

**МАГІСТЕРСЬКА РОБОТА**

**МР. ШМ - 50.00.00.000 ПЗ**

**Група ШМ-24-3**

**Смаль Юрій**

**2025**

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

**Смаль Юрій Олександрович**

(прізвище, ім'я, по батькові)

УДК 004.9  
(індекс)

## **МАГІСТЕРСЬКА РОБОТА**

**Засоби та інструменти моделювання цифрових бібліотек**

(назва роботи)

**Інженерія програмного забезпечення**

(назва освітньої програми)

**121 - Інженерія програмного забезпечення**

(шифр і назва спеціальності)

**Смаль Ю.О**

(підпис, ініціали та прізвище здобувача освітнього ступеня)

**Науковий керівник Корнута Володимир Андрійович, к.т.н., доцент**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

**Допущено до захисту**

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

**Нормоконтроль**

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2025

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІІЗ

доц.

В.В. Бандура

“ 04 ” вересня 2025 р.

# ЗАВДАННЯ

## НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

**Смалю Юрію Олександровичу**

(прізвище, ім'я, по-батькові)

**1. Тема магістерської роботи** “Засоби та інструменти моделювання цифрових бібліотек”

керівник проекту (роботи) Корнута В.А., к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 05 ” листопада 2025 р. № 695/7

**2. Строк подання студентом проекту (роботи)** 15 грудня 2025 р.

**3. Вихідні дані до проекту (роботи)** Концепції та моделі побудови та моделювання цифрових бібліотек

**4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)**

1. Аналіз предметної області моделювання та побудови цифрових бібліотек

2. Методи та алгоритми інструментів моделювання цифрових бібліотек

3. Представлення концепції інструменту моделювання цифрових бібліотек на основі домену

4. Імплементация метамоделей для інструментів моделювання цифрових бібліотек

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

1. Приклад реалізації мінімальної цифрової бібліотеки у фреймворку 5S (рис. 1.1)

2. Архітектура MARIAN (рис. 1.2)

3. Архітектура побудови системи цифрової бібліотеки на основі MARIAN (рис. 1.3)

4. Частина ієрархії класів, що використовується в системах MARIAN (рис. 1.4)

5. Приклад мережевої архітектури Open Digital Libraries (рис. 1.5)

## 6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2025 р.

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2025	виконано
2	Аналіз предметної області моделювання та побудови цифрових бібліотек	01.10.2025	виконано
3	Методи та алгоритми інструментів моделювання цифрових бібліотек	17.10.2025	виконано
4	Представлення концепції інструменту моделювання цифрових бібліотек на основі домену	02.11.2025	виконано
5	Імплементация метамodelей для інструментів моделювання цифрових бібліотек	19.11.2025	виконано
6	Специфікація метамodelі та її роль у моделюванні цифрових бібліотек	02.12.2025	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2025	виконано

Студент – магістр \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

## АНОТАЦІЯ

**Магістерська робота:** 81 с., 39 рис., 1 табл., 43 джерела.

**Тема:** Засоби та інструменти моделювання цифрових бібліотек

**Метою роботи** є розроблення та обґрунтування метамодельного підходу до моделювання цифрових бібліотек на основі формальної концепції 5S, а також дослідження інструментальних засобів для підтримки цифрових бібліотек.

**Об'єктом дослідження** є процес моделювання цифрових бібліотек як складних інформаційних систем.

**Предметом дослідження** є методи, моделі, засоби та метамоделі, що застосовуються для формального опису, побудови й оптимізації цифрових бібліотек.

### **Результати дослідження**

В роботі запропоновано архітектурний підхід до створення інструменту моделювання, який ґрунтується на метамодельних принципах і підтримує адаптивність та масштабованість цифрових бібліотек.

### **Висновок**

Розроблено модель керування обмеженнями, яка забезпечує логічну цілісність під час побудови цифрової бібліотеки на рівні моделювання. Побудовано специфікацію метамоделі, яка розширює класичну 5S-архітектуру та підтримує доменні сутності, що є критично важливими для опису цифрових бібліотек

**ЦИФРОВА БІБЛІОТЕКА, МОДЕЛЮВАННЯ ЦИФРОВИХ БІБЛІОТЕК, МЕТАМОДЕЛЬ, ІНФОРМАЦІЙНІ СИСТЕМИ, СТРУКТУРИ ДАНИХ, ІНСТРУМЕНТ МОДЕЛЮВАННЯ, НОРМАЛЬНІ МОДЕЛІ**

## ABSTRACT

**Master Thesis:** 81 pp., 39 fig., 1 tab., 43 sources.

**Topic:** Tools and instruments for modeling digital libraries

**The method of the work** is the development and justification of a metamodel approach to modeling digital libraries based on the formal concept of 5S, as well as the study of tools for supporting digital libraries.

**The object of the research** is the process of modeling digital libraries as complex information systems.

**The subject of the research** is the methods, models, tools and metamodels used for the formal description, construction and optimization of digital libraries.

### **Research results**

In the work, we propose an architectural approach to creating a modeling tool that is based on metamodel principles and supports the adaptability and scalability of digital libraries.

### **Conclusion**

A model for managing constraints has been developed that ensures logical integrity when building a digital library at the modeling level. A metamodel specification has been built that extends the classic 5S architecture and supports domain entities, which is critically important for describing digital libraries.

**DIGITAL LIBRARY, DIGITAL LIBRARY MODELING,  
METAMODEL, INFORMATION SYSTEMS, DATA STRUCTURES,  
MODELING TOOL, FORMAL MODELS**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	9
ВСТУП.....	10
<b>РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ МОДЕЛЮВАННЯ ТА ПОБУДОВИ ЦИФРОВИХ БІБЛІОТЕК .....</b>	<b>14</b>
1.1. Концептуальні та інструментальні підходи до створення цифрових бібліотек .....	14
1.1.1. Комплексність цифрової бібліотеки як інформаційної системи .....	14
1.1.2. Процес генерації цифрової бібліотеки на основі 5SL.....	17
1.2. Еволюція концепції цифрових бібліотек .....	18
1.3. Концептуальне визначення та архітектурні моделі цифрових бібліотек .....	19
1.4. Підходи до моделювання цифрових бібліотек .....	25
Висновки до розділу .....	29
<b>РОЗДІЛ 2. МЕТОДИ ТА АЛГОРИТМИ ІНСТРУМЕНТІВ МОДЕЛЮВАННЯ ЦИФРОВИХ БІБЛІОТЕК .....</b>	<b>31</b>
2.1. Огляд комплексних інструментів для створення цифрових бібліотек .	31
2.2. Оптимізація процесу генерації цифрових бібліотек на основі моделі 5S .....	35
2.2.1. Початковий процес генерації бібліотеки.....	35
2.2.2. Мінімізація складності та роль метамоделі.....	37
2.3. Представлення концепції та архітектури інструменту моделювання цифрових бібліотек на основі домену .....	38
2.4. Огляд формальної моделі 5S/5SL.....	40
2.4.1. Модель потоків .....	41
2.4.2. Структурна модель .....	42
2.4.3. Просторова модель .....	44

