5];
- XML-формат для зберігання та обміну метаданими (XML Metadata Interchange - XMI) [6].

2.2. Моделі документів ПЗ в процесі вдосконалення комп'ютерних систем

Проводячи дослідження документів та їхніх частин з точки зору генерування документації з врахуванням особливостей загального підходу та вимог, які висуваються до технологічного стеку розробки та конкретних документів, визначають типи моделей, які показані на рис.2.2.

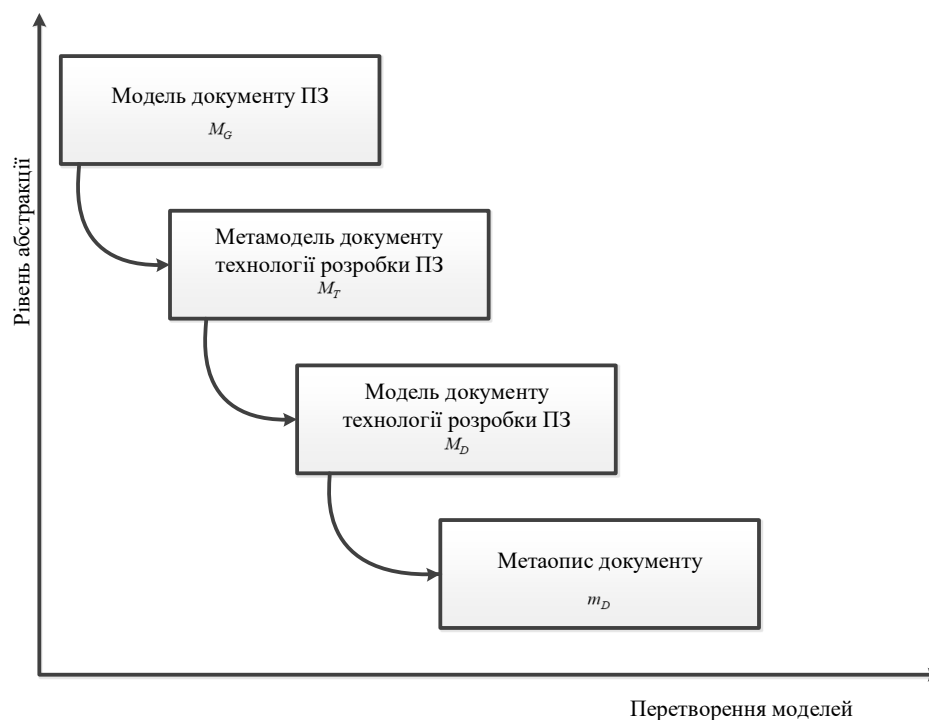


Рис. 2.2. Моделі документу у процесі вдосконалення комп'ютерних систем

Модель документу ПЗ M_G призначена для опису структурних елементів документу ПЗ та враховує особливості автоматизованого створення документації та не залежить від вимог до використовуваних технологій розробки. Метамоделю документації M_T щодо використовуваних технологій реалізації описує структурні компоненти документу ПЗ і враховує вимоги R_i , що володіють спільними рисами технологій розробки. Модель M_T є метамоделлю M_G . Модель M_D , інтерпретує документ технології розробки, що забезпечує опис структурних елементів документу з врахуванням вимог R_i для конкретної технології. Ця модель застосовується при створенні описів конкретних документів технології як метаданих, що можуть використовуватись при формуванні екземплярів документів при синтезі документації автоматизованих шляхом.

2.2.1. Формалізація моделі документів при вдосконаленні комп'ютерних систем

При формалізації документів при вдосконаленні комп'ютерних систем документ доцільно представити у вигляді моделі кортежа $D = \langle C, S, P \rangle$. Перший компонент кортежа C описує контекст (інформаційне наповнення), другий компонент моделі описує структурні елементи, а останній – тип подання для зовнішнього застосування.

Інформаційне наповнення документу представляється у вигляді множини $C = \{c_j / j = 1..L\}$, компонентами якої виступають елементарні інформаційні елементи, що відповідають властивостям логічної завершеності, а доступ до них напряму є обмеженим. При автоматичній генерації документації, інформаційні елементи формуються шляхом інтеграції представлень, одержаних за допомогою методів оберненої інженерії. До таких представлень належать текстове вираження сукупності вимог до комп'ютерних систем, use case діаграми, статичні UML діаграми, ER-діаграми, структура і вид інтерфейсів користувачів та ін.

Представлення V може приймати деяке значення z , що інтерпретує текстову інформацію, графічний вид, відео і т.п.. Окрім цього, V описується атрибутами p_1, \dots, p_n , які визначають, наприклад, носій, матеріальна розташування та ін.

$$v = \langle z, (p_1, \dots, p_n) \rangle \quad (2.1)$$

При автоматизованій генерації документів представлення v_i з множини V може інтерпретувати один або кілька елементів c_j множини C . Таку залежність можна представити у вигляді матриці, як показано у табл. 2.1. Формально взаємозв'язок між множинами представлень документу і його інформаційним наповненням відображається за допомогою матриці $A[K,L]$, K - потужність V ; L - потужність C ; $A = \{a_{ij} \mid a_{ij} = \{0,1\}\}$. Значення 1 у матриці $A[K,L]$ передбачає відповідність та взаємозв'язок v_i та c_j , нуль відображає відсутність взаємозв'язку [6].

Таблиця 2.1

Відповідність елементів множини представлень і вмісту документу

| Множина представлень (V) | Множина ІЕД (C) | | | | |
|-----------------------------|-----------------|-------|-------|-------|-------|
| | c_1 | c_2 | c_3 | | c_L |
| v_1 | 1 | 0 | 0 | ... | 0 |
| v_2 | 0 | 1 | 0 | ... | 0 |
| v_3 | 1 | 0 | 0 | | 1 |
| | ... | ... | ... | | |
| v_K | 0 | 0 | 1 | | 1 |

Інформаційному елементу документу, окрім представлень, може відповідати нотація-пояснення ex у вигляді тексту $z(ex)$. Такі нотатки формують сукупність $EX = \{ex_i / i = 1, \dots, M\}$.

Це означає, що для $\forall c_k \in C$ істинним є:

$$c_k \Leftrightarrow \bigcup_{i=1}^L v_i^k \vee \bigcup_{j=0}^M ex_j^k \quad (2.2)$$

де $c_k \in C$, $v_i^k \in V$, $ex_j^k \in EX$.

Структуру документу можна зобразити у вигляді дерева $S = \langle N, E \rangle$, де $N = \{n_i \mid i = 1, \dots, M\}$ – множина структурних елементів документу (СЕД), E – сукупність впорядкованих відношень між вершинами дерева (ребер), які візуально представлено на рис. 2.3 [5].

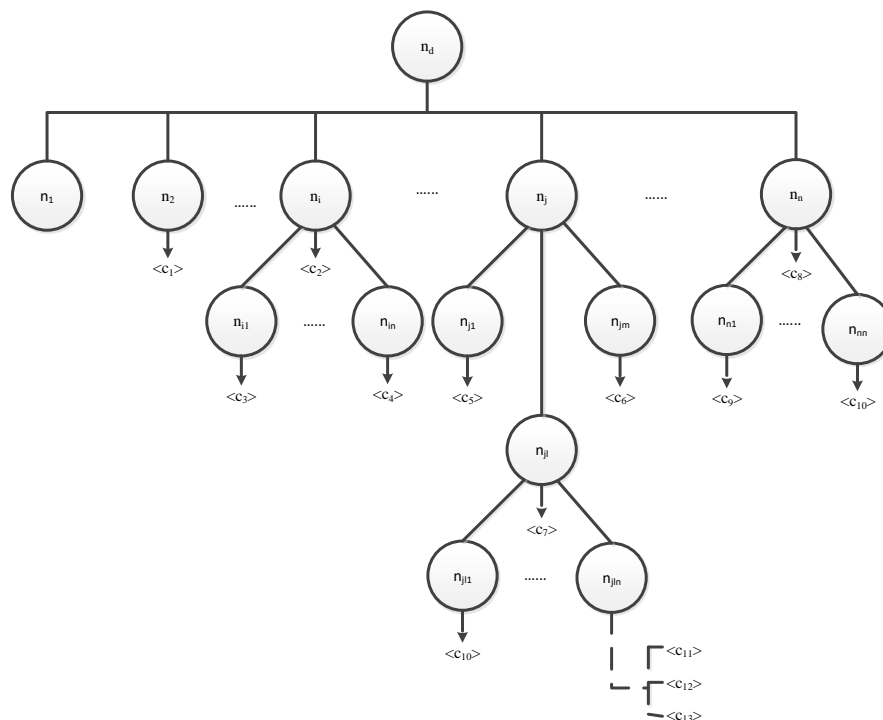


Рис. 2.3. Деревовидне зображення компонентів документу

Множина N утворена шляхом застосування операції об'єднання підмножин структурних елементів, які не перетинаються, тобто $N = N_c \cup N_{id} \cup N_s$. У даному випадку компоненти N включають:

- елементи підмножини Nc , які відображають інформаційний контент документу (розділи, пункти, підпункти);
- ідентифікатори Nid , призначені для унікального визначення документу, може включати його назву та версію;
- службові елементи Ns , які використовуються для організації пошуку та навігації по документу (глосарій).

Структурні елементи, які відносяться до групи змістових і службових, можуть володіти вкладеністю елементів, а їхній порядок визначається набором побудованих ребер. Для прикладу, корінь дерева n_d містить набір ребер, як проілюстровано на рис. 2.3, а формально це можна описати наступним виразом

$$((n_d, n_1), (n_d, n_2), \dots, (n_d, n_i), \dots, (n_d, n_j), (n_d, n_k)) \quad (2.3)$$

Для елементів множин C і N можна задати зв'язок O такий, що у випадку наявності деякого інформаційного елемента $c \in C$ завжди передбачає існування структурного елемента $n \in N$. Таким чином, $O: C \mapsto N$ представляє собою відображення інформаційних елементів у структурні і навпаки [6].

Зазвичай, структурний елемент описується значенням деякого тексту $z(nc)$, що інтерпретується як заголовок і йому у відповідність можна поставити інформаційний елемент документу. Формально, у випадку істинності умови $\forall nc \in Nc$, це можна описати наступним чином:

$$nc_k: \Leftrightarrow \bigcup_{i=0}^L nc_i^k \oplus \bigcup_{j=0}^M c_j^k, \quad (2.4)$$

де $nc_k \in Nc$, $nc_i^k \in Nc$, $c_j^k \in C$.

Прикладом може служити другий розділ документу «Додаткові специфікації» технології RUP [3, 10]. Цей розділ представляється як змістовний структурний елемент, який має значення «2. Функціональні вимоги». Цьому структурному елементу відповідатиме інформаційний елемент, що описує

функціональні вимоги до системи та інструментальні засоби розробки для роботи з вимогами, і змістовні СЕД для кожної з вимог. В свою чергу, ці СЕД є кінцевими в ієрархії змістовних СЕД документу. Тому кожен з них має значення з назвою функціональної вимоги у відповідності до інформаційного елементу, що описує цю вимогу. Службовий елемент nc представляється значенням $z(nc)$, що описує заголовок і йому можна поставити у відповідність структурний елемент змісту, або інформаційний елемент документу. Іншими словами $\forall ns \in Ns$ характерне наступне відношення:

$$ns_k : \Leftrightarrow \bigcup_{i=0}^L nc_i^k \oplus \bigcup_{j=0}^M c_j^k \quad (2.5)$$

де $ns_k \in Ns$, $nc_i^k \in Nc$, $c_j^k \in C$.

Для прикладу, структурному елементу зі значенням «Глосарій», відповідають ІЕД, що містять визначення понять предметної області. Службовому СЕД «Вступ», відповідають змістові структурні елементи «Мета», «Контекст», «Визначення, акроніми та скорочення».

Ідентифікаційні СЕД включають лише значення $z(nid)$ (наприклад, назва проекту, версія документа), які не містять залежностей та відповідності до жодного елементу.

2.2.2. Метамоделі документу опису технологій вдосконалення комп'ютерних систем

Мета-модель, що описує документ технологій вдосконалення комп'ютерних систем M_T , будується на базі моделі M_G та загальних вимог для технологій імплементації $R = \bigcap_{i=1}^N R_i$.

Мета-модель $M_T = \langle C(R^c), S(R^s), P(R^p) \rangle$ містить параметри, які визначають три складові документу: наповнення, структуру і вид подання. Параметри формуються за допомогою вимог R^c, R^s, R^p [7]. Для наповнення, в

якості параметра виступає нотація $p_{notation}$. Для структури – фрагмент дерева S , що визначається для сукупності ідентифікаторів Nid і службових Ns структурних елементів документу, $S'_1 = \langle Nid \cup Ns, E'_1 \rangle$, де $E'_1 \in E$. Параметр вид подання описується сукупністю спільних вимог до форматування документів використовуваних технологій. Таким чином, метамоделі документу набуває наступного вигляду:

$$M_T = (p_{notation}, S'_1, F) \quad (2.6)$$

Окрім спільних вимог щодо представлення документа опису технологій розробки комп'ютерних систем, до моделі M_T можуть включатися атрибути $prop = \langle p_{tech}, p_{phase}, p_{name} \rangle$, що мають відношення до процесу вдосконалення ПЗ КС. Атрибути документу характеризують назву технології p_{tech} , стадію на якій згенеровано документ p_{phase} , і саме ім'я документу p_{name} [6].

2.2.3. Модель представлення документу використовуваних технологій вдосконалення комп'ютерних систем

Модель, що інтерпретує документ для представлення аспектів використовуваних технологій вдосконалення комп'ютерних систем M_D формується на базі моделі M_T та вимог R_i , які стосуються конкретної технології. Модель $M_D = \langle C(R_i^C), S(R_i^C), P \rangle$ включає сукупність параметрів двох категорій: наповненість і структура, що задається вимогами R_i^C і R_i^S .

Наповненість документу описується за допомогою параметрів типу подання p_{type} і множини нотацій EX . Структура зображається фрагментом дерева S і є визначеною для структурних елементів Nc : $S'_2 = \langle Nc, E'_2 \rangle$, де $E'_2 \in E$, матриця A , що відображає залежності C і V та перетворення $O: C \mapsto N$.

Отже, беручи до уваги параметри моделі M_T , модель M_D набуде наступного вигляду:

$$M_D(p_{notation}, A, EX, S, O, F) \quad (2.7)$$

2.3. Метод генерації документації при вдосконаленні і модернізації комп'ютерних систем

У контексті вдосконалення комп'ютерних систем, що передбачає зміну програмного забезпечення застосовуються зворотна інженерія і генерація документації. Зворотна інженерія розглядається як підхід до відновлення інформації, який має вигляд деяких представлень, що можуть бути використані для розуміння подальшої розробки або міграції уже створеного ПЗ. До цих представлень не висувають формальних вимог, тобто не регламентуються структура представлень, їх кількість, носії і ступінь деталізації.

Генерація документації – це не лише відновлення множини даних під час вдосконалення комп'ютерних систем з уже діючого ПЗ, але і представлення її у вигляді документації, регламентованої вимогами або стандартами [3, 14].

Запропонований метод призначений для використання в реінженерії існуючого у комп'ютерній системі ПЗ. Реінженеринг передбачає виконання двох типів процесів, які належать до прямої і зворотної інженерії.

Сучасні технології реалізації комп'ютерних систем та їх компонентів формалізують процеси імплементації, стадії ЖЦ, документацію і програмні продукти, які одержують на кожній фазі. Кожен з документів розробки, регламентованих технологією, стосується певної фази ЖЦ і містить опис програмного продукту цієї фази. Тому документація, що створюється у процесі вдосконалення комп'ютерних систем і яка використовується при прямій інженерії, повинна відповідати вимогам підходу, обраного при зворотній інженерії. У такому випадку виконання методів і процесів прямої інженерії передбачає використання документації щодо реалізації і вдосконалення системи, і у зв'язку з цим інші задачі, наприклад, генерація документації для користувачів і процесів на стадіях ЖЦ у кваліфікаційній роботі не досліджуються.