2.4.4. Модель сценаріїв .....	44
2.4.5. Модель суспільств .....	44
Висновки до розділу .....	45
РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ МЕТАМОДЕЛЕЙ ДЛЯ ІНСТРУМЕНТІВ МОДЕЛЮВАННЯ ЦИФРОВИХ БІБЛІОТЕК .....	47
3.1. Специфікація метамоделі та її відображення в 5S* .....	47
3.1.1 Вимоги до метамоделі .....	47
3.1.2. Огляд прикладу метамоделі .....	48
3.2. Специфікація елементів метамоделі .....	49
3.2.1. Елемент DataSet .....	49
3.2.2. Елемент DataType .....	50
3.3. Опис сценаріїв моделювання .....	52
3.4. Опис властивостей елемента DataType у метамоделі 5S* .....	54
3.5. Процес керування обмеженнями при моделюванні цифрової бібліотеки .....	56
3.6. Архітектурні рішення та дизайн інтерфейсу для моделювання .....	61
3.7. Специфікація метамоделі та її роль у моделюванні цифрових бібліотек .....	66
Висновки до розділу .....	68
ВИСНОВКИ .....	70
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	73
ДОДАТКИ .....	78

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

5S - Streams, Structures, Spaces, Scenarios, Societies (5S Model)

5SL - 5S Language (XML-імплементация для 5S Model)

ЦБ – Цифрова бібліотека

DL - Digital Library

MIME - Multipurpose Internet Mail Extensions

ETD - Electronic Thesis and Dissertation

UIML - User Interface Markup Language

MOF - Meta Object Facility

OMG - Object Management Group

NDLTD - Networked Digital Library of Theses and Dissertations

CITIDEL - Computing Indexes and Tables for Internet Digital Educational  
Libraries

IMS - Information Management System (Metadata Standard)

MARC - Machine-Readable Cataloging (Metadata Standard)

IEEE LTSC-LOM - IEEE Learning Technology Standards Committee -  
Learning Object Metadata

## ВСТУП

### **Актуальність теми.**

Сучасний розвиток цифрових технологій зумовлює докорінні трансформації у сфері зберігання, організації та поширення інформаційних ресурсів. У цьому контексті цифрові бібліотеки постають як інтелектуальні інформаційні системи, що забезпечують не лише доступ до електронних даних, але й підтримку процесів пошуку, класифікації, структуризації та взаємодії користувачів із ресурсами. Зростання обсягу електронних документів, різноманіття форматів і необхідність інтеграції контенту з різних джерел актуалізують потребу в ефективних методах моделювання та формалізації цифрових бібліотечних систем. Водночас відсутність стандартизованих підходів ускладнює побудову масштабованих, гнучких і багаторазово використовуваних моделей цифрових бібліотек.

Моделювання цифрових бібліотек є складним багаторівневим процесом, оскільки охоплює архітектуру даних, інформаційні потоки, користувацькі сценарії, механізми управління метаданими та організацію суспільної взаємодії. Наявні інструменти часто не забезпечують належної формальної узгодженості між структурними та функціональними елементами системи, що призводить до підвищення складності та неконсистентності моделей. Саме тому актуальною є розробка підходів, які дозволяють формально описувати цифрові бібліотеки на різних рівнях абстракції та забезпечувати уніфікацію процесів проєктування.

У роботі обґрунтовано застосування моделі 5S/5SL як комплексної основи для опису цифрових бібліотек через системну взаємодію структур, потоків, сценаріїв, просторів і суспільств. На основі цієї моделі запропоновано метамодельний підхід до моделювання цифрових бібліотек, що дозволяє формалізувати ключові доменні сутності, типи даних та правила їх взаємодії. Імплементация метамоделі дає змогу автоматизувати процес генерації цифрових бібліотек, підвищити гнучкість інструментальних рішень

та забезпечити можливість повторного використання компонентів. Таким чином, дослідження спрямоване на створення теоретичної та методологічної основи для побудови сучасних інструментів моделювання цифрових бібліотек, здатних відповідати вимогам масштабованості, адаптивності та структурної узгодженості.

Актуальність теми зумовлена стрімким зростанням обсягу цифрових ресурсів та необхідністю розроблення інтелектуальних інструментів для їх ефективного представлення й управління. Цифрові бібліотеки стають ключовим елементом інформаційної інфраструктури в науковій, освітній, культурній та адміністративній сферах, забезпечуючи збереження, доступність і структуроване використання різноманітних даних. Проте багатокomпонентність цифрових бібліотек потребує впровадження формальних моделей, здатних охопити складні інформаційні зв'язки та забезпечити узгодженість між різними рівнями їхньої архітектури.

У науковому середовищі недостатньо представлено комплексні інструменти моделювання цифрових бібліотек, які враховують доменні особливості, формальні обмеження та структурні залежності між інформаційними сутностями. Сучасні рішення або надмірно спеціалізовані, або недостатньо формалізовані, що створює труднощі у побудові масштабованих цифрових платформ. Водночас методологія 5S та її розширення, які пропонують формальне представлення потоків, структур, просторів і сценаріїв, дають змогу систематизувати процеси проектування цифрових бібліотек. Розробка метамоделі для моделювання цифрових бібліотек забезпечує можливість створення багаторівневих, взаємопов'язаних моделей, які зменшують складність і підвищують якість реалізації системи.

Таким чином, актуальність дослідження визначається потребою у науково обґрунтованих інструментах і методах моделювання цифрових бібліотек, здатних забезпечити системність, узгодженість і адаптивність сучасних інформаційних платформ.

**Метою роботи** є розроблення та обґрунтування метамодельного підходу до моделювання цифрових бібліотек на основі формальної концепції 5S, а також дослідження інструментальних засобів для підтримки цифрових бібліотек.

**Об'єктом дослідження** є процес моделювання цифрових бібліотек як складних інформаційних систем.

**Предметом дослідження** є методи, моделі, засоби та метамоделі, що застосовуються для формального опису, побудови й оптимізації цифрових бібліотек.

**Завдання дослідження:**

- дослідити предметну область цифрових бібліотек та проаналізувати сучасні концептуальні й інструментальні підходи до їх створення;
- проаналізувати модель 5S/5SL та визначити можливості її застосування для комплексного моделювання цифрових бібліотек;
- дослідити існуючі інструменти моделювання та визначити їхні обмеження;
- розробити концепцію метамоделі, що формалізує ключові структурні та функціональні елементи цифрової бібліотеки;
- побудувати специфікацію елементів метамоделі та визначити між ними логічні зв'язки;
- сформулювати архітектурні рішення для інструменту моделювання цифрових бібліотек;
- дослідити можливості автоматизації процесу створення цифрової бібліотеки на основі метамоделі.

**Методи дослідження**

У роботі застосовано сукупність загальнонаукових та спеціальних методів, серед яких аналіз і синтез для вивчення предметної області цифрових бібліотек, порівняльний метод для оцінювання інструментів моделювання та підходів до їх реалізації. Використано методи формалізації й абстрагування для побудови метамоделі цифрової бібліотеки, а також методи

структурного моделювання для опису компонентів 5S та визначення їх взаємозв'язків. Застосовано системний підхід для розроблення архітектури інструменту моделювання та методи дизайну для представлення структур і сценаріїв.

### **Наукова новизна отриманих результатів**

Наукова новизна роботи полягає у розробленні та обґрунтуванні метамоделі цифрової бібліотеки, побудованої на основі формальної концепції 5S, яка забезпечує системну підтримку моделювання інформаційних структур, потоків, просторів та сценаріїв взаємодії. Запропоновано формальний опис ключових елементів метамоделі — DataSet, DataType та інших сутностей — що дозволяє уніфікувати структури даних та забезпечити їх багаторазове застосування.

### **Практичне значення результатів**

Практичне значення результатів полягає у можливості застосування запропонованої метамоделі та інструментальних рішень для розроблення програмних систем моделювання цифрових бібліотек. Результати можуть бути використані для автоматизації процесу побудови цифрової бібліотеки на основі формальних моделей та доменних специфікацій. Розроблені концепції можуть бути інтегровані у системи управління знаннями, цифрові репозитарії, освітні платформи та наукові інформаційні системи.

**Структура магістерської роботи.** Представлена робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 81 сторінку, і містить 39 рисунків, 1 таблицю, перелік використаних джерел із 43 позицій та один додаток.

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ МОДЕЛЮВАННЯ ТА ПОБУДОВИ ЦИФРОВИХ БІБЛІОТЕК

## 1.1. Концептуальні та інструментальні підходи до створення цифрових бібліотек

З інтеграції мережевих технологій, зокрема Інтернету та всесвітньої павутини (WWW), сфера цифрових бібліотек (ЦБ) набула статусу ключової прикладної галузі. На відміну від традиційних бібліотечних систем, ЦБ оперують значними колекціями цифрових об'єктів і функціонують як онлайн-платформи інформаційних послуг. Їхня критична важливість полягає в архівації та забезпеченні доступності записів людських знань в умовах глобалізованого мережевого простору.

Значимість і складність архітектури ЦБ стимулювали інтенсивні дослідницькі зусилля. Серед пріоритетних напрямків досліджень у галузі ЦБ виділяють:

- Класифікацію інформаційних ресурсів.
- Взаємодію між гетерогенними колекціями (інтероперабельність).
- Комунікаційні протоколи та стандарти.
- Пошукові механізми та системи.
- Візуалізацію інформації.
- Юзабіліті та проблеми людино-комп'ютерної взаємодії (Human-Computer Interaction, HCI).

Незважаючи на значний фокус на удосконаленні функціональності ЦБ, недостатня увага приділяється спрощенню процесу їх створення.

### *1.1.1. Комплексність цифрової бібліотеки як інформаційної системи*

Цифрова бібліотека являє собою складну інформаційну систему, що інтегрує низку дисциплін комп'ютерних наук, зокрема інформаційний пошук,

бази даних та гіпертекстові технології. Процес розробки ЦБ вимагає вирішення низки фундаментальних питань:

- Специфікація зберіганого контенту.
- Методи організації, структурування, опису та доступу до контенту.
- Діапазон надаваних послуг (наприклад, пошук, перегляд, персоналізація, кооперація).
- Моделі використання послуг та взаємодії користувачів у середовищі ЦБ [2].

Оскільки універсальні та ідеальні рішення для цих питань наразі відсутні, існуюча складність та висока ресурсомісткість створення нових систем залишаються значними бар'єрами.

Існує значний попит на створення нових ЦБ, про що свідчить наявність сотень функціонуючих систем та поточних проєктів. Різноманітні спільноти користувачів висувають унікальні вимоги до функціональності ЦБ. Проте, багато існуючих ЦБ характеризуються монолітністю, щільною інтеграцією, низькою гнучкістю та обмеженими можливостями взаємодії (інтероперабельності).

Створення ЦБ, що відповідає специфічним потребам, вимагає значних часових та трудових витрат. Додатковим ускладненням є те, що розробники ЦБ часто не є експертами у галузях програмної інженерії чи інформаційних наук (наприклад, технічний персонал бібліотек, викладачі).

Для спрощення процесу створення ЦБ необхідне попереднє коректне узагальнене моделювання системи. Мультидисциплінарний характер ЦБ ускладнює це узагальнене моделювання, породжуючи різноманіття концептуальних підходів та, як наслідок, численні визначення ЦБ [3].

На підставі труднощів моделювання ЦБ, в [4] запропонували теорію 5S. Модель 5S охоплює п'ять ключових аспектів:

1. Потоки (Streams),
2. Структури (Structures),
3. Простори (Spaces),

4. Сценарії (Scenarios),
5. Суспільство (Societies).

Модель потоків визначає комунікаційний вміст між ЦБ та користувачами. Модель структур описує організацію інформації для зручного представлення. Модель просторів визначає способи представлення інформації (доступність та зручність). Модель сценаріїв специфікує доступні інформаційні послуги. Модель суспільств визначає, як ЦБ задовольняє інформаційні потреби користувачів.

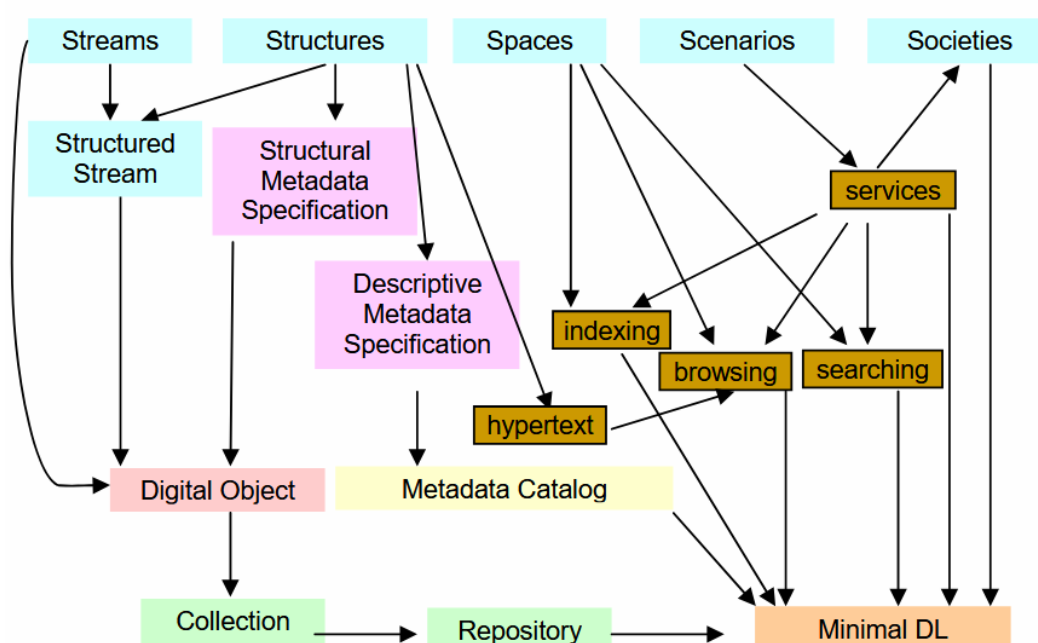


Рис. 1.1. Приклад реалізації мінімальної цифрової бібліотеки у фреймворку 5S

Рисунок 1.1 демонструє концепції метамоделі для мінімальної цифрової бібліотеки (ЦБ), використовуючи фреймворк 5S. Стрілки на рисунку вказують на те, що одні концепції використовуються при визначенні інших концепцій. Наприклад, цифрові об'єкти складаються з потоків (streams) і структур (structures).