2.3.1. Обґрунтування та опис методу

Суть методу автоматизованої генерації документації у процесі оновлення чи вдосконалення КС з врахуванням її застосування при реінженерії ПЗ, полягає у формуванні документації у відповідності до вимог методів та інструментів, передбачених у прямій розробці ПЗ, як показано на рис.2.4.

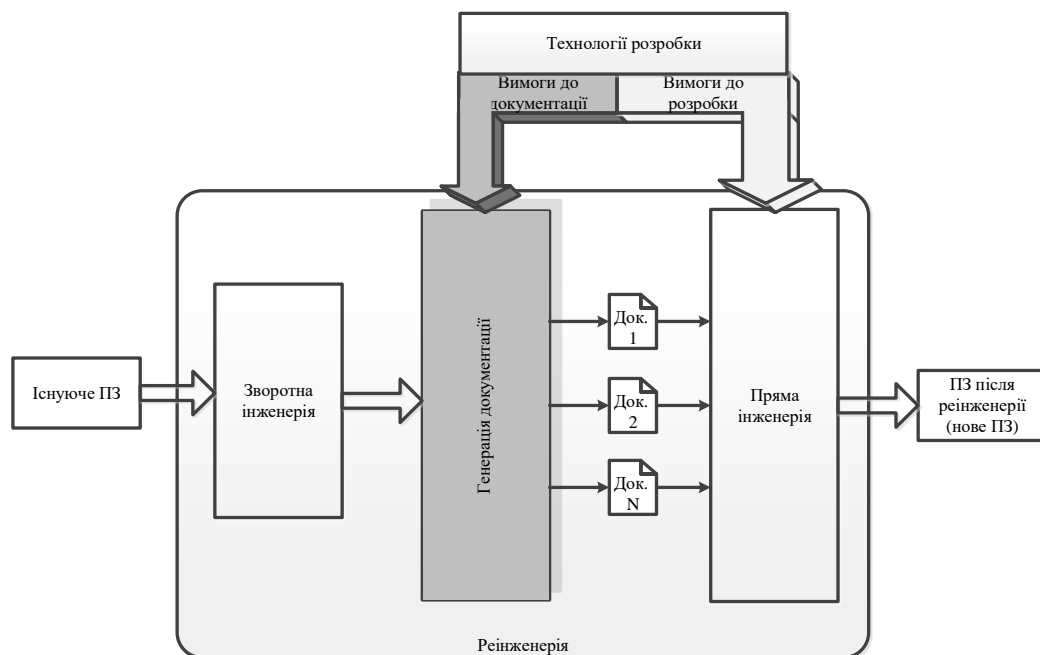


Рис. 2.4. Метод генерації документації

Застосування методів і засобів зворотної інженерії до існуючого ПЗ комп'ютерної системи залежить від стадії, коли починають використовуватися інструменти прямої інженерії. Враховуючи той факт, що будь-яка наступна стадія при прямій інженерії використовує результати попередньої, то при генерації документації доцільно сформувати документи попередньої стадії, як проілюстровано на рис. 2.4. Для виконання процедур прямої інженерії достатньо, щоб були відновленими документи лише на одній стадії реалізації ПЗ. Одержані документи належать до підмножини документів, які характеризують технологію розробки комп'ютерної системи і програмного забезпечення. Для прикладу, якщо виконується процес реінженерії з метою переведення програмного забезпечення на новішу мову програмування чи відмінну від існуючої

операційну систему, але у структури даних і в алгоритми не будуть внесені зміни, то формувати документацію потрібно лише для етапу детального проектування.

У випадку, коли передбачається повна або часткова зміна архітектури комп'ютерної системи чи програмного забезпечення, то потрібно генерувати документи початкової стадії життєвого циклу з метою встановлення тих функціональних вимог, які мають безпосередній вплив на структурні компоненти ПЗ і зв'язки між ними. Отже, процес генерації документації при вдосконаленні комп'ютерних систем передбачає врахування аспектів не лише обраної технології реалізації ПЗ, а ще й особливості стадії з якої потрібно починати його подальшу модернізацію (рис. 2.5).

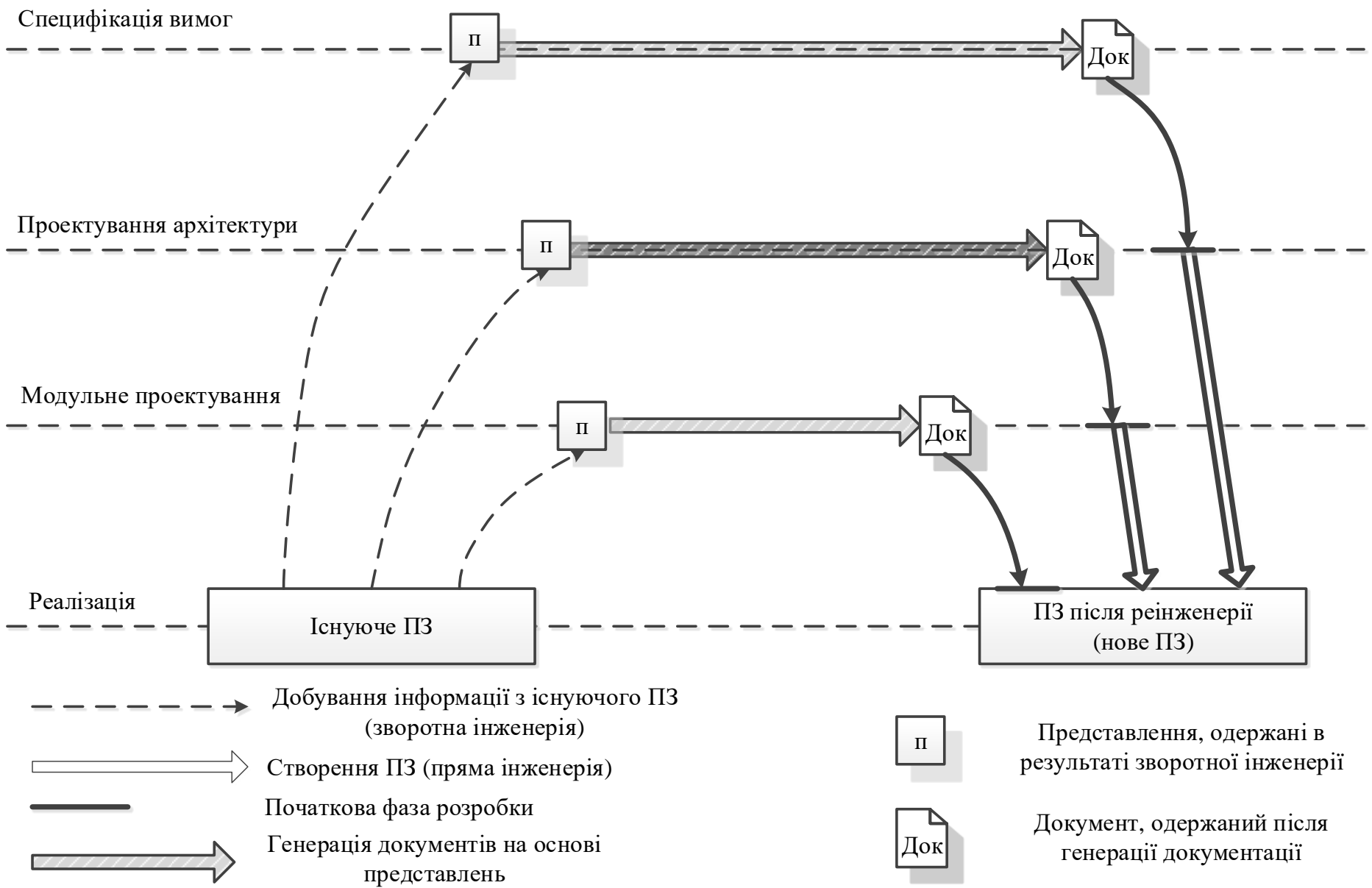


Рис. 2.5. Генерація документації ПЗ з врахуванням стадії розробки

На основі процесів зворотної інженерії, метод передбачає попереднє визначення сукупності V представлень існуючого ПЗ. Для кожного добутого представлення повинні бути визначені його P_{type} й нотація $P_{notation}$.

У методі генерація документації RD визначається процесом формування переліку документів D_{RD} шляхом перетворення моделей M документу з генерацією їх наповненості за допомогою представлень V існуючого ПЗ. Перетворення відбувається під керуванням двох параметрів: технології розробки T і етапом розробки Ph . Отже, $D_{RD} = RD(M, V, T, Ph)$.

Застосування методу не передбачає повного відновлення вмісту документації. Рівень відновленої інформації залежатиме від потужності множини V представлень наслідуваного ПЗ. Можливості відновлення визначатимуться існуючим ПЗ і наявними технологічними властивостями методів і засобів зворотної інженерії, а також особливостями технології розробки, яка застосовувалася при розробці ПЗ. Розміщення представлень у документах встановлюється значеннями матриці $A [K, L]$.

2.3.2. Реалізація методу базованого на моделях

Реалізація методу генерації документації заснована на принципах підходу MDD [4, 9] і передбачає виконання сукупності впорядкованих трансформацій моделей:

- $M_G \xrightarrow{R_i} M_T$ – перехід від моделі документу до метамоделі опису технологій, що враховує і конкретизує загальні вимоги до них і може бути реалізована із застосуванням інструментів UML [11];

- $M_T \xrightarrow{R_j} M_D$ – перехід від M_T до моделі M_D , що також може використовувати засоби UML;

- $M_D \xrightarrow{R_{ij}} m_D$ – перехід від моделі M_D до метаописів m_D у результаті встановлення в якості параметрів M_D поточних значень, визначених R_{ij}

В якості інструменту формування m_D може бути використане розширення RDDL для XML, що використовується в процесі редокументування [2].

Завершальний етап імплементації методу полягає у формуванні D_{RD} як екземплярів m_D .

Схема трансформації і переходу M_G у метамодель M_T виконується за алгоритмом приведеним на рис.2.6.

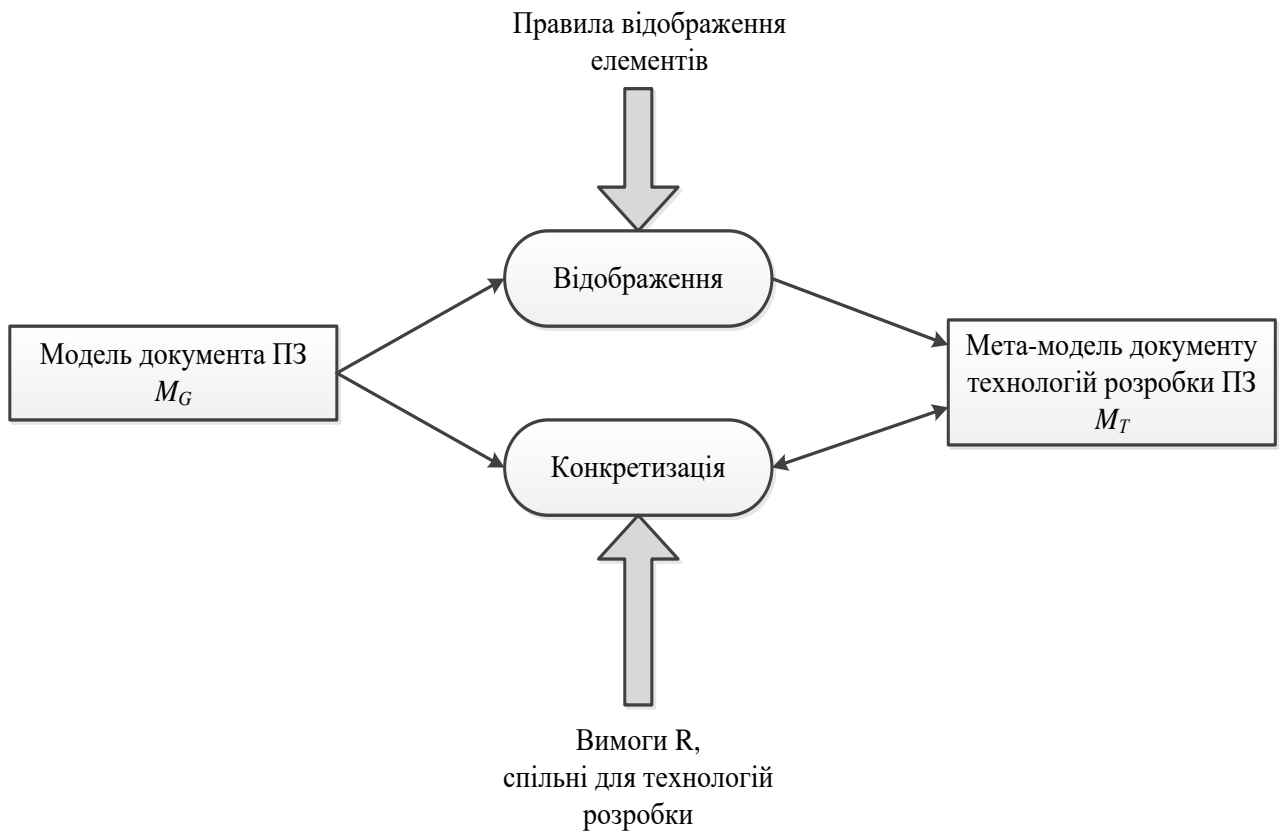




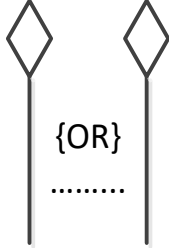


Рис 2.6. Схема перетворення моделі документа M_G в мета-модель M_T

Схема, наведена на рис. 2.6 демонструє і передбачає застосування UML при відображенні компонентів M_G у компоненти M_T , а також конкретизацію M_G враховуючи R [7]. Варто відмітити, що модель M_T може бути створена одноразово і постійно застосовуватись при побудові M_D у випадку використання однакових і незмінних вимог, які висуваються до документів опису технологій реалізації комп'ютерної системи, зокрема, програмного забезпечення. Для того, щоб побудувати модель M_G запропоновано скористатися правилами для відображення складових однієї моделі в елементи іншої моделі. Правила відображення наведено у табл. 2.2.

Таблиця 2.2

Правила відображення компонентів M_G у компоненти M_T

| Правило відображення | | Приклад відображення | |
|--|--------------------------------------|-------------------------|---|
| Елемент моделі M_G | Елемент мета-моделі M_T | Елемент моделі M_G | Елемент мета-моделі M_T |
| Елемент документу (ІЕД, СЕД, представлення, пояснення) | Клас | nc |  |
| Значення елемента документу (ІЕД, СЕД, пояснення) | Атрибут класу стрічкового типу | $\langle z(nc) \rangle$ | значення: стрічка |
| Відповідність елементів моделі | Відношення агрегації («містить») | \rightarrow |  |
| Кількість елементів | Кратність асоціації | $\bigcup_{i=1}^N$ |  |
| Об'єднання множин, які не перетинаються | Відношення узагальнення («це-є») | \cup |  |
| Операції «АБО», «виключне АБО» | Обмеження (для відношення агрегації) | \vee |  |

Структурно документ моделі M_G представляється як ієрархія, що продемонстрована на рис.2.3., а у випадку використання класів в термінах UML-нотації залежності між класами моделі M_T показані на рис. 2.7.