Формальність моделі 5S дозволяє чітко описувати складні системи ЦБ, забезпечуючи однозначне визначення їхніх характеристик і поведінки. Ця

формалізація також відкриває шлях до автоматичного перетворення моделей 5S на готові програмні реалізації.

Також було розроблено 5SL (5S Language), мову для декларативного опису та генерації ЦБ [6]. 5SL є XML-реалізацією моделі 5S, являючи собою високорівневу, спеціалізовану мову, що забезпечує деталізовану специфікацію ЦБ.

### *1.1.2. Процес генерації цифрової бібліотеки на основі 5SL*

У разі реалізації генератора ЦБ на базі 5SL, процес напівавтоматичної генерації системи передбачає такі етапи:

1. Розробник ЦБ формує специфікацію 5SL, яка відображає вимоги до конкретної системи.

2. Генератор ЦБ обробляє файл 5SL та автоматично генерує відповідну цифрову бібліотеку.

Такий підхід значно знижує вимоги до кваліфікації розробника (необхідне лише концептуальне розуміння системи та здатність до кодування у 5SL), прискорює процес розробки та підвищує якість створюваних систем.

Незважаючи на переваги, 5SL має низку обмежень:

- Вимоги до знань 5SL. дизайнер повинен мати ґрунтовне розуміння мови 5SL для коректного вираження вимог до системи.

- Складність семантики. Файл 5SL складається з п'яти підмоделей (5S), які, хоча і виражені в XML, використовують різні набори концепцій та семантик. Це забезпечує сумісність та розширюваність (інтеграція існуючих стандартних форматів), але водночас вимагає від розробника розуміння множини різних семантичних специфікацій для опису єдиної системи.

- Труднощі ручного кодування. Для великих і складних ЦБ, ручне написання XML-файлів стає надзвичайно складним навіть для експертів.

- Проблеми оглядовості. Великий набір XML-файлів утруднює отримання загального уявлення про ЦБ, що негативно впливає на процеси підтримки, оновлення та розуміння існуючої системи.

Подолання цих недоліків 5SL є необхідним кроком для повної реалізації потенціалу моделі 5S/5SL.

## **1.2. Еволюція концепції цифрових бібліотек**

Концептуальні основи цифрових бібліотек (ЦБ) було закладено ще в середині ХХ століття. Спочатку було сформульовано ідею Memex [7] — футуристичної особистої інформаційної системи. Memex було описано як "пристрій, у якому людина зберігає всі свої книги, записи та комунікації, і який механізований так, щоб його можна було консультувати з надзвичайною швидкістю та гнучкістю".

Оскільки ця ідея передувала комерційній доступності цифрових технологій зберігання, то було передбачено використання мікрофільмів для зберігання контенту Memex. Хоча телекомунікаційні технології того часу були примітивними, концепція Memex передбачала розширення функціональності пристрою для забезпечення дистанційного доступу.

З появою та комерціалізацією комп'ютерних систем, а також розвитком технологій цифрового зберігання та комунікацій, концепція ЦБ набула комп'ютеризованої форми.

В 1950-1960-ті роки дослідження були сфокусовані на комп'ютеризованій індексації інформації. Проект Intrex (MIT, 1960-ті) була розроблена система зберігання та пошуку інформації. Учасники проекту припускали, що комп'ютеризовані ЦБ матимуть справу з двома основними типами даних: цифровими та фотографічно збереженими (на мікрофільмах) [8].

Дослідники Intrex передбачали майбутнє ЦБ як центральний ресурс інформаційної мережі передачі, інтегрованої в академічну спільноту: бібліотека стане центральним ресурсом інформаційної мережі передачі, яка поширяться на всю академічну спільноту. Користувачі мережі спілкуватимуться один з одним, а також з бібліотекою. Інформаційний

трафік контролюватиметься за допомогою комп'ютерної утиліти університету з розподілом часу".

Розквіт інтернету та всесвітньої павутини (WWW) надав потужний імпульс для розвитку сучасних ЦБ. Швидке зростання WWW (до мільйона нових електронних сторінок щодня) призвело до значного обсягу доступної інформації, яка, не має організації та структури. Цифрові бібліотеки почали виконувати критичну функцію організації інформації в мережевому середовищі та сприяння ефективному інформаційному пошуку для різних спільнот користувачів.

Концепція цифрових бібліотек пройшла кардинальну трансформацію: від Metex (індивідуальна, мікрофільмова, механізована система) до ідеальних цифрових бібліотек DLI —які є людино-орієнтованими системами, здатними обслуговувати користувачів із усіх дисциплін у глобальному мережевому середовищі.

### **1.3. Концептуальне визначення та архітектурні моделі цифрових бібліотек**

З огляду на високу складність цифрових бібліотек як інформаційних систем та відносну новизну галузі досліджень, не існує єдиного універсального визначення цифрової бібліотеки. Різноманіття існуючих визначень відображає різні перспективи та акценти у їхньому розгляді.

#### **1. Інституційно-орієнтоване Визначення (DLF)**

Цифрові бібліотеки — це організації, які надають ресурси, включаючи спеціалізований персонал, для вибору, структурування, забезпечення інтелектуального доступу, інтерпретації, розповсюдження, збереження цілісності та забезпечення збереження з часом колекцій цифрових робіт, щоб вони були легко доступні та економічно вигідні для використання визначеною спільнотою або набором спільнот.

## 2. Колекційно-орієнтоване визначення

З цієї точки зору, ЦБ – це бібліотека, яка підтримує всю або значну частину своєї колекції в комп'ютерно-доступній формі як альтернативу, доповнення або доповнення до традиційних друкованих та мікрофільмових матеріалів... Термін 'колекція' позначає документи, які бібліотека набуває або підтримує.

Було ідентифіковано низку елементів, спільних для більшості визначень, що підкреслюють мережевий та розширений характер ЦБ:

- Цифрова бібліотека не є єдиною сутністю;
- Вона вимагає технологій для зв'язку ресурсів багатьох установ;
- Зв'язки між численними ЦБ та інформаційними службами є прозорими для кінцевих користувачів;
- Універсальний доступ до ресурсів є однією з цілей;

Колекції ЦБ не обмежуються лише сурогатами документів (копіями друкованих видань), а включають цифрові артефакти, які не можуть бути представлені у друкованих форматах.

Наведемо приклади архітектур цифрових бібліотек.

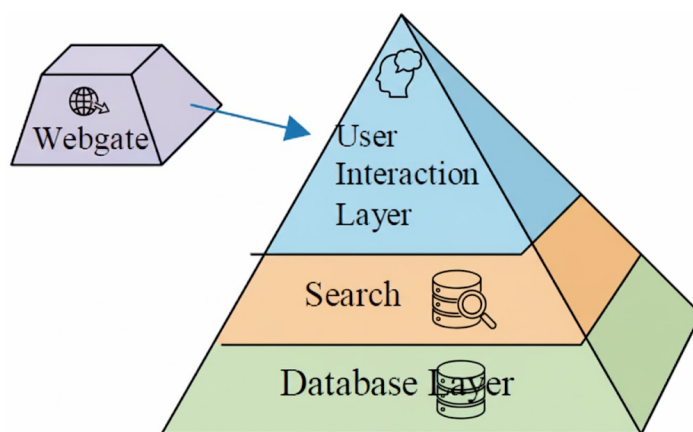


Рис. 1.2. Архітектура MARIAN

MARIAN являє собою монолітну систему індексації, пошуку та отримання, оптимізовану для ЦБ. Її розробка розпочалася в технологічному інституті Вірджинії та продовжилася в лабораторії досліджень цифрових

бібліотек (DLRL). Монолітна архітектура системи MARIAN передбачає тісну інтеграцію всіх функціональних модулів.

Розглянемо архітектуру системи MARIAN, яка функціонує як об'єднаний архів (union archive) для цифрових бібліотек, зокрема, імплементації на базі мережевої цифрової бібліотеки дипломних і дисертаційних робіт (NDLTD).

### 1. Архітектурні компоненти та інтероперабельність

Система побудована навколо програмного забезпечення MARIAN (MARIAN Mediation Middleware) яке надає єдиний інтерфейс запитів до системи, пропонуючи потужний та гнучкий набір функціональних можливостей для пошуку інформації.

Оболонки (Wrappers) використовуються для подолання бар'єрів гетерогенності (відмінностей у форматах та протоколах) між віддаленими колекціями. Мова 5SL (XML-based specification language) використовується для декларативного опису можливостей віддалених колекцій та наповнення різних структур даних всередині оболонок.

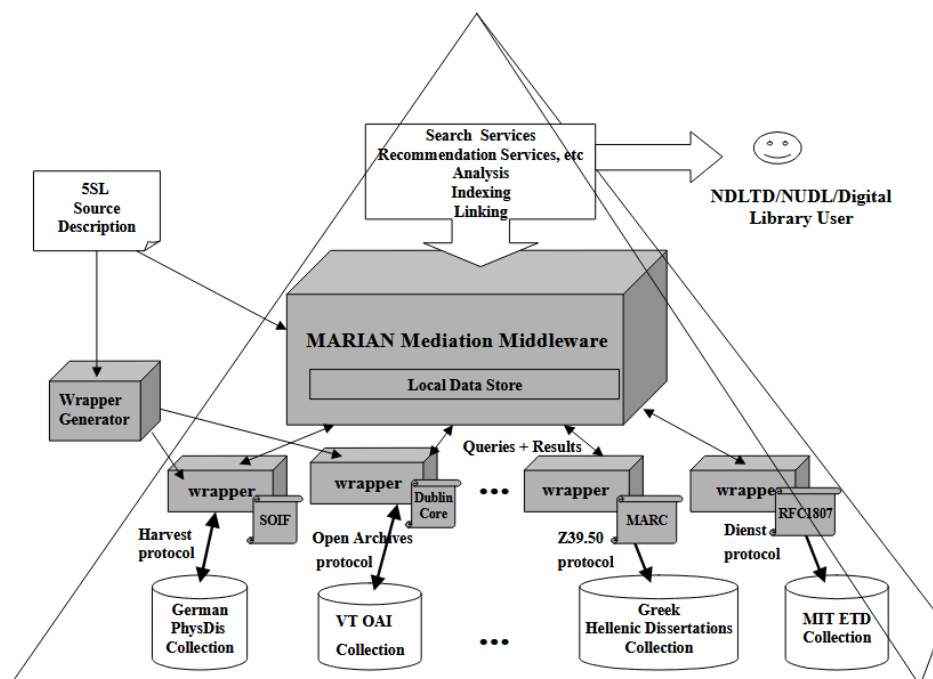


Рис. 1.3. Архітектура подубови системи цифрової бібліотеки на основі MARIAN Mediation Middleware

## 2. Створення об'єднаного архіву (Union Archive)

Імплементація на базі об'єднаного архіву передбачає централізоване зберігання даних, зібраних із різних джерел-членів NDLTD:

- Збір - інформація періодично витягується з різних джерел за допомогою різних підходів збору. Прототипова система використовує пакет Harvest та протоколи, такі як Open Archives [12], Dienst та Z39.50 [13]. Зокрема, протокол Open Archives забезпечує часткове вирішення проблем інтероперабельності метаданих і пропонує простий, але ефективний механізм збору для подолання гетерогенності між членами NDLTD.

- Обробка та завантаження. Зібрані документи обробляються, об'єднуються з даними з інших джерел і завантажуються в централізоване сховище даних – об'єднаний архів (union archive).

- Деталі обробки включають виявлення структури в документах, розпізнавання термінів у тексті та генерацію індексів.

- Зібрані документи проходять фільтрацію (наприклад, за релевантністю, часом оновлення) і використовуються для оновлення об'єднаного архіву.

- Зберігання зібраних даних локально дозволяє виконувати запити до локальних даних без необхідності подальшої взаємодії з оригінальними джерелами.

## 3. Об'єктно-орієнтована модель інформаційних об'єктів

Цифрові бібліотечні об'єкти в MARIAN представлені в об'єктно-орієнтованій моделі, що дозволяє адаптувати методи пошуку для кожного класу об'єктів та забезпечує наслідування між тісно пов'язаними класами.

Для структурованих текстових документів, таких як дипломні та дисертаційні роботи, використовуються стандартні класи MARIAN, наприклад, StructuredDocument, PersonalName та Text. Специфічні типи документів, наприклад, XML-тези або записи метаданих Dublin Core, визначаються як такі, що мають особливу структуру, але успадковують загальні методи пошуку від батьківського класу. Клас Text, наприклад,

визначає метод пошуку, заснований на наближеній відповідності (approximate match) між набором термінів запиту та послідовністю термінів, знайдених у тексті.

Цей клас далі спеціалізується на підкласи для різних мов. У поточному прототипі MARIAN NDLTD підкласи включають English (англійська) та Non-English European (неанглійська європейська), причому останній клас розуміє структуру речень, але обробляє всі слова як неаналізовані рядки. Отримана ієрархія доступних для пошуку DigitalInformationObjects (цифрових інформаційних об'єктів) може бути розширена в будь-який момент.

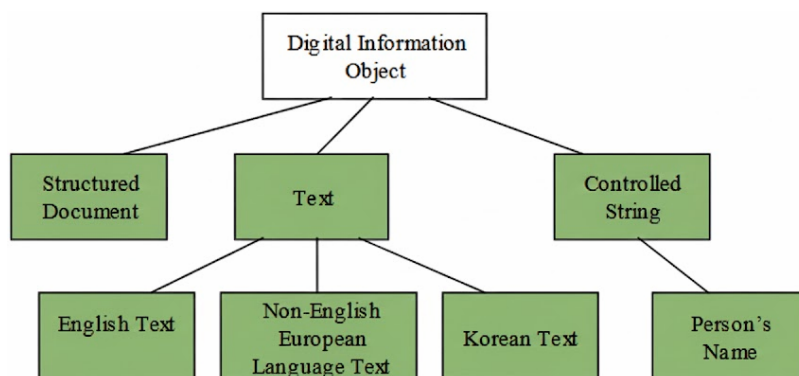


Рис. 1.4. Частина ієрархії класів, що використовується в системах MARIAN

Відкриті цифрові бібліотеки (Open Digital Libraries, ODLs) — це системи, реалізовані як мережі розширених відкритих архівів. ODLs запозичують ідеї простоти та повторного використання з відкритих архівів, додаючи принципи розширюваності та компонентності.