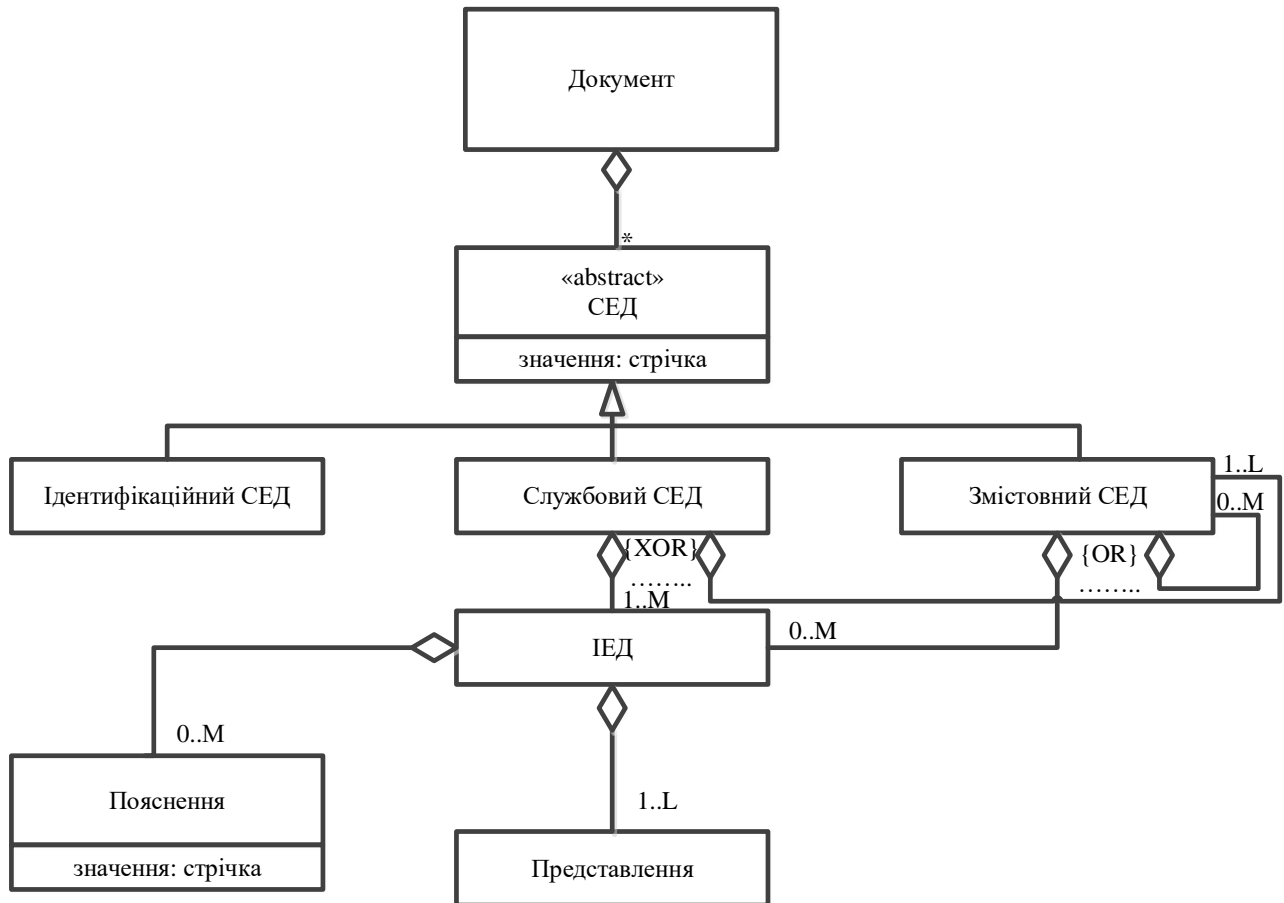


Рис. 2.7. Модель M_G в нотації UML

З рис. 2.7 видно, що коренем дерева є суперклас «Документ». Процес конкретизації моделі M_G передбачає її доповнення UML-діаграмами, які використовується при описі параметрів M_T (формула 2.6).

Метамоделі M_T документу, що описує технології реалізації при вдосконаленні комп'ютерних систем представлено на рис. 2.8. Параметр $p_{notation}$ описується типом перелічення «Нотація представлення» для визначення області значення параметра.

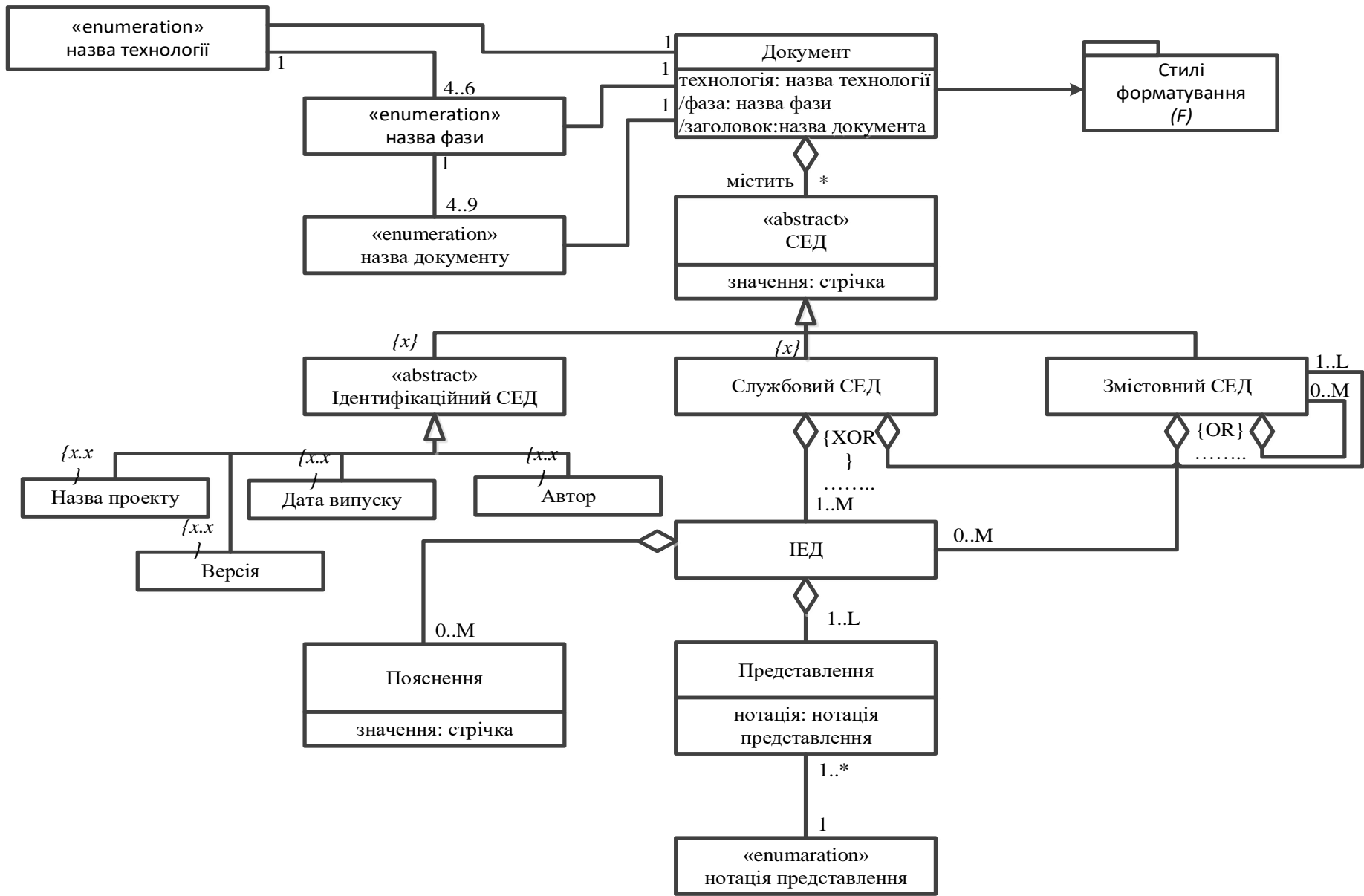


Рис. 2.8. Метамоделі M_T

Параметр S_1' описується класами, що відповідають за інтерпретацію назви проекту, його версії, дати створення та автора і дає змогу сформувати Nid - ідентифікатори структурних елементів для технологій реалізації ПЗ. Окрім цього, передбачено абстрактний клас для опису Ns , тобто допоміжних (службових) структурних елементів, які в подальшому реалізуються відповідними класами. Накладання обмежень на ребра $E_1':\{x\}$ виконується шляхом вказання індексів вузлів та відповідних ребер, з врахуванням кореня дерева та абстрактних класів. Класи, які визначають елементи форматування документів формують параметр F .

Конкретизацію метамоделі M_T виконують шляхом її довизначення елементами параметрів моделі M_D (формула 2.7) із застосування UML-нотацій, представлених на рис.2.8. У результаті одержують модель документу технології реалізації M_D , яка наведена на рис.2.9.

Параметр p_{type} визначений типом «перелічення» та інтерпретує тип представлення (візуального відображення).

Параметр $A[K,L]$ представляється у вигляді відношення агрегації між класами «ІЕД» (агрегат) і «Представлення». Кратність $1..L$ показує діапазон числа представлень ν , що відповідають ІЕД c_k .

Атрибут класу, що забезпечує пояснення визначає параметр EX .

За допомогою класів, які визначають опис технології, етапу проектування і назви документу визначають параметр $prop$.

Класи призначені для опису діапазону значень параметрів $p_{tech}, p_{phase}, p_{name}$. Означені вище класи пов'язані відношенням асоціації з кратністю, яка показує кількість етапів у технології і кількість документів у стадії.

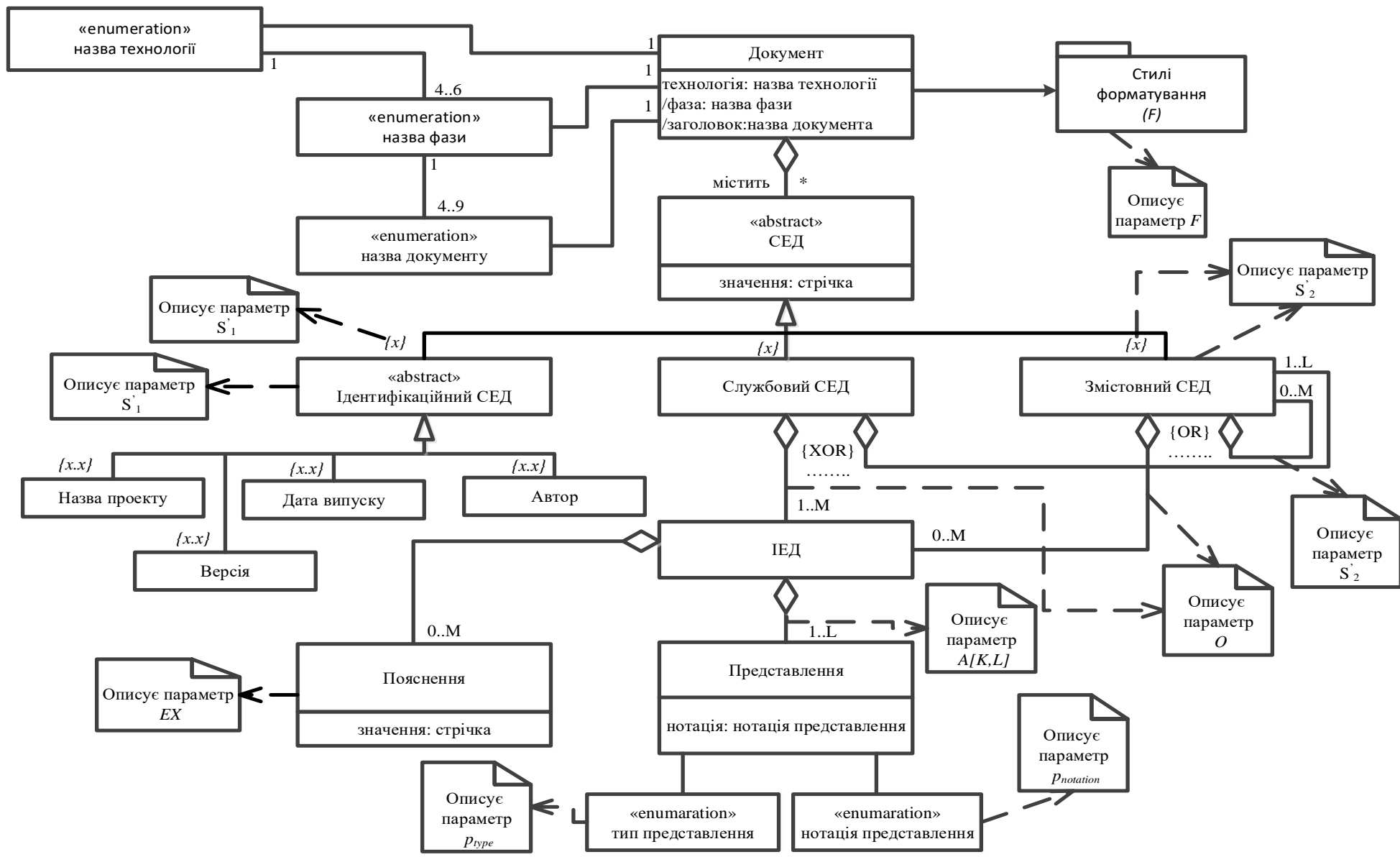


Рис. 2.9. Модель M_D

По аналогії до описаного вище алгоритму переходу від моделі M_G у модель M_T , можна побудувати перехід від моделі M_T у модель M_D .

Метаопис m_D документу є екземпляром моделі M_D , тому елементи моделі M_D на мові UML відображаються в екземпляри елементів метаопису m_D мови RDDDL на основі правил, наведених у табл.2.3.

Мова RDDDL характеризується множиною міток, множиною атрибутів та правилами опису елементів метаопису. Наприклад, при відображенні клас моделі M_D має один або декілька екземплярів в метаописі: клас «Версія» має один екземпляр у вигляді елемента метаопису з міткою *version*, а клас «Змістовний СЕД» – декілька екземплярів у вигляді елементів метаопису з мітками *item* [3].

Таблиця 2.3

Правила перетворення компонентів M_D у компоненти m_D

| Правило відображення елементів | |
|--|--|
| моделі M_D | метаописи m_D |
| Клас | Елемент |
| Назва класу | Мітка мови RDDDL |
| Атрибут типу «перелічення» | Атрибут елемента |
| Назва атрибута | Атрибут мови RDDDL |
| Атрибут типу «стрічка» | Текстовий вміст елемента |
| Відношення агрегації («містить») | Вкладеність елементів |
| Кратність асоціації | Кількість вкладених елементів |
| Обмеження (для відношень агрегації) _____ ^ _ ^ _____ | Порядок вкладеності елементів у мові RDDDL |

Створення документів наслідуваного D_{RD} – це створення екземплярів метаописів m_D , результатом якого є підмножина D_{RD} всієї множини документів технології T , яка визначається обраною фазою Ph . Заповнення документів поданнями існуючого ПЗ здійснюється шляхом інтеграції у кожен із згенерованих документів необхідних представлень ПЗ.

Формування документів наведено на рис. 2.10. Для кожної технології створюється свій набір описів m_D документів за допомогою мови RDDDL, з якого необхідні метаописи вибираються на основі параметрів керування – значень P (технологія) і Ph (фаза). У метаописі вміст документа задається у вигляді сукупності ІЕД, що містять описи одного або декількох представлень у вигляді пар властивостей «тип представлення» і «нотація представлення» $(p_{type}, p_{notation})$. Створюючи документ, значення $(p_{type}, p_{notation})$ метаопису замінюються на представлення з множини V з такими ж властивостями $(p_{type}, p_{notation})$. Вибір необхідного представлення автоматизується за рахунок застосування дескрипторів, яким забезпечується кожне подання, виявлене засобами реінженерії.