ODLs є компонентними системами. Протоколи комунікації між внутрішніми компонентами ЦБ розроблені як розширення протоколу збору метаданих OAI. Компоненти, що дотримуються цих протоколів, об'єднуються для формування бекенду цифрової бібліотеки.

Типова мережа ODL ілюструє децентралізований та компонентний підхід до організації інформаційних ресурсів, що показано на рис. 1.5.

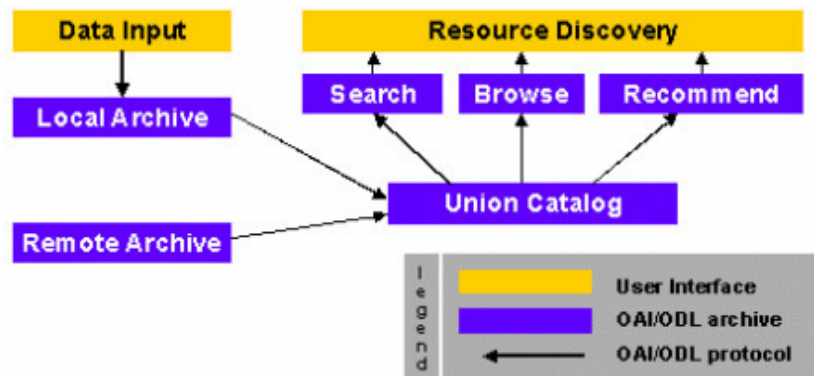


Рис. 1.5. Приклад мережевої архітектури Open Digital Libraries

Система здійснює збір даних із семи вихідних архівів через інтерфейси протоколу збору метаданих відкритих архівів (OAI-PMH). Один із цих архівів вимагав застосування фільтрації через незначні відмінності у реалізації протоколу. Зібрані дані були агреговані в центральний архів для подальшого використання локальними службами.

На основі агрегованих даних були реалізовані три ключові високорівневі служби:

- Search - здійснював індексацію даних та надавав OAI-подібний інтерфейс для формування запитів за ключовими словами.

- Browse - здійснював сортування даних та надавав дещо інший OAI-подібний інтерфейс для доступу до елементів за елементами контрольованої лексики.

- Recent - зберігав інформацію про нещодавно додані елементи та за запитом повертав їх випадкову вибірку.

Користувацькі інтерфейси (UI) були створені з використанням мінімальних скриптів. Функціональність цих скриптів полягала в:

- Трансляції елементів HTML-форм у запити ODL (Open Digital Library).

- Конвертації XML-відповідей у людино-орієнтовані дисплеї за допомогою таблиць стилів XSL (XSL stylesheets).

Користувацький інтерфейс цієї системи ODL NDLTD, що ілюструє операцію перегляду, представлений на рисунку 1.6.



Рис. 1.6. Користувацький інтерфейс системи Open Digital Library NDLTD, що зображує операцію перегляду

Дизайн та імплементація системи ODL NDLTD відповідають сформульованим принципам проектування, що забезпечує розширювану архітектуру для подальшого розвитку.

#### 1.4. Піходи до моделювання цифрових бібліотек

Цифрові бібліотеки (ЦБ) є, як правило, великими та складними інформаційними системами. Для таких комплексних систем критично важливим є наявність формальних моделей та теорій, оскільки вони дозволяють точно та чітко описувати, специфікувати та розуміти ці системи. Більшість зрілих класів інформаційних систем вже мають усталені формальні моделі:

- Системи баз даних - мають реляційну та об'єктно-орієнтовану моделі.
- Інформаційний пошук - оперує трьома класичними моделями: булевою, векторною та імовірнісною.
- Гіпертекстові системи мають власні формальні моделі, зокрема еталонну модель гіпертексту Dexter.

Робота над формальним моделюванням цифрових бібліотек є менш зрілою, ніж для інших інформаційних систем.

### 1. Гібридний системний підхід.

Суть: абстрактна структура ЦБ визначається як поєднання спеціалізованої бази даних та зручного користувацького інтерфейсу. Автор запропонував формальну структуру даних для зв'язування об'єктно-орієнтованої бази даних з гіпермедіа.

Недолік: модель не описує багато інших специфічних характеристик ЦБ, таких як інтегрованість, класифікація, інструменти організації тощо.

### 2. Канонічна модель

Суть: представлена канонічна модель, здатна уніфіковано репрезентувати гетерогенні колекції ЦБ. Досліджувалося відображення гетерогенних даних та об'єктів на канонічну модель для збереження інформації та забезпечення семантичної інтегрованості.

Недолік: Спроба є неповною, оскільки не включає описів інформаційних послуг, класифікації та багатьох інших аспектів ЦБ.

### 3. Модель Greenstone

Програмне забезпечення Greenstone [10] інкорпорує просту та практичну, хоча й неформальну модель ЦБ. Цифрова бібліотека Greenstone містить організовану інформацію, доступну через Інтернет. Бібліотека включає багато різних колекцій, кожна з яких організована по-своєму. Колекції складаються з документів — фундаментальних інформаційних повідомлень в електронній формі (текст, аудіо, відео, зображення тощо).

Кожна колекція має уніфікований інтерфейс. Після автоматичного процесу побудови (building process), який створює структури для кожної колекції, вона стає доступною для обслуговування, пошуку та перегляду (maintainable, searchable, and browsable). Новий матеріал інтегрується шляхом повторної побудови колекції.

Основний недолік: модель Greenstone є неформальною.

### 4. Моделі Кастеллі для цифрових бібліотек

Авторами було запропоновано дві окремі моделі для ЦБ:

- Математична модель

Суть: моделює цифровий контент і складається з двох основних елементів: моделі документа та моделі метаданих.

Модель документа включає чотири аспекти: документи, екземпляри документів, подання (views — репрезентують вміст та структуру документа) та маніфестації. Модель метаданих складається з форматів та специфікацій, здатних інтегрувати якомога більше форматів метаданих.

- Архітектурна модель (OpenDLib).

Специфікує послуги, які надає ЦБ. Вона включає федерацію ЦБ-сервісів, які реалізують функціональність з мінімальними припущеннями щодо документів. Модель поділена на ядро (kernel), яке моделює основні елементи архітектури та їхні екземпляри, та специфікацію типів послуг (service types specification).

Функції OpenDLib: набуття (аквізиція), зберігання та збереження (preservation), пошук, перегляд та отримання, відбір та розповсюдження документів, авторизація та автентифікація користувачів.

## 5. Модель 5S

Модель 5S [11] покликана забезпечити формальну модель, яка повністю описує системи ЦБ.