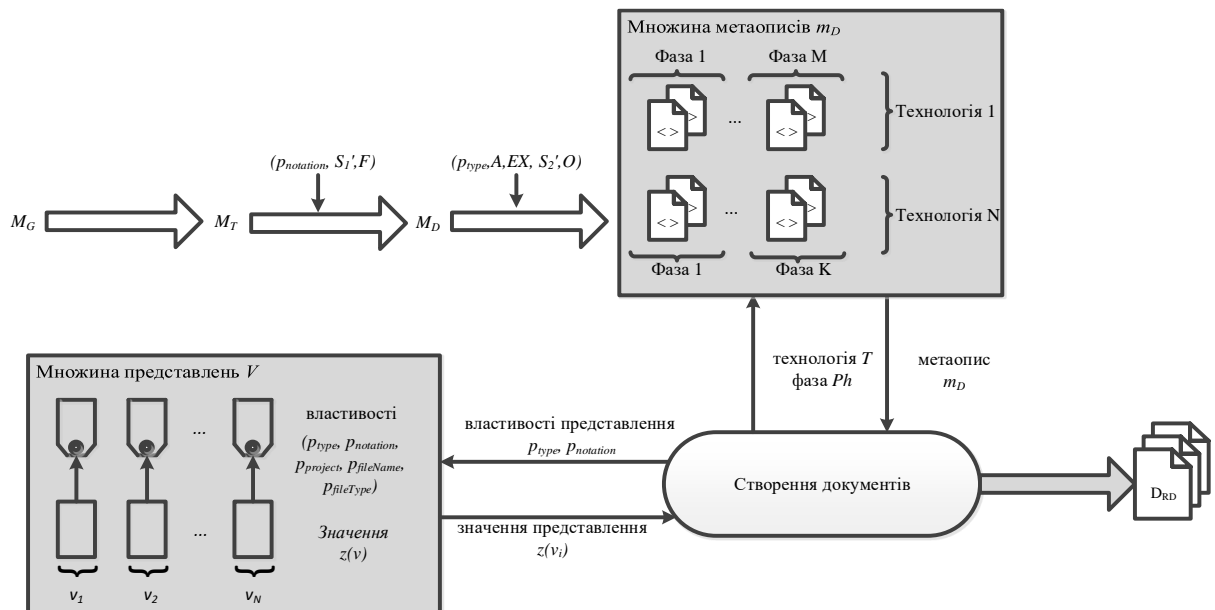


Рис. 2.10. Схема створення документів за допомогою метаописів подання існуючого ПЗ

Дескриптор містить властивості представлення: тип представлення p_{type} , нотація $p_{notation}$, ідентифікатор проекту $p_{project}$ до якого відноситься представлення, ім'я $p_{fileName}$ і тип $p_{fileType}$ файлу, що містить представлення. Для опису дескрипторів використовується спеціальна мова RVDL (Redocumentation View Description Language), розроблена на основі мови XML [7, 12]. Мова характеризується множиною міток і правилами опису елементів дескриптора.

2.4. Висновки до розділу

Основні результати полягають в наступному:

1. Формалізовано та обгрунтовано моделі для представлення документації програмного забезпечення у процесі вдосконалення комп'ютерних систем, що дало змогу побудувати ланцюжок перетворення однієї моделі документу в іншу на трьох рівнях абстракції і врахувати властивості використовуваних технологій реалізації компонентів комп'ютерних систем.

2. Створено метод автоматизованої генерації документів, який враховує вимоги технологій вдосконалення комп'ютерних систем і дає змогу формувати набір документації для майбутнього оновленого програмного забезпечення та набори його представлення, що керуються параметрами конкретної технології й етапом початку проведення відновлення документації.

3. Побудовано та імплементовано процедуру автоматизованої генерації документації з використанням підходу Model Driven Development, що передбачає виконання впорядкованих трансформацій над моделями документу і, як наслідок, формування документації оновленого ПЗ у вигляді екземплярів метаописів.

РОЗДІЛ 3

ПРОЕКТУВАННЯ АРХІТЕКТУРИ ІНСТРУМЕНТУ АВТОМАТИЗОВАНОЇ ГЕНЕРАЦІЇ ДОКУМЕНТАЦІЇ ПРИ ВДОСКОНАЛЕННІ КОМП'ЮТЕРНИХ СИСТЕМ

3.1. Концептуальна архітектура інструменту автоматизованої генерації документів

Функціональністю запропонованого інструментального засобу автоматизованої генерації документів в процесі вдосконалення комп'ютерних систем передбачено етапи, які відображено на рис.3.1.

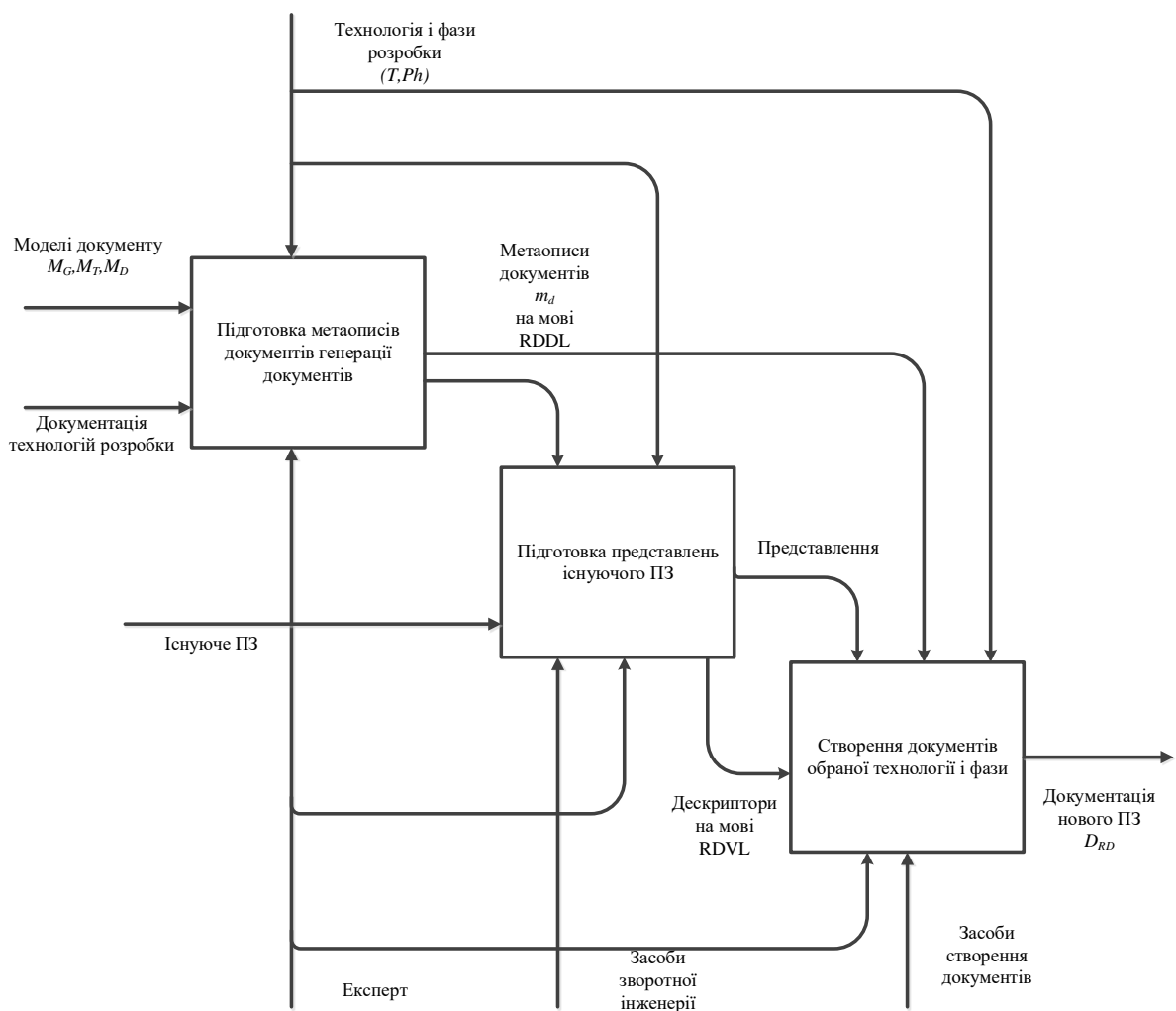


Рис. 3.1. Функціональна архітектура інструментального засобу автоматизованої генерації документів в процесі вдосконалення КС

На першому етапі виконуються процедури щодо підготовки опису властивостей технологій реалізації ПЗ КС з метою формування нової документації. Другий етап передбачає формування матеріалів щодо різних представлень майбутнього програмного забезпечення, а завершальний етап забезпечує безпосередню генерацію документації.

Процеси першого рівня забезпечують ввід фахівцями метаописів з використанням мови RDDL у вигляді запропонованих у другому розділі моделей і документів, що описують технології реалізації комп'ютерної системи.

Реалізація процесів другого рівня функціональної архітектури повинна підтримувати формування тегів подання програмного забезпечення, які одержані із застосуванням засобів зворотної інженерії. Окрім цього, передбачається комплексне збереження використаних тегів з відповідним поданням у сховищі даних.

Третій рівень реалізує автоматичне формування документів, що враховує метаописи згенерованої структури і одержані на попередньому рівні представлення. Форматування документів відбуваються на основі обраного стилю.

Структурну схему (архітектуру) компонентів інструментального засобу автоматичної генерації документів при вдосконаленні комп'ютерних систем проілюстровано на рис.3.2.

Для опрацювання запитів і даних, які вносить фахівець з генерації документації, використовується підсистема управління, яка ще й відповідає за узгодженість та взаємодію між іншими компонентами системи.

Функції користувачького інтерфейсу фахівця при генерації документів полягають у забезпеченні ефективного діалогового режиму.

Для формування і підготовки метаописів наявної документації, а також налаштування параметрів процесу генерації призначений редактор метаописів.

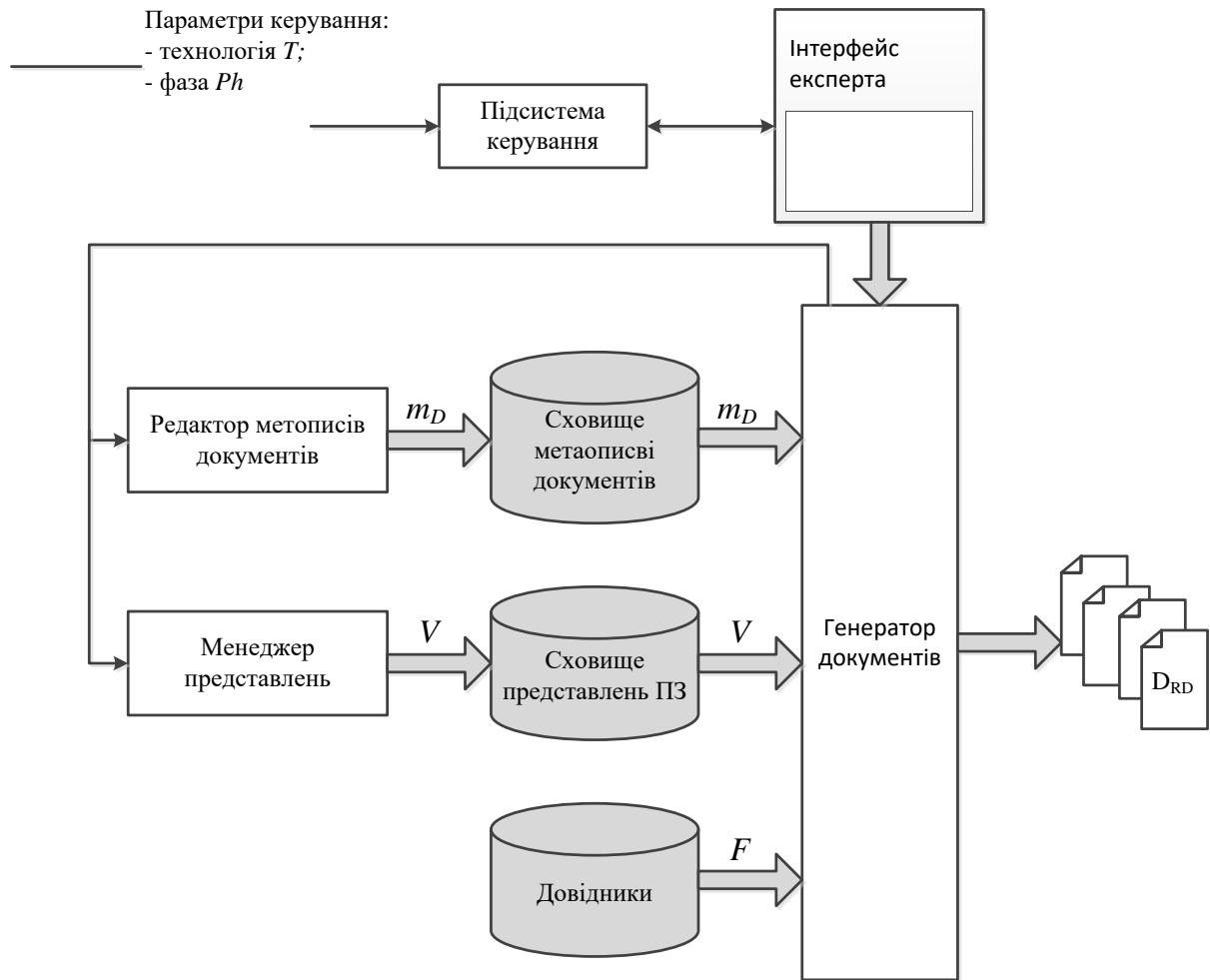


Рис. 3.2. Архітектура засобів автоматизованого синтезу супровідної документації програмного забезпечення

Для забезпечення функціональності, пов'язаної зі створенням і визначенням представлень документів, автоматизованого формування тегів їх опису, а також управління процесом реінженерії використовується менеджер представлень.

Підсистема генерації формує документи за описом обраної технології реалізації комп'ютерної системи і вказаної стадії відновлення документів з врахуванням їх метаописів.

Для зберігання даних, важливих в контексті відновлення та формування нової документації при вдосконаленні комп'ютерних систем, використовуються довідники, які можуть бути представлені у вигляді

реляційних структур даних, а також сховища, які зберігають інформацію у форматі XML.

3.2. Характеристика компонентів архітектури засобу автоматизованої генерації документів

Для реалізації архітектури засобу автоматизованої генерації документів розроблені компоненти, які наведено на рис. 3.2.

Інтерфейс експерта призначений для взаємодії експерта із засобами автоматизованої генерації документації і складається з чотирьох функціональних компонентів, показаних на рис.3.3.