Функціональне призначення компонентів:

1) модель суспільств (society model): описує, як задовольняти інформаційні потреби різних спільнот користувачів.

2) Модель сценаріїв (scenarios model): описує, як надавати всі види інформаційних послуг користувачам.

3) Модель просторів (space model): вирішує проблему представлення інформації у зручний спосіб.

4) Модель структур (structure model): описує організацію інформації в об'єкті.

5) Модель потоків (stream model): вирішує проблему комунікації інформації користувачам.

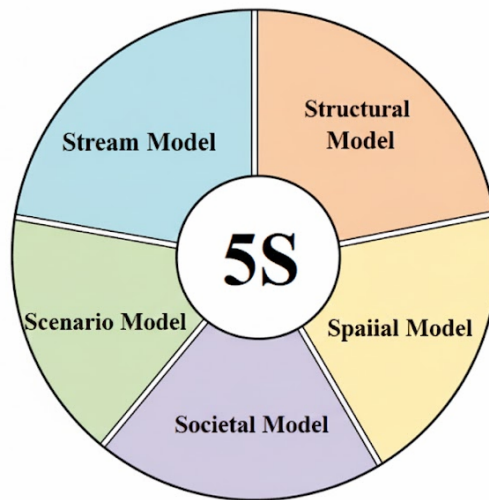


Рис. 1.7. Огляд моделі 5S

Перевагою є те, що 5S представляє ЦБ з дуже високим рівнем деталізації (*fine granularity*) та охоплює майже всі примітиви у ЦБ. Вона є більш повною порівняно з ранніми спробами, більш формальною та високорівневою порівняно з моделлю Greenstone, а також більш інтегрованою порівняно з моделями Каstellі.

5S надає спільну термінологію та фреймворк, який дозволяє різним зацікавленим сторонам (бібліотекарям, розробникам, користувачам, адміністраторам) ефективно спілкуватися та обговорювати різні аспекти цифрової бібліотеки. Систематичний аналіз кожного "S" допомагає виявити вимоги, які могли б бути пропущені при менш структурованому підході. Наприклад, аналіз "Societies" може виявити важливі юридичні або етичні питання, а "Scenarios" — несподівані сценарії використання.

5S є концептуальною моделлю, а не жорстким стандартом. Вона може бути адаптована до конкретних потреб та контексту кожної цифрової бібліотеки, дозволяючи вибирати відповідні технології та методології. Цифрові бібліотеки є складними системами. 5S допомагає розбити цю складність на керовані частини, що робить процес проектування менш обтяжливим та більш ефективним.

Включення "Scenarios" та "Societies" забезпечує сильний фокус на потребах та поведінці користувачів, що є критично важливим для створення

успішної та зручної цифрової бібліотеки. Чітке розуміння кожного аспекту 5S допомагає приймати обґрунтовані рішення щодо архітектури, технологій, політик та стратегій розвитку цифрової бібліотеки.

Наприклад, розробляючи цифрову бібліотеку, можна спочатку визначити Streams (які типи документів, медіафайлів будуть зберігатися), потім Structures (як вони будуть організовані, які метадані використовуватимуться), потім Spaces (як виглядатиме інтерфейс, як буде організований доступ), потім Scenarios (як користувачі шукатимуть, переглядатимуть, взаємодіятимуть з контентом) і, нарешті, Societies (хто є цільовою аудиторією, які є політики використання та авторського права).

### **Висновки до розділу**

У першому розділі було з'ясовано, що цифрові бібліотеки формуються як складні інформаційні системи, у яких поєднуються технологічні, структурні та соціальні компоненти. Розгляд концептуальних підходів дав змогу визначити, що сучасні цифрові бібліотеки виходять далеко за межі простих електронних архівів і перетворюються на інтерактивні інформаційні платформи. Аналіз інструментальних засобів показав, що їх розвиток безпосередньо залежить від формальних методологій, які забезпечують узгодженість структури та функціональності. Особливу увагу було приділено моделі 5S/5SL, яка дозволяє цілісно описувати бібліотеку через п'ять взаємопов'язаних вимірів, формуючи фундамент для побудови системних моделей. Дослідження еволюції цифрових бібліотек продемонструвало поступову інтеграцію інтелектуальних механізмів управління знаннями та автоматизованих процесів оброблення даних. Розгляд архітектурних моделей дав можливість визначити ключові принципи побудови масштабованих і гнучких бібліотечних систем. Виявлено, що моделювання цифрових бібліотек вимагає узгодження інформаційних потоків, структур і сценаріїв взаємодії користувачів із ресурсами. Системний аналіз підходів до

моделювання уможливив формування узагальненої концептуальної основи, яка враховує як технічні, так і організаційні аспекти. Доведено, що саме комплексний підхід забезпечує повноцінну підтримку життєвого циклу цифрової бібліотеки. У підсумку розділ сформував цілісне бачення предметної області та окреслив ключові методологічні засади, необхідні для реалізації наступних етапів дослідження.

## **РОЗДІЛ 2. МЕТОДИ ТА АЛГОРИТМИ ІНСТРУМЕНТІВ МОДЕЛЮВАННЯ ЦИФРОВИХ БІБЛІОТЕК**

### **2.1. Огляд комплексних інструментів для створення цифрових бібліотек**

Існує низка чинних керівних принципів щодо конструювання ЦБ, більшість з яких акцентують увагу на плануванні проекту. Основна проблема цих керівництв полягає в тому, що вони безпосередньо не призводять до робочої версії ЦБ. Вони, як правило, припускають наявність у розробників та дизайнерів ЦБ достатнього технічного досвіду, що часто не відповідає дійсності.

Водночас, розроблено або перебуває у процесі розробки дедалі більше інструментів та технологій, призначених для допомоги у створенні ЦБ. До них відносяться інструменти для серверної сторони, такі як Mini SQL, Eprints, Harvest, Sprite, Real Audio, Dienst, ISite, SiteSearch та багато інших. Однак, більшість цих інструментів сфокусовані лише на одному або кількох компонентах цілісної системи ЦБ. Тому від розробників ЦБ все ще вимагається знання того, де використовувати ці інструменти та як їх інтегрувати між собою.

Далі детальніше розглянуто Greenstone. На відміну від згаданих вище інструментів, Greenstone є комплексним та простим у використанні інструментом, здатним створити завершену цифрову бібліотеку.

Greenstone — це програмний комплекс (suite of software), який має здатність обслуговувати колекції цифрових бібліотек та створювати нові колекції. Він пропонує новий підхід до організації інформації та її публікації в Інтернеті.

Для забезпечення широких вимог до ЦБ, Greenstone є відкритим програмним забезпеченням (open-source), що розповсюджується на умовах Загальної публічної ліцензії GNU (GNU General Public License).

Програмне забезпечення Greenstone задовольняє численні потреби ЦБ:

- Доступність через веббраузер.
- Повнотекстовий та пошук за полями (fielded search).
- Гнучкі засоби перегляду (browsing facilities).
- Автоматичне створення структур доступу.
- Використання наявних метаданих.
- Плагіни розширюють можливості системи.
- Документи можуть бути будь-якою мовою.