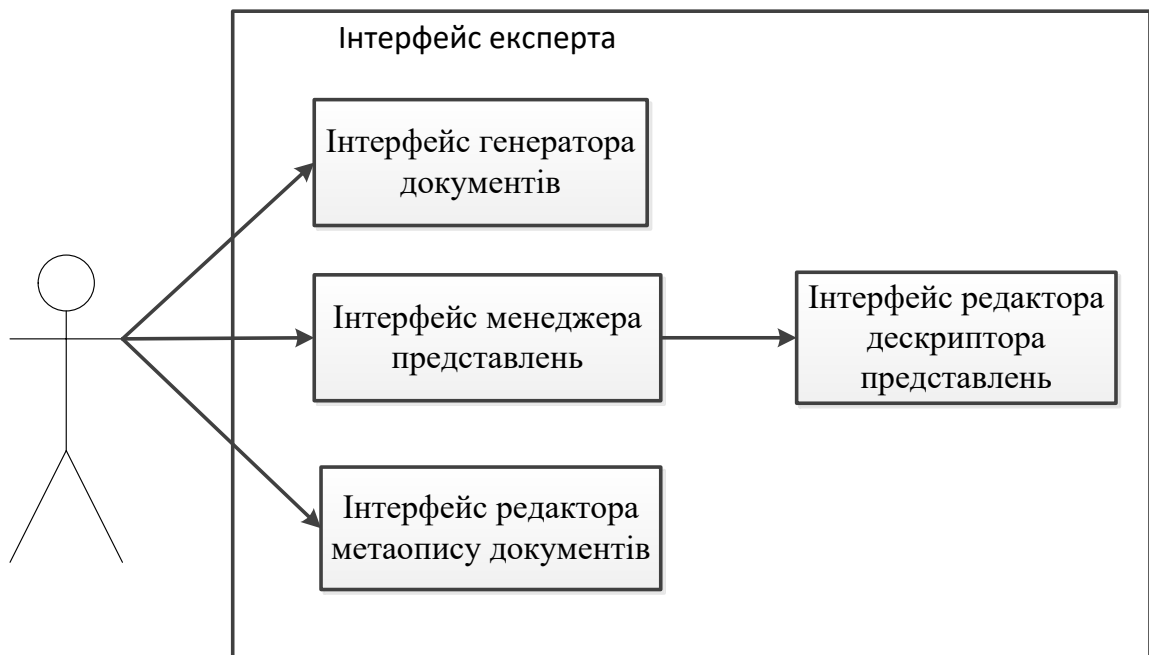


Рис. 3.3. Компоненти інтерфейсу користувача

На інтерфейс генератора документів покладено наступні функції:

- ввід вхідних даних для проведення генерації документації (назва проекту, назва технології розробки і її фази, формат вихідної документації);
- вивід списку метаописів документів, які відповідають обраній технології і фазі та знаходяться у сховищі метаописів;

- ініціалізація процесу генерації документів;
- вивід ідентифікаційних даних для вихідних документів (назва проекту, ім'я автора і номер версії).

Інтерфейс менеджера представлень забезпечує виконання таких функцій:

- вивід сукупності представлень, які є важливими для конкретної технології та етапу реалізації комп'ютерної системи;
- вивід списку представлень, які знаходяться в сховищі представлень у каталозі відповідного проекту;
- вивід реєстру доступних засобів зворотної інженерії;
- вивід даних розміщення виділеного представлення та дескриптора;
- взаємодія з редактором дескрипторів представлень.

Інтерфейс редактора тегів представлень є частиною інтерфейсу менеджера представлень і забезпечує можливість створення і зміни тегів визначених представлень.

Інтерфейс корекції метаописів забезпечує доступність функцій:

- вводу значень елементів і атрибутів метаопису документу;
- перегляду і внесення зміни у метаопис документів, які знаходяться у сховищі.

3.2.1. Підсистема управління генерацією документів

Підсистема управління опрацьовує звернення, які формуються в інтерфейсі експерта, і організовує коректне функціонування інших компонентів, які входять до складу запропонованої архітектури. Підсистема управління реалізує наступні функції:

- авторизація експерта перед початком роботи із засобами;

- реалізація логіки спільного використання засобів, у тому числі використання процедур авторизації, запуск генератора документів, запуск менеджера виділення представлень, запуск редактора метаописів, збереження метаописів, представлень і дескрипторів представлень.

3.2.2. Редактор метаописів при автоматичній генерації документації

Редактор підтримує створення метаописів документів із застосуванням мови RDDL, забезпечуючи вирішення наступних задач:

- відображення схеми метаопису з наперед визначеними назвами міток, атрибутів і правил розташування елементів;
- відображення та редагування довідкової інформації: назви технологій, фаз і документів технологій, типи і нотації представлень;
- обробка схеми метаописів у діалоговому режимі, для перевірки на коректність додавання і видалення елементів і їх вмісту;
- збереження метаописів у сховище;
- перевірка метаописів на основі XSD-схеми при збереженні описів у сховище.

Вимірювання метаопису документа для визначення значень таких реквізитів: кількість змістовних СЕД в метаописі (КСЕДМ), кількість змістовних СЕД з атрибутом *iterative* в метаописів (КСЕДМ_I), кількість представлень в метаописі (КПМ).

Створення метаопису виконується експертом шляхом перетворення моделі M_D документа обраної технології.

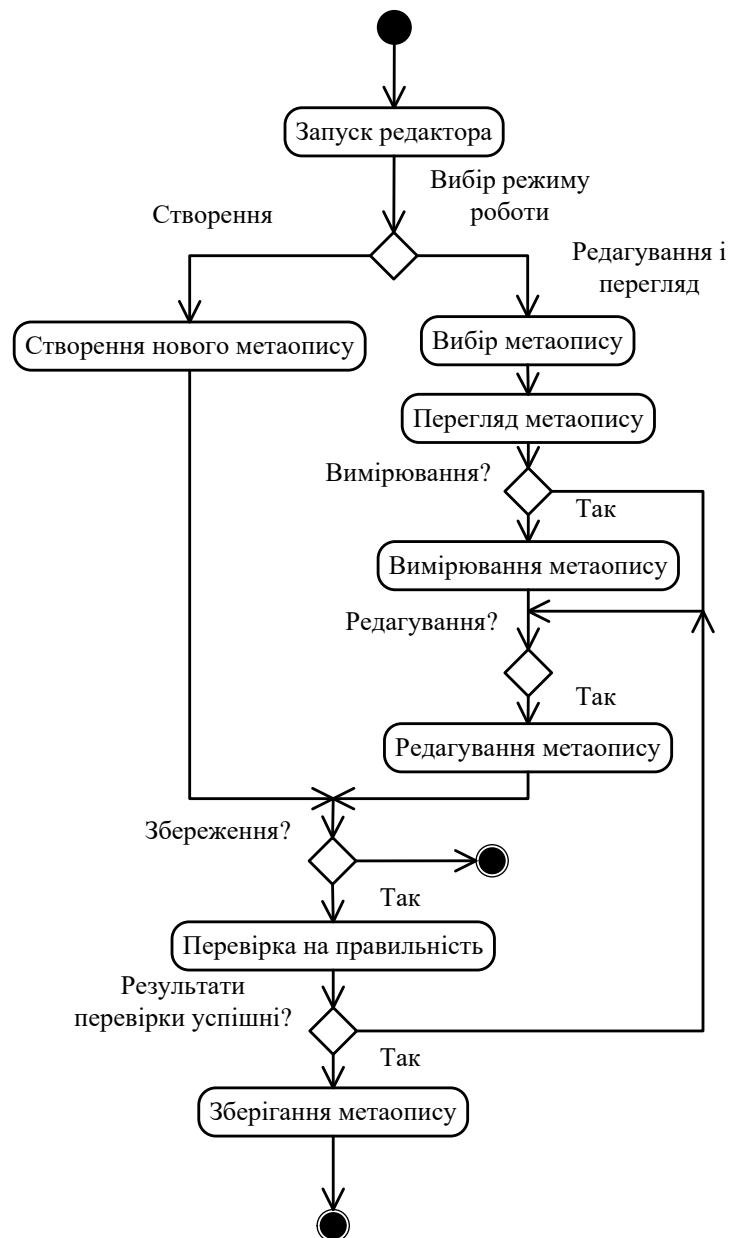


Рис. 3.4. Логіка роботи редактора метаописів

Редактор надає схему і засоби для створення метаопису шляхом виконання наступних операцій:

- встановлення властивостей документа *technology*, *phase* і *title* шляхом присвоєння значень відповідних атрибутів;
- опис наступних СЕД:
 - а) ідентифікаційних: *projectName*, *date*, *version*, *author*, які розташовуються згідно послідовності, описаної в моделі M_D ;

б) службових: `tableofContents`, `glossary`, `introduction`, `listOfFigures`, `listOfTables`, `index`, `bibliography`, `appendix` - розташованих згідно послідовності, описаної в моделі M_D ;

в) змістовних – `item`, які розташовуються у кореневому елементі та по рівнях основних СЕД у відповідності до вимог технології до документу. Для тих СЕД, які можуть повторюватися в документі, встановлюється атрибут `iterative`;

– опис вмісту документа. Для кожного змістовного СЕД обов'язковим є його назва;

– формування відповідності змісту документа технології та представлень. При цьому необхідно забезпечити:

а) додавання ІЕД (`infoItem`) у вміст основних СЕД (`item`);

б) додавання міток пояснень (`explanation`) і представлень (`view`) у вміст ІЕД;

в) ввід тексту пояснень;

г) встановлення значень атрибутів представлень (`view`): `type` – тип представлення та `notation` – нотація представлення.

– перевірка створеного метаопису на правильність на основі XSD-схеми;

– збереження метаопису у сховище. Метаописи документу «Документ архітектури програмного забезпечення», побудовані за допомогою редактора.

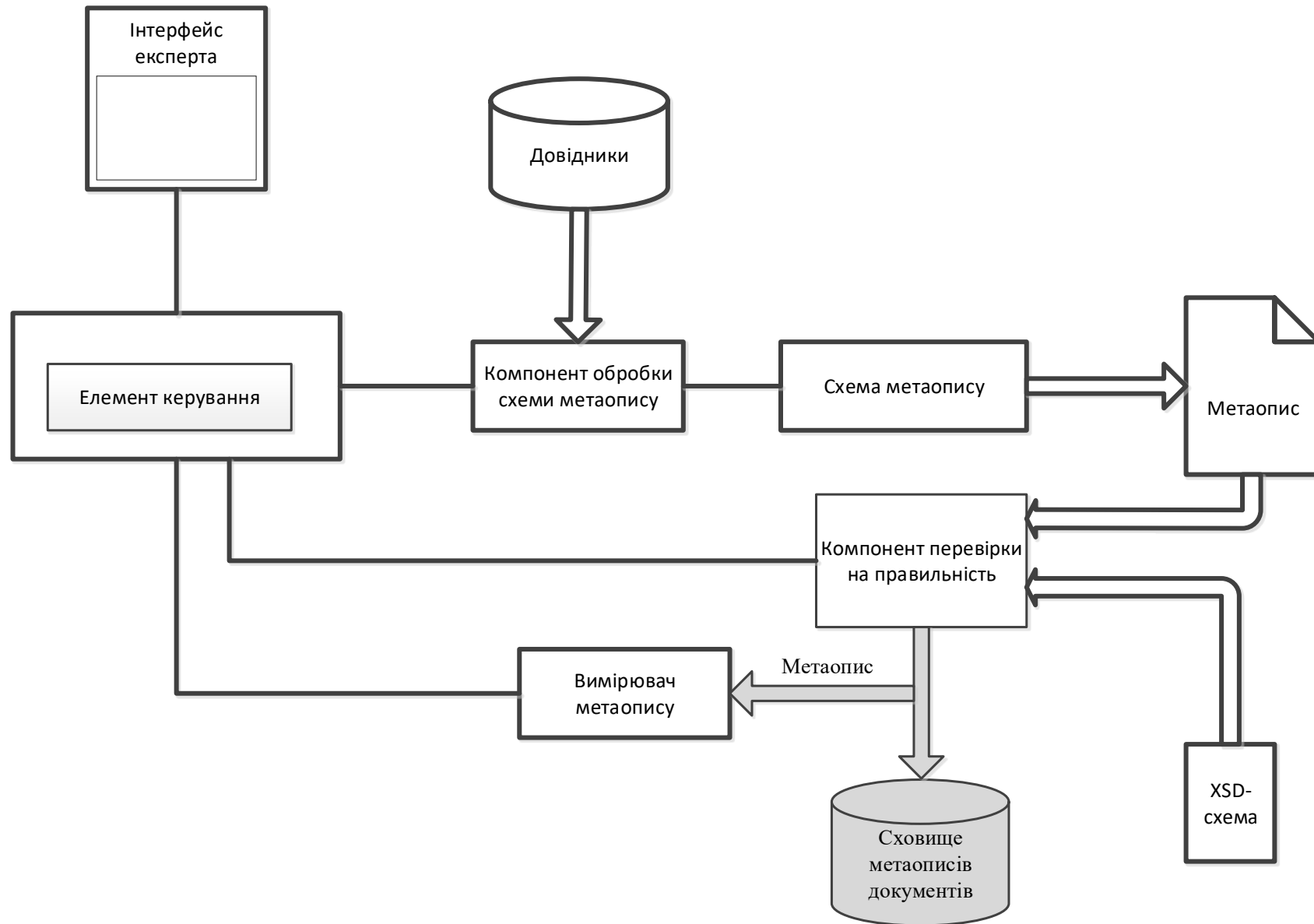


Рис. 3.5. Редактор метаописів

3.2.3. Менеджер представлень

Менеджер представлень використовується експертом на другому етапі синтезу супровідної документації для підготовки представлень, виділених засобами зворотної інженерії, яке полягає у формуванні переліку необхідних в метаописі представлень, створенні дескрипторів представлень і збереженні отриманих представлень і їх дескрипторів в сховище (рис.3.6). Архітектура менеджера представлень представлена на рис. 3.7.

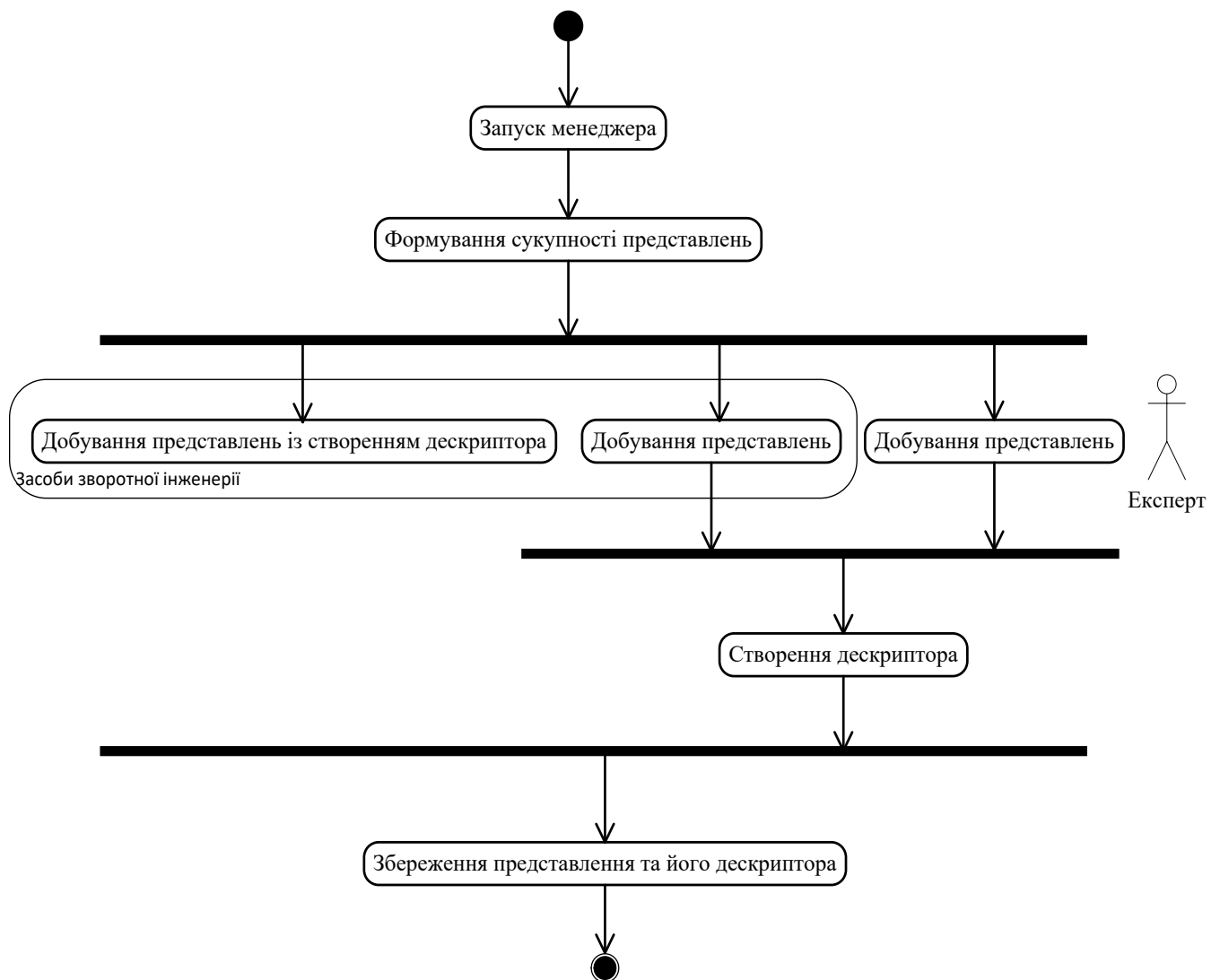


Рис. 3.6. Архітектура менеджера представлень

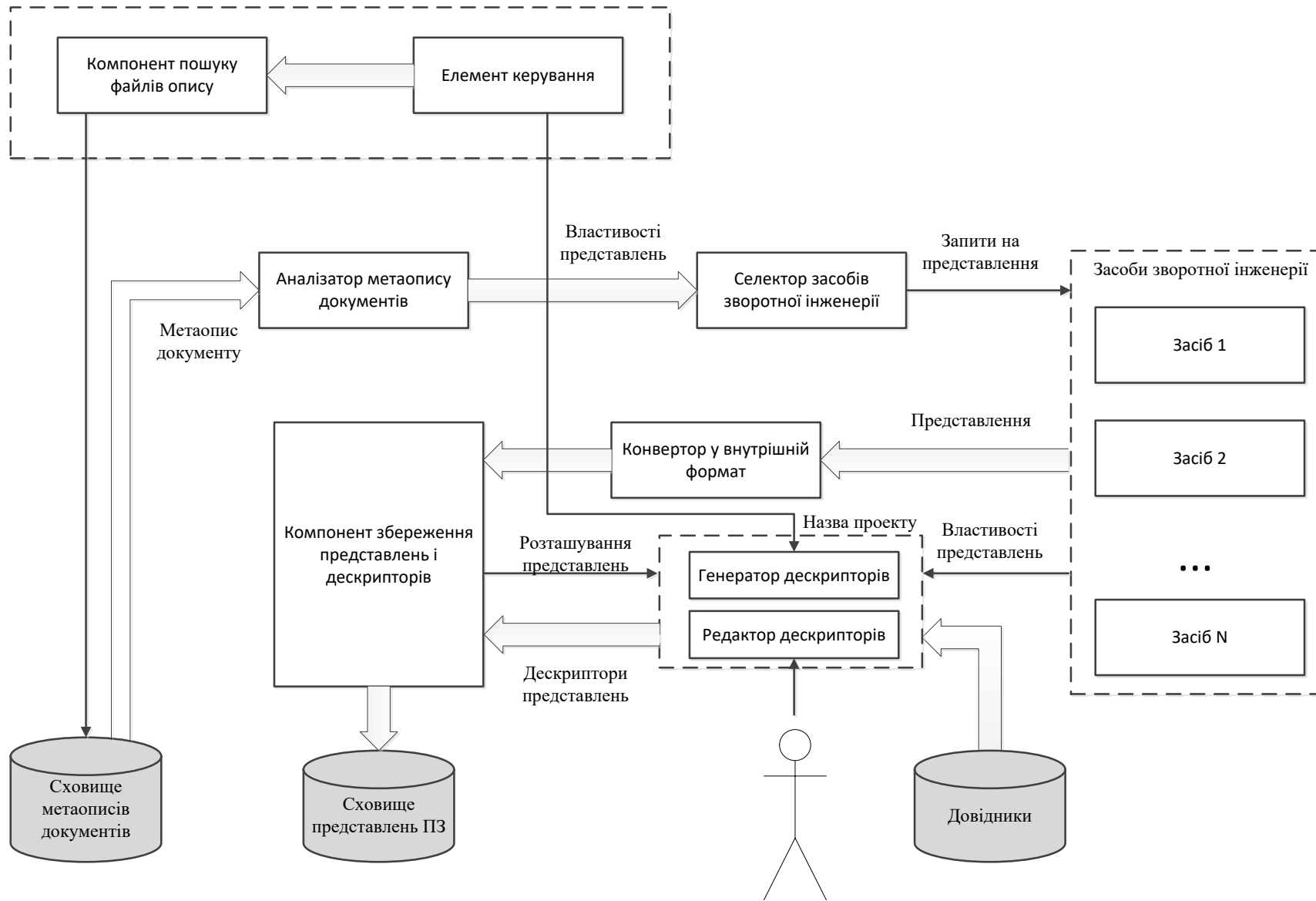


Рис. 3.7. Архітектура менеджера представлень

Аналізатор метаопису документу формує перелік необхідних для синтезу документації представлень. Для цього в обраному метаописі проводиться пошук елементів з міткою `view`, значення атрибутів `type` і `notation` яких, визначають типи і нотації необхідних представлень. Перелік необхідних представлень для метаопису документу «Документ архітектури програмного забезпечення» (додаток К) наведено на рис. 3.8

| | |
|----------------------------------|---------------|
| listOfViews | text |
| architectureGoals | text |
| architectureConstraint | text |
| useCaseDiagram | UML |
| useCaseDescription | text |
| packageDiagram | UML |
| HstOfPackages | text |
| subsystemsDiagram | figure |
| packageSpecification | text |
| listOfClasses | text |
| classDiagram | UML |
| classSpecification | text |
| sequenceDiagram | UML |
| collaborationDiagram | UML |
| processView | figure |
| interruptsExceptions | text |
| deploymentDiagram | UML |
| networkConfiguration | figure |
| structureDiagram | figure |
| componentDiagram | UML |
| HstOfDBtables | text |
| schemeDB | ER |
| classDBDiagram | UML |
| systemRequirements | text |
| nonfunctionalRequirements | text |

Рис. 3.8. Приклад сукупності необхідних представлень

Представлення можуть добуватись вручну або засобами зворотної інженерії. Експерт обирає і запускає за допомогою селектора засоби зворотної

інженерії. Селектор надає реєстр доступних засобів, у якому дано типи і нотації видобутих представлень. Добуті обраним засобом представлення надходять на вхід конвертора, де вони перетворюються в текстовий (.xml) або графічний (jpg) формат. Компонент збереження представлень і дескрипторів поміщає добуте представлення до сховища. Інформація про місце розташування необхідна для створення дескриптора цього представлення. Менеджер добування представлень містить генератор дескрипторів для автоматичного і редактор дескрипторів для ручного їх створення. Генератор автоматично створює дескриптори на основі властивостей представлень, отриманих від інструментів зворотної інженерії. Редактор надає інтерфейс, враховує зумовлені мітки і правила мови, для ручного створення дескриптора експертом. Приклад дескриптора представлень для діаграми класів проекту "SkyPassword" наведено на рис. 3.9.

```
<? xml version = "1.0" encoding = "utf-8"?>
<descriptor>
    <type> classDiagram </type>
    <notation> UML </notation>
    <project> SkyPassword </project>
    <fileName>C:\SkyPassword\classDiagram1</fileName>
    <fileType> jpg </fileType>
</descriptor>
```

Рис. 3.9. Дескриптор представлення

Створений дескриптор розміщується разом з представленням, що виконується компонентом збереження представлень і дескрипторів.

3.2.4. Генератор документів

Генерація документів здійснюється на основі метаописів, підготовлених на першому етапі автоматизованої генерації документації. Важливою функціональністю генератора є те, що він може формувати попередній перегляд документів у деякому проміжному форматі, а в подальшому генерувати обраний користувачем стиль документу (рис. 3.10).

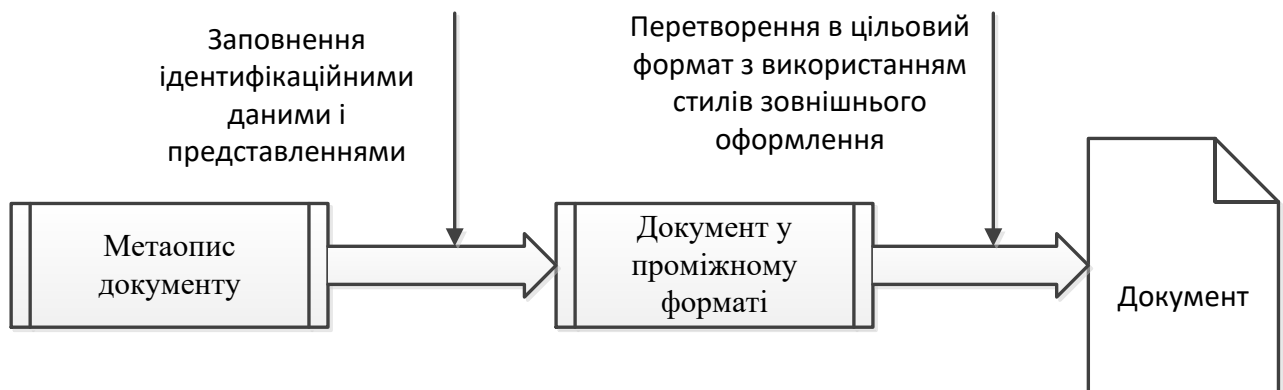


Рис. 3.10. Схема генерації документу

Архітектура генератора документів побудована на архітектурному партерні «Канал-фільтр» [6] (рис. 3.11). Канали забезпечують передачу метаописів, які послідовно обробляються фільтрами. Метаописи, які використовуються для генерації документів, вибираються зі сховища метаописів за запитом, який містить параметри пошуку: назва технології розробки і фази ЖЦ.

Знайдений метаопис поступає на вхід аналізатора допустимості документа і перевіряється на правильність відповідно до XSD-схеми (додаток Д). Перевірені метаописи проходять три компоненти (фільтри): генератор ідентифікаційних СЕД, генератор ІЕД, перетворювач документа з проміжного формату в цільовий. Генератор ідентифікаційних СЕД формує їх вміст, використовуючи дані, які вводяться через інтерфейс експерта.

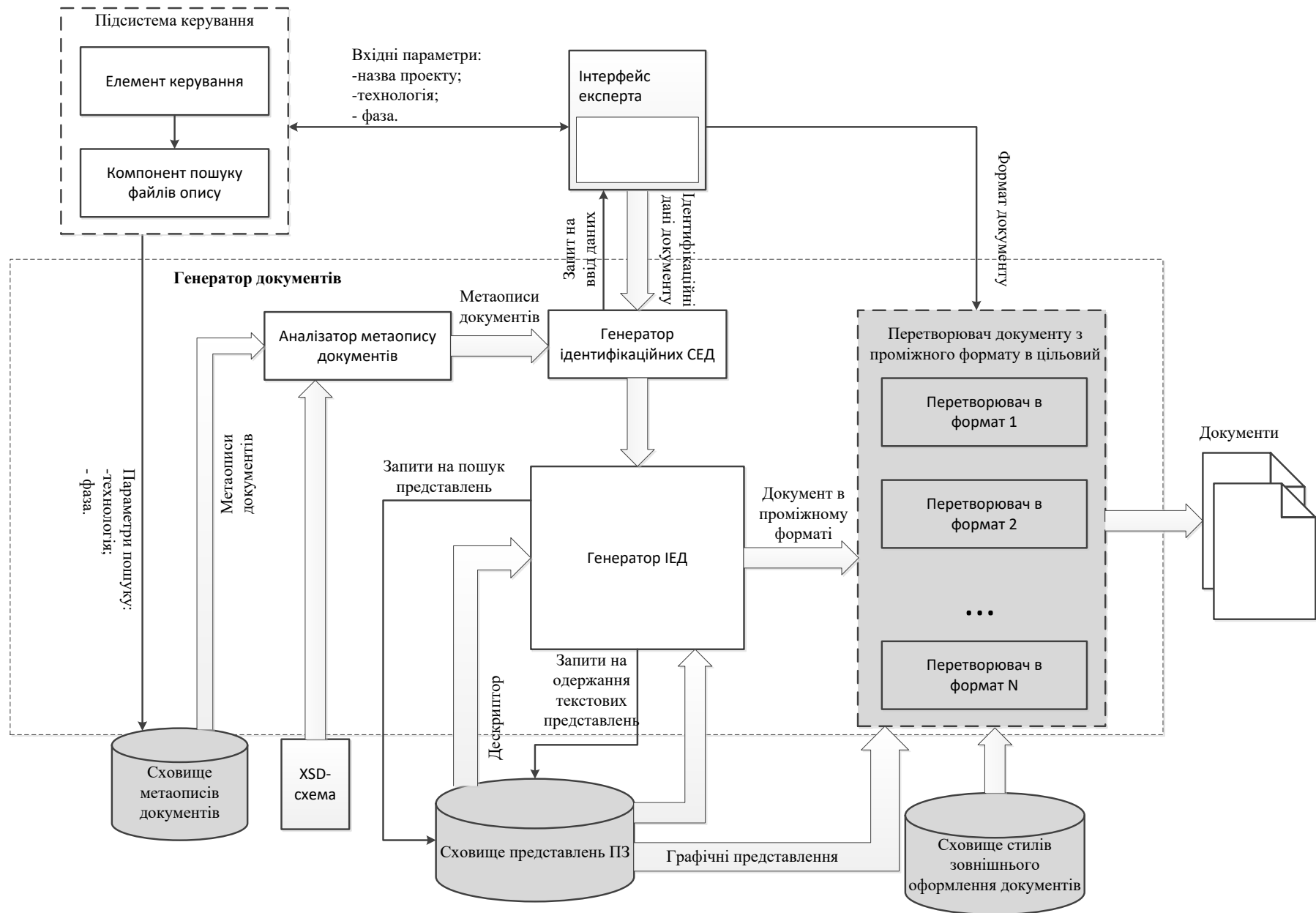


Рис. 3.11. Архітектура «канал-фільтр» при генерації документів

Генератор ІЕД призначений для формування ІЕД документів. Метаопис, що поступає на вхід генератора, сканується з метою пошуку елементів з міткою view. Значення атрибутів (тип і нотація представлення) знайденого елемента і назва проекту включаються в запит на пошук необхідного представлення в сховище. Якщо у відповідь на запит знайдено текстове представлення (тобто значення властивості filetype в дескрипторі xml), то значення представлення включається в метаопис в якості вмісту відповідного елемента з міткою view. Якщо у відповідь на запит знайдено графічне представлення (тобто значення властивості filetype в дескрипторі jpg), то в метаопис включається посилання на файл, який містить значення цього представлення.

На виході генератора ІЕД видається документ в проміжному форматі, у вигляді XML-розмітки, що містить всю структуру документа, текстові представлення наслідуваного ПЗ та інформацію про місцезнаходження графічних представлень.

Перетворювач документа з проміжного формату забезпечує одержання документів в цільовому форматі, вибраному експертом. Для цього він виконує наступні дії:

- перетворення XML-структури документа в структури, визначені цільовим форматом;
- включення в документ графічних представлень;
- форматування документа відповідно до заданих стилями зовнішнього оформлення. На виході перетворювача видаються документи обраної технології і фази, які є результатом автоматизованої генерації документації.

3.2.5. Довідники у системі автоматичної генерації документації

У системі автоматизованої генерації документів створено довідники, які забезпечують зберігання даних і мають структуру, як показано на рис. 3.12.

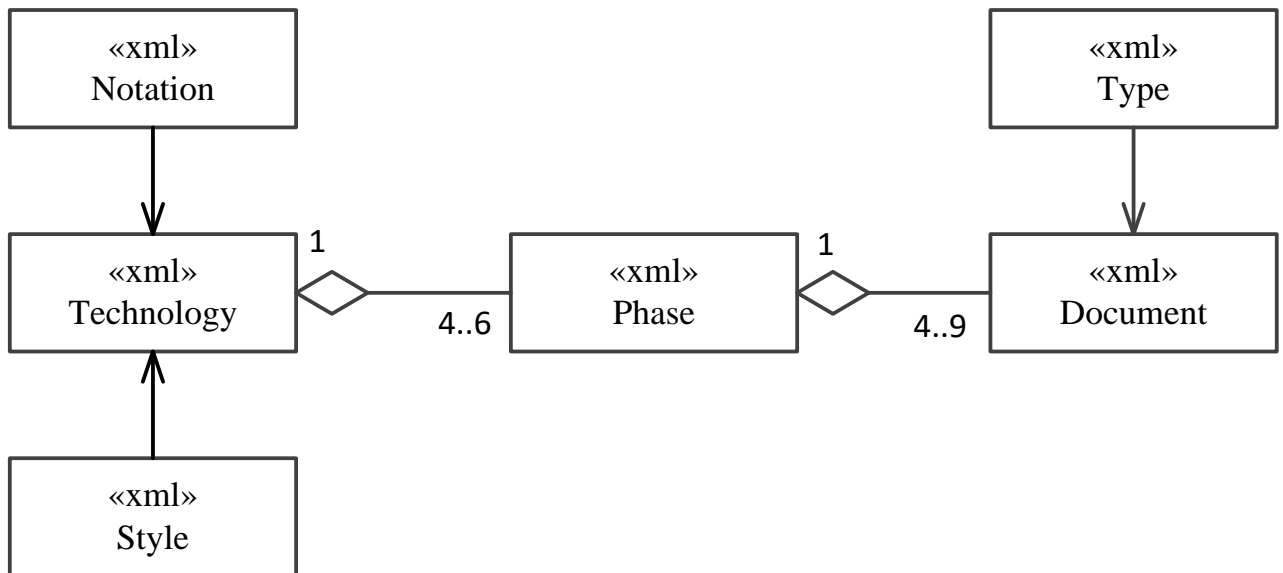


Рис. 3.12. Схема організації довідників

Довідники представляють собою XML-файл, який містить елементи, створені як екземпляри таких класів:

- Notation – нотація представлення;
- Technology – назва технології розробки;
- Style – стиль зовнішнього оформлення документа;
- Phase – назва фаз технології;
- Document – назва документу технології;
- Type – тип представлення.

Відношення агрегації між класами встановлює вкладеність елементів XML-файлу.

3.3. Висновки до розділу

1. Розроблено алгоритм функціонування засобу автоматизованої генерації документації при вдосконаленні комп'ютерних систем на основі послідовного виконання трьох процесів: формування опису документів використовуваних технологій при вдосконаленні системи, створення представлень майбутнього програмного забезпечення та процесу генерації документів з врахуванням конкретної технології і стадії початку відновлення документації.

2. Побудовано архітектуру інструментального засобу генерації документів при вдосконаленні та модернізації комп'ютерних систем, що дало змогу забезпечити автоматизацію процесів, характерних для запропонованого методу .

3. Реалізовано засобами мови програмування Java і технології XML структурні елементи архітектури засобу автоматизованої генерації документації, що дало змогу забезпечити ефективність підготовки та управління даними і на основі шаблону «каналів і фільтрів» автоматично генерувати необхідні документи.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

У дипломній роботі магістра проводиться дослідження методів і засобів автоматизованої генерації документації програмних компонентів комп'ютерних систем. Обов'язковим елементом дослідження є визначення та аналіз вимог з охорони праці і техніки безпеки при розробці програмного засобу і проведенні експериментальних досліджень, що супроводжується використанням комп'ютерної техніки. Дотримання норм і правил охорони праці є важливим аспектом у контексті дотримання норм організації робочого місця, забезпечення комфортних та зручних умов праці осіб, які беруть участь у процесі, а це вимагає дослідження та дотримання вимог з охорони праці.

В Україні розроблено й діють ряд нормативних документів, які визначають вимоги і правила щодо використання комп'ютерної техніки, приміщень з екранними пристроями та ін. Основним нормативним документом при використанні комп'ютерної техніки є НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Він регламентує, що приміщення для експлуатації комп'ютерної техніки повинно розміщуватися в північній або північно-східній частині будівлі. Площа одного робочого місця повинна становити щонайменше 6 м², об'єм — щонайменше 20 м³, відстань між робочими столами — щонайменше 2,5 м у ряду і 1,2 м між рядами. Стіни приміщень потрібно фарбувати у пастельні тони з коефіцієнтом відбиття 0,5-0,6 [20].

З метою зменшення напруження очей потрібно, щоб відстань між краями сусідніх точок зображення на моніторі не перевищувала гранично оптимальний розмір літеро-цифрових знаків – 16-20, складних знаків – 35-40. Оптимальні співвідношення параметрів літер і цифр такі: ширина знака – 0,75 їх висоти, товщина ліній при зворотному контрасті – 1/6-1/8, відстань між

знаками — 0,25-0,5 висоти знака, між словами – 0,75-1, між рядками – 0,5-1 [20].

Для профілактики загальної втоми і особливо зорового аналізатора важливе значення має організація режиму праці та відпочинку. Загальна тривалість робочого дня не повинна перевищувати 8 год. Частота і тривалість перерв залежать від типу та інтенсивності виконуваних робіт. Під час робіт, які виконуються з великим навантаженням, рекомендуються перерви на 10-15 хв. через кожну годину, а при неінтенсивній і монотонній роботі — на 10-15 хв. через кожні дві години. Кількість мікропауз (тривалістю до хвилини) потрібно регулювати індивідуально.

Зміст регламентованих перерв може бути різний: виробнича гімнастика (вправи для очей, гімнастика, спрямована на корекцію вимушеної робочої пози, поліпшення венозного кровообігу, часткову дисфункцію рухової активності), альтернативна допоміжна робота, приймання їжі тощо.

Для того, щоб особи, які займаються дослідженням методів автоматизованої генерації документації при вдосконаленні комп'ютерних систем меншою мірою втомлювались і зберігали високий рівень працездатності, потрібно раціонально організувати їхні робочі місця. Зокрема, робоче місце має відповідати основним антропометричним даним людини. Крісло або стілець на робочому місці повинні мати висоту сидіння 40-50 см від рівня підлоги, а також відповідний кут нахилу спинки.

Монітори потрібно розміщувати на висоті рівня очей (висота від підлоги до нижнього краю екрана має становити 95-100 см) на відстані 60-70 см від оператора (відстань від краю столу — 50-70 см). Кут зору працюючого щодо екрану має дорівнювати 10-20°, але не більше 40°, кут між верхнім краєм монітора і рівнем очей користувача має становити менш як 10°. Найдоцільніше розміщувати екран перпендикулярно до лінії погляду користувача. Кут нахилу екрана по вертикалі має становити 0-30° [26]. З цією метою сучасні монітори комплектують підставкою з поворотним кронштейном, що дає змогу регулювати кут нахилу монітора і горизонтально

обертати його навколо вертикальної осі. Висоту екрана від поверхні підлоги регулюють змінюючи висоту робочої поверхні столу. Іноді монітори встановлюють на спеціальні підставки, що уможлиблює його переміщення у просторі у вертикальному та горизонтальному напрямках.

У приміщеннях, де виконуються роботи на ПК, повинно бути передбачене природне і загальне штучне освітлення. Робочі місця користувачів потрібно розміщувати так, щоб у поле зору не потрапляли вікна і освітлювальні прилади (монітори потрібно розміщувати під кутом 90-105° до вікон і на відстані 2,5-4 м від стін і віконних прорізів). У поле зору користувача не повинні потрапляти поверхні, що відбивають світло. Покриття столу має бути матовим з коефіцієнтом відбиття 0,25-0,4.

Для штучного освітлення приміщення рекомендується застосовувати світильники матового світла з розсіювачами, а спектральний склад ламп має наближатися до спектру сонячного світла (наприклад, люмінесцентні типу ЛБ). Оптимальна освітленість робочих місць — 400-500 лк.

У разі ураження електричним струмом необхідно терміново звільнити потерпілого від дії електричного струму (через відключення електроживлення в кімнаті, загального електроживлення на розподільному щиті або іншим способом). Викликати швидку медичну допомогу (подзвонивши за міським телефоном 103). Надати першу медичну допомогу потерпілому, враховуючи наступне:

- якщо потерпілий знепритомнів, але дихає, його необхідно рівно і зручно вкласти, розстебнути одяг, створити приплив свіжого повітря і забезпечити повний спокій;

- при відсутності ознак життя до прибуття лікарів потерпілому необхідно робити штучне дихання.

Фахівці та експерти з генерації документації програмних компонентів у процесі вдосконаленні комп'ютерних систем при виконанні відповідних робіт

несуть відповідальність за порушення вимог з охорони праці і правил техніки безпеки.

При дослідженні та розробці методу і засобу автоматизованої генерації документації програмних компонентів було дотримано усіх вище наведених вимог нормативних документів щодо охорони праці і техніки безпеки при експлуатації комп'ютерної техніки.

4.2. Здоровий спосіб життя людини та його вплив на професійну діяльність

Здоров'я людини ґрунтується на основі генетичних факторів, способу життя та екологічних умов. Однак певною мірою воно залежить також від свідомого ставлення людини до себе та оточуючого середовища [21].

Здоров'я людини – стан повного соціально-біологічного комфорту коли функція всіх органів і систем організму виважені з природним і соціальним середовищем, відсутні будь-які хвилювання, хворобливі стани та фізичні дефекти.

Критерій здоров'я визначається комплексом показників. Однак за найзагальнішими рисами здоров'я індивідуума можна визначити як природний стан організму, що характеризується повною зрівноваженістю будь-яких виражених хворобливих змін [21].

Здоров'я залежить від багатьох факторів, які об'єднуються в одне інтегральне поняття – здоровий спосіб життя. Його метою є навчити людину розумно ставитися до свого здоров'я, фізичної та психічної культури, загартовувати свій організм, вміло організовувати працю і відпочинок.

До основних складових здорового способу життя, згідно [21], належать:

1. Спосіб життя. Має велике значення для здоров'я людини і складається з чотирьох категорій:

- економічної (рівень життя),
- соціологічної (якість життя),

- соціально-психологічної (стиль життя),
- соціально-економічної (устрій життя)

2. Рівень культури. Культура – це самосвідоме ставлення до самого себе. Однак люди дуже часто нехтують своїм здоров'ям, ведуть неправильний спосіб життя, не дотримуються режиму, переїдають, курять. Тому для здоров'я потрібні знання, які увійшли б у повсякденну звичку людини [21].

3. Здоров'я в ієрархії потреб. Не завжди в житті людини здоров'я займає перше місце порівняно з речами та іншими матеріальними благами. У результаті це призводить до шкоди не лише своєму здоров'ю, а й здоров'ю майбутніх поколінь. Отже, здоров'я повинно займати перше місце в ієрархії потреб людини [21].

4. Мотивування. На превеликий жаль, ціну здоров'я більшість людей усвідомлює лише тоді, коли воно значно втрачено. Тільки тоді виникає прагнення вилікувати захворювання, стати здоровим [21].

5. Зворотні зв'язки – нерозумне і довге випробовування стійкості свого організму нездоровим способом життя (алкоголь, нікотин). Тільки через певний час спрацьовують зворотні зв'язки людини, коли вона кидає шкідливі звички, проте це вже доволі часто надто запізно.

6. Навчання здоровому способу життя. Джерелом навичок з цього питання є передусім приклад батьків, допомагає також і медична освіта. Важливим фактором, що визначає реакцію людини на екстремальну ситуацію, є її психофізичні якості та загальний стан. Вони проявляються через чутливість людини до виявлення сигналів небезпеки, перед реакцією на ці сигнали. Показники, які зумовлюють можливості людини виявити небезпечну ситуацію та адекватно реагувати на неї, залежать від індивідуальних особливостей, зокрема від її нервової системи. На поведінку людини у небезпечній ситуації впливає й її психічний та фізичний стан.

7. Психічний стан. Сучасна людина зустрічається з багатьма факторами ризику, що негативно впливають на стан нервової та серцево-судинної систем, знижує опірність організму. При цьому виникає стресова реакція організму.

Так, наприклад, психічна травма, отримана внаслідок конфлікту, виводить людину з нормального психічного стану, що може призвести до суттєвих змін у виконанні професійних функцій і загального функціонального стану. У перекладі «стрес» означає «напруження», тобто відповідь організму на поставлену перед ним проблему.

Стрес – це сукупність загальних неспецифічних біохімічних, фізіологічних і психологічних реакцій організму внаслідок дії надзвичайних подразників різної природи і характеру, які викликають порушення функцій органів.

Повне звільнення від стресу означає смерть, тому слабкий стрес є нормальним явищем у житті і потрібним для реалізації людської повноцінності. Однак якщо він інтенсивний і довготривалий, то може стати основою розвитку захворювань або зумовити смерть.

Медичні та соціологічні дослідження серед різних категорій населення показують, що люди по-різному реагують на надзвичайні ситуації. Є люди, стресостійкі до побутових негараздів, але дуже стресореактивні до сімейних проблем та невдач у коханні, інші боляче сприймають невдачі на роботі, ще інші – втрату соціального статусу.

Відомо, що в осіб до 30 років життєві потреби значно більші, ніж у людей старшого віку, а відтак стресові стани у них переважають.

Велике значення для розвитку стресового стану має поведінка в екстремальних умовах (аварія, кримінальна ситуація, стихійне лихо). Неправильна поведінка у таких ситуаціях найчастіше є причиною шкідливих наслідків стресу. Вона зумовлює результат стресу більше, ніж фактори зовнішнього середовища. У цих випадках стрес може виявитись у вигляді паніки, суєти, істерики.

Стійкість організму до різноманітних стресових станів є дуже індивідуальною. Деякі люди без усіляких наслідків переносять надзвичайно складні екстремальні ситуації, ніколи не непритомніють, не втрачають сили

волі, психологічної рівноваги. Інші вже при незначних екстремальних ситуаціях втрачають витримку і віру в себе.

Для загартованості психічного стану людині треба використовувати фізичну працю, заняття спортом, прогулянки на свіжому повітрі та інші природні фактори.

По-друге, уміння володіти собою, керувати емоціями, психоемоційним напруженням. Це значить постійно контролювати свої дії, вчинки, залишатися врівноваженим навіть у найбільш напружених обставинах

Слід зазначити, що існують різноманітні психологічні засоби зняття нервового напруження для відновлення працездатності, до яких належать: психотерапія, психопрофілактика, психогігієна.

Для більш швидкого відновлення сил після втоми рекомендується використовувати навіюваний сон, тобто навчитися вводити себе на певний час у сон і самостійно виходити з нього бадьорим. Тривалість навіюваного сну 30 – 40 хв.

Найважливішим для людини є її фізичний стан здоров'я, який залежить як від біологічних факторів (спадковості), так і від складного комплексу соціальних, економічних, гігієнічних, кліматогеографічних та інших умов навколишнього середовища.

Під впливом несприятливих факторів рівень фізичного стану здоров'я знижується, а поліпшення умов сприяє його підвищенню.

Виходячи із концепції фізичного здоров'я, основним його критерієм слід вважати енергопотенціал біосистеми, оскільки життєдіяльність будь-якого живого організму залежить від акумуляції і мобілізації енергії для забезпечення фізичних функцій.

Здоров'я людини, опірність її організму до несприятливих умов навколишнього середовища, працездатність значною мірою залежать від харчування. Правильне і раціональне харчування є важливим фактором забезпечення життєдіяльності людини, росту та розвитку організму,

запобігання та лікування хвороб, у тому числі й тих, які сталися внаслідок надзвичайних ситуацій.

Важливим фактором фізичного здоров'я є загартування організму. В основі загартування лежить властивість організму людини пристосовуватись до зміни умов навколишнього середовища. У людини відбувається процес пристосування організму до нових умов існування – виникає адаптація.

Отже, здоровий спосіб життя людини безпосередньо впливає на професійну діяльність. Сучасна професійна діяльність негативно позначається на здоров'ї і якості життя людини, однак з дотримання визначених правил і піклування про своє здоров'я, працівники можуть зменшити негативний вплив стресів та психологічного тиску, фізичного перенавантаження, а також підвищити продуктивність праці.

ВИСНОВКИ

Основні наукові та практичні результати роботи полягають у наступному:

1. Проаналізовано наукові публікації і практичні рекомендації щодо автоматизації процесу генерування документації при вдосконаленні комп'ютерних систем з використанням методів прямої і зворотної інженерії, що дало змогу підтвердити їхню важливість та актуальність імплементації на стадіях життєвого циклу, а також встановити структуру і вміст документів на кожному з них.

2. Досліджено існуючі технології автоматизованого формування структури документів та залежностей між ними, що дало змогу класифікувати їх за ознаками атрибутів, які є спільними для усіх технологій, спільними для конкретного типу документації і відповідно конкретного документу конкретної технології при вдосконаленні комп'ютерних систем.

3. Проведено аналіз існуючих методів генерації документів, що орієнтовані на використання структурного або інкрементного підходів та веб-технологій, у результаті якого підтверджено той факт, що їх застосування у процесі зворотної інженерії не забезпечує можливості одержання документації, яка б задовольняла встановленим вимогам стандартів.

4. Формалізовано та обгрунтовано моделі для представлення документації програмного забезпечення у процесі вдосконалення комп'ютерних систем, що дало змогу побудувати ланцюжок перетворення однієї моделі документу в іншу на трьох рівнях абстракції і врахувати властивості використовуваних технологій реалізації компонентів комп'ютерних систем.

5. Створено метод автоматизованої генерації документів, який враховує вимоги технологій вдосконалення комп'ютерних систем і дає змогу формувати набір документації для майбутнього оновленого програмного

забезпечення та набори його представлення, що керуються параметрами конкретної технології й етапом початку проведення відновлення документації.

6. Побудовано та імплементовано процедуру автоматизованої генерації документації з використанням підходу Model Driven Development, що передбачає виконання впорядкованих трансформацій над моделями документу і, як наслідок, формування документації оновленого ПЗ у вигляді екземплярів метаописів.

7. Розроблено алгоритм функціонування засобу автоматизованої генерації документації при вдосконаленні комп'ютерних систем на основі послідовного виконання трьох процесів: формування опису документів використовуваних технологій при вдосконаленні системи, створення представлень майбутнього програмного забезпечення та процесу генерації документів з врахуванням конкретної технології і стадії початку відновлення документації.

8. Побудовано архітектуру інструментального засобу генерації документів при вдосконаленні та модернізації комп'ютерних систем, що дало змогу забезпечити автоматизацію процесів, характерних для запропонованого методу .

9. Реалізовано засобами мови програмування Java і технології XML структурні елементи архітектури засобу автоматизованої генерації документації, що дало змогу забезпечити ефективність підготовки та управління даними і на основі шаблону «каналів і фільтрів» автоматично генерувати необхідні документи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Авраменко О.А. Архітектура засобів редокументування наслідуваного програмного забезпечення. Вісник НАУ: Науковий журнал. 2007. №3-4 (33). С. 58-62.
2. Андон Ф. Методы инженерии распределенных компьютерных систем. Киев, Изд. «Наукова думка», 1997 г. 228 с.
3. Аптекарь М.Д. История инженерной деятельности: учебное пособие. К.: Аристей. 2013. 568 с.
4. Бабенко Л.П. Основи програмної інженерії. К.: Т-во «Знання» . 2011 269 с.
5. Introduction to XML. URL: https://www.w3schools.com/xml/xml_what.asp (дата звернення 15.11.2022 р.)
6. Грэхем И. Объектно-ориентированные методы. Принципы и практика. Издательский дом «Вильямс», 2004. 880 с.
7. Глушков В.М, Цейтлин Г.Е., Ющенко Е.Л..Алгебра. Языки. Программирование К.: Наук, думка, 1989. 376 с.
8. Інформаційні технології. Процеси життєвого циклу програмного забезпечення (ISO/IEC 12207:2008): ДСТУ 3918:1999. К.: Держспоживстандарт України, 2000. 49 с.
9. Задорожна Н.Т. Менеджмент документообігу в інформаційних системах освіти (для ВНЗ і ППО). К.: 2007. 228 с.
10. Крамар Ю. М. Автоматизация процесса контроля применения стиля языка программирования. Проблемы программирования. Материалы четвертой международной научно-практической конференции по программированию «УкрПРОГ2004». Спец. вып. 2004. № 2-3. с. 208.
11. Маторин С. Анализ и моделирование бизнес-систем: системологическая объектно-ориентированная технология. Харьков: ХНУРЭ. 2002. 322 с.

12. Программное обеспечение IBM Rational. Методология и инструментальные средства разработки программных систем. IBM corp., 2006. 96 с.
13. Сидоров Н.А. Восстановление, повторное использование и переработка программного обеспечения. УСиМ. 1998. №3.С. 74-83.
14. Сидоров Н.А. Реинженерия проектов программного обеспечения / Н.А. Сидоров, В.А. Хоменко, Е.А. Авраменко // Проблемы программирования; НАН України. 2006. № 2,3. С. 31-38.
15. Albin S. T. The Art of Software Architecture: Design Methods and Techniques / S. T. Albin. - John Wiley & Sons. 2003. 312 p.
16. Луцків А.М., Тимощук М.В. Важливість документування в процесі вдосконалення комп'ютерних систем. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 71.
17. Луцків А.М., Тимощук М.В. Моделі представлення документації програмного забезпечення при вдосконаленні комп'ютерних систем. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 72.
18. Apple Publications Style Guide - Apple Computer, Inc. January, 2006. 1996 p.
19. ARISSAP URL: <http://www.sap.com/plattform/netweaver/components/aris/index.erx> (дата звернення 18.11.2022 р.).
20. Жидецький В.Ц. Охорона праці користувачів комп'ютерів. Львів: Афіша, 2000. 176с.
21. Желібо Є. Безпека життєдіяльності. К.: 2001. 483 с.

Додаток А
Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Національна академія наук України
Університет імені П'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Шауляйська державна колегія (Литва)
Жешувський політехнічний університет ім. Лукасевича (Польща)
Білоруський національний технічний університет (Республіка Білорусь)
Міжнародний університет цивільної авіації (Марокко)
Національний університет біоресурсів і природокористування України (Україна)
Наукове товариство ім. Шевченка
ГО «Асоціація випускників Тернопільського національного технічного
університету імені Івана Пулюя»

**АКТУАЛЬНІ ЗАДАЧІ
СУЧАСНИХ ТЕХНОЛОГІЙ**

Збірник

тез доповідей

Том II

**VII Міжнародної науково-технічної
конференції молодих учених та студентів**

28-29 листопада 2018 року



**УКРАЇНА
ТЕРНОПІЛЬ – 2018**

| | |
|---|----|
| А. Луцків, І. Барна АНАЛІЗ СЕРВІСНО-ОРІЄНТОВАНОЇ АРХІТЕКТУРИ В ПРОЦЕСІ ЗАСТОСУВАННЯ DEVOPS ПРАКТИК | |
| A. Lutskiv, I. Barna ANALYSIS OF SERVICE-ORIENTED ARCHITECTURE IN THE PROCESS OF APPLICATION OF DEVOPS PRACTICES | 66 |
| А. Луцків, М. Бондаренко ОСОБЛИВОСТІ ОПТИМІЗАЦІЇ СИСТЕМ ПІДТРИМКИ КОРИСТУВАЧІВ ІЗ ЗАСТОСУВАННЯМ ПІДХОДУ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ | |
| A. Lutskiv, M. Bondarenko FEATURES OF USER SUPPORT SYSTEMS OPTIMIZATION USING THE APPROACH OF MASS SERVICE SYSTEMS | 67 |
| А. Луцків, М. Бондаренко АРХІТЕКТУРА СИСТЕМИ ПІДТРИМКИ З ОПТИМІЗАЦІЄЮ ПРОЦЕСУ УПРАВЛІННЯ ЗВЕРНЕННЯМИ КОРИСТУВАЧІВ | |
| A. Lutskiv, M. Bondarenko SUPPORT SYSTEM ARCHITECTURE WITH OPTIMIZATION OF THE USER COMPLAINT MANAGEMENT PROCESS | 68 |
| В. Яцишин, Т. Кобець ТЕХНОЛОГІЯ MESH В КОМП'ЮТЕРНИХ СИСТЕМАХ ПЕРЕДАЧІ ДАНИХ | |
| V. Yatsyshyn, T. Kobets TECHNOLOGIES OF NON-INVASIVE GLUCOSE LEVEL MEASUREMENT IN BLOOD | 69 |
| В. Яцишин, Т. Кобець МЕТОДИ ВИБОРУ ОПТИМАЛЬНИХ КОМПОНЕНТІВ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ АНАЛІЗУ ІЄРАРХІЇ | |
| V. Yatsyshyn, T. Kobets METHODS OF SELECTING OPTIMUM COMPUTER SYSTEM COMPONENTS BASED ON HIERARCHY ANALYSIS | 70 |
| А. Луцків, М. Тимошук ВАЖЛИВІСТЬ ДОКУМЕНТУВАННЯ В ПРОЦЕСІ ВДОСКОНАЛЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ | |
| A. Lutskiv, M. Tymoshchuk THE IMPORTANCE OF THE DOCUMENTATION IN THE PROCESS OF IMPROVING COMPUTER SYSTEMS | 71 |
| А. Луцків, М. Тимошук МОДЕЛІ ПРЕДСТАВЛЕННЯ ДОКУМЕНТАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИ ВДОСКОНАЛЕННІ КОМП'ЮТЕРНИХ СИСТЕМ | |
| A. Lutskiv, M. Tymoshchuk MODELS OF SOFTWARE DOCUMENTATION VIEW IN THE IMPROVEMENT OF COMPUTER SYSTEMS | 72 |
| І. Головатий ОБРОБКА ЗОБРАЖЕНЬ ІЗ ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ | |
| I. Holovatiy IMAGE PROCESSING USING GENETIC ALGORITHM | 73 |
| Ю. Гук, А. Паламар МЕТОД АДАПТИВНОГО РЕГУЛЮВАННЯ ДОРОЖНЬОГО РУХУ НА ПЕРЕХРЕСТІ НА ОСНОВІ ІНТЕРНЕТУ РЕЧЕЙ | |
| Y. Huk, A. Palamar METHOD OF ADAPTIVE TRAFFIC CONTROL AT AN INTERSECTION BASED ON INTERNET OF THINGS | 74 |

УДК 004.01

А. Луцків, М. Тимошук

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ВАЖЛИВІСТЬ ДОКУМЕНТУВАННЯ В ПРОЦЕСІ ВДОСКОНАЛЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

UDC 004.01

A. Lutskev, M. Tymoshchuk

THE IMPORTANCE OF THE DOCUMENTATION IN THE PROCESS OF IMPROVING COMPUTER SYSTEMS

Документування – це процес фіксування інформації, створюваної в рамках життєвого циклу комп'ютерної системи. Фіксація інформації виконується шляхом її документування. Процес генерування документації, зазвичай, відбувається разом з іншими процесами реалізації КС (рис. 1.1).

Процес формування документації включає набір операцій, які мають відношення до планування, проектування, реалізації, випуску, редагування, поширення і супроводу документів необхідних для керівників, інженерів і користувачів. Потрібно відмітити, що тільки програмне забезпечення КС, яке містить документацію, є програмним продуктом. Тому документування – один з ключових процесів ЖЦ, а повна і якісна документація знижує витрати на його розробку і супровід. Навпаки, неповна і застаріла документація ускладнює розробку і супровід комп'ютерної системи, будучи причиною додаткових матеріальних і часових витрат і, врешті-решт стає однією з причин проведення зворотної інженерії.

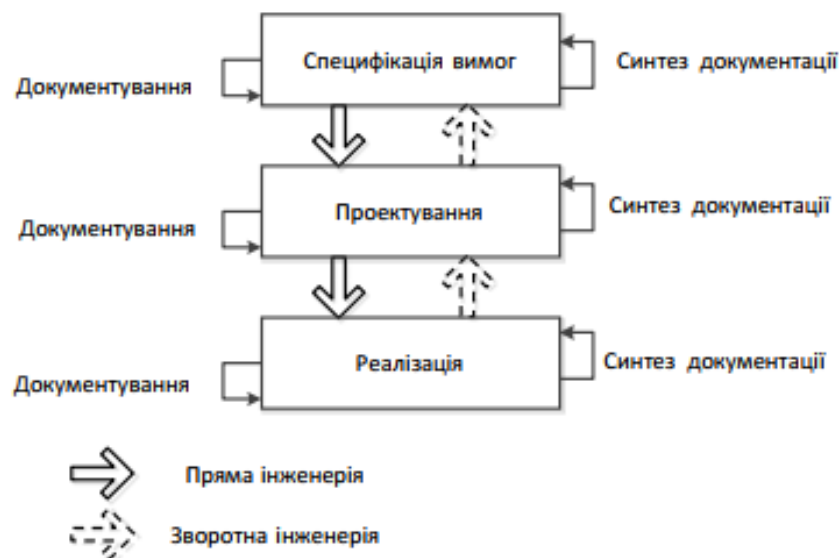


Рисунок 1. Процеси документування в процесі прямої і зворотної інженерії

В контексті зворотної інженерії застосовується синтез документації КС (рис. 1.1). Синтез документації в процесі зворотної інженерії практикується при вдосконаленні КС або при реінженерії наслідуваного ПЗ КС, в тому випадку коли документація програмного продукту загублена, неповна, неточна або погано структурована.

УДК 004.01

А. Луцків, М. Тимошук

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

МОДЕЛІ ПРЕДСТАВЛЕННЯ ДОКУМЕНТАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИ ВДОСКОНАЛЕННІ КОМП'ЮТЕРНИХ СИСТЕМ

UDC 004.01

A. Lutskiv, M. Tymoshchuk

MODELS OF SOFTWARE DOCUMENTATION VIEW IN THE IMPROVEMENT OF COMPUTER SYSTEMS

Аналіз документації програмного забезпечення з точки зору: структури та загальної організації документації; вимог, які висувуються до технологій реалізації комп'ютерної системи; вимог побудови конкретного документу на різних етапах життєвого циклу, показав, що для представлення документів можна використати наступні моделі (рис. 1):

- структурна модель документу;
- метамодель документу, що описує використовувані технології розробки;
- модель документації технології при розробці КС.

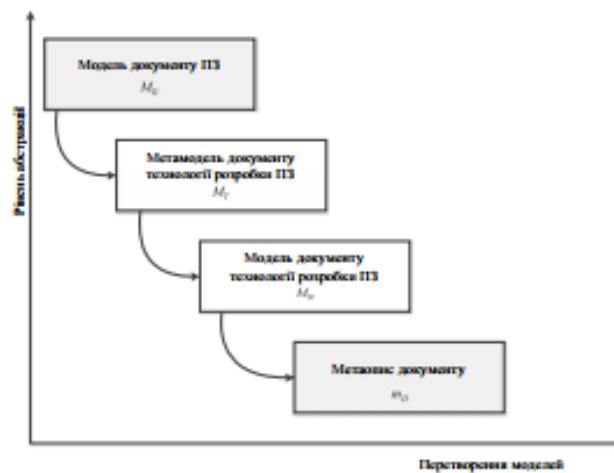


Рисунок 1. Моделі документу з точки зору синтезу документації

Модель документу ПЗ M_G призначена для опису структурних елементів документу ПЗ та враховує особливості автоматизованого створення документації та не залежить від вимог до використовуваних технологій розробки. Метамодель документації M_T щодо використовуваних технологій реалізації описує структурні компоненти документу ПЗ і враховує вимоги R_i , що володіють спільними рисами технологій розробки. Модель M_T є метамоделлю M_G . Модель M_D інтерпретує документ технологій розробки, що забезпечує опис структурних елементів документу з врахуванням вимог R_i для конкретної технології. Ця модель застосовується при створенні описів конкретних документів технології як метаданих, що можуть використовуватись при формуванні екземплярів документів при синтезі документації автоматизованих шляхом.