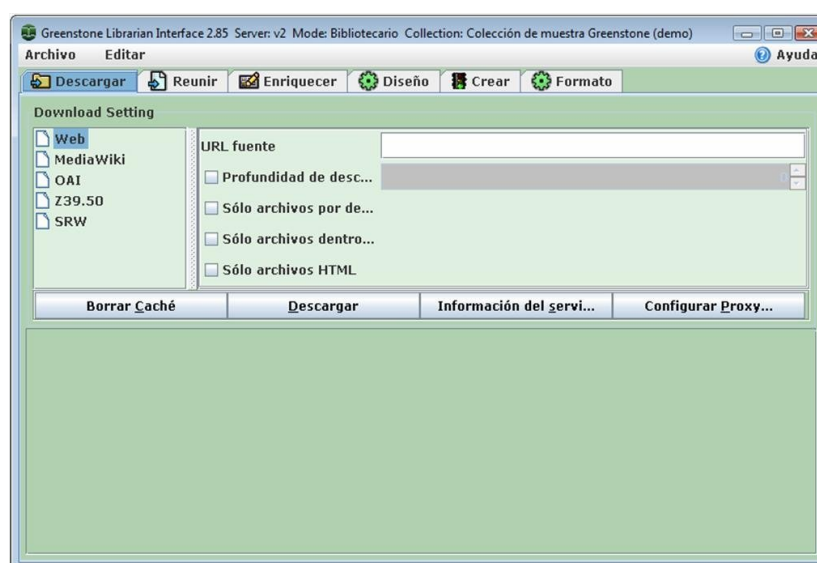


Рис. 2.1. Інтерфейс системи Greenstone для цифрових бібліотек

Однак, у порівнянні з моделлю 5S, концепція ЦБ у Greenstone є простішою та грубішою (simpler and coarser). Модель 5S здатна описати цифрову бібліотеку з більшої кількості перспектив і більш повно, ніж модель Greenstone.

З огляду на це, стверджується, що інструмент, заснований на моделі 5S, буде більш описовим, більш інтегрованим та більш формальним, ніж інструмент Greenstone.

Існують дві основні категорії інструментів для створення XML-файлів:

Категорія 1: звичайні редактори XML

До цієї категорії належать будь-які звичайні інструменти редагування XML. Вони є переважно текстовими редакторами з деякими специфічними функціями для створення XML-файлів. Сюди також відносяться ті XML-редактори, які поєднують текстовий редактор із деревоподібним представленням (tree views) даних.

## Категорія 2: інструменти моделювання XML

Більшість інструментів моделювання XML надають візуальний інтерфейс для користувачів. Вони відрізняються від інструментів моделювання, подібних до UML, які дозволяють будувати модель, а також від інструментів дизайну XML-схем, які використовуються для створення або редагування самої XML-схеми.

Розглянемо типовий робочий процес. Інструмент завантажує DTD або XML-схему і використовує інформацію про структуру, щоб допомогти користувачеві побудувати екземпляр XML (XML instance), що відповідає цій DTD/схемі. При цьому користувач може не мати прямого доступу до редагування фактичного тексту.

Xerlin - представник інструментів моделювання XML з відкритим кодом. Для роботи Xerlin вимагає DTD і надає інтерфейс для додавання, видалення, перестановки та редагування XML-елементів та атрибутів. DTD визначає, які елементи та атрибути є дійсними. Наразі Xerlin не підтримує XML-схему.

Xerlin – це репрезентативний приклад інструменту моделювання XML з відкритим кодом (open-source XML modeling tool). Він був розроблений в рамках ініціативи з відкритим кодом.

Основна функція Xerlin полягає в тому, щоб допомогти користувачам будувати екземпляри XML (XML instances) шляхом надання візуального інтерфейсу, який керується заданою схемою.

Для своєї роботи Xerlin вимагає наявності DTD (Document Type Definition). DTD використовується для формального визначення дійсної структури класу XML-файлів.

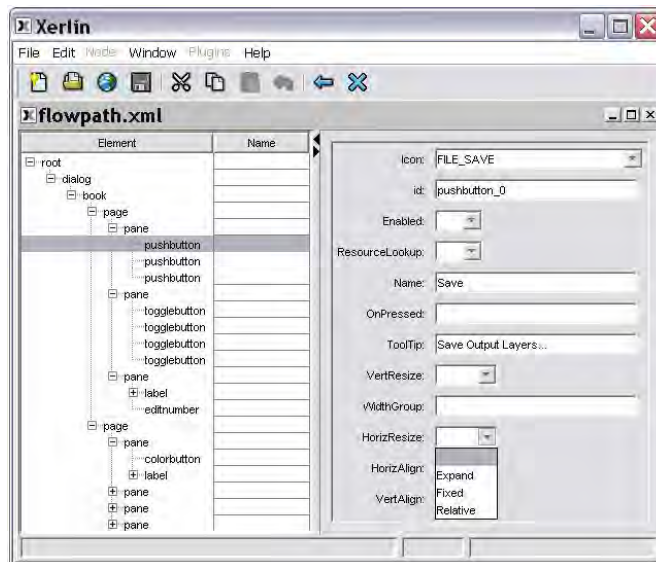


Рис. 2.2. Інструмент моделювання XML Xerlin

Замість ручного написання XML-коду, Xerlin надає користувачеві інтерфейс для маніпулювання елементами та атрибутами. Користувач може:

- Додавати нові XML-елементи.
- Видаляти існуючі елементи.
- Переставляти елементи.
- Редагувати XML-атрибути.

Як інструмент моделювання XML, він працює на рівні синтаксичної валідації (на основі DTD), що дає йому семантичні обмеження. Хоча XML-схема була запропонована для більш формального та строгого опису вмісту даних XML-файлів, оскільки DTD має серйозні обмеження, самої DTD чи XML-схеми недостатньо для опису складної системи з належною семантичною потужністю.

Моделювання, специфічне для домену (Domain-Specific Modeling) передбачає декларативне створення чіткої моделі (формальної чи неформальної) концепцій, взаємозв'язків та знань, що стосуються конкретного домену. DSM передбачає створення моделі, яка оперує концепціями та термінологією, безпосередньо взятими з цільового домену. Наприклад, для домену "цифрові бібліотеки", модель оперує об'єктами "Суспільства", "Сценарії" або "Потоки", а не загальними класами чи

об'єктами. Загальноприйнято вважати, що інструменти DSM є значно ефективнішими, ніж інструменти загального призначення (наприклад, UML) при виконанні завдань, специфічних для певного домену. Це пояснюється тим, що абстрактні концепції великих систем, які складно візуалізувати в загальних інструментах, у DSM легко відображаються на візуальні позначення, оскільки домен чітко визначений.

## 2.2. Оптимізація процесу генерації цифрових бібліотек на основі моделі 5S

### 2.2.1. Початковий процес генерації бібліотеки

Створення цифрової бібліотеки на основі формальної моделі 5S та мови специфікації 5SL спочатку передбачало три основні етапи:

1. Дизайнер ЦБ проводить аналіз системи, яку необхідно створити, згідно з п'ятьма аспектами моделі 5S (Потоки, Структури, Простори, Сценарії, Суспільства).
2. Специфікація 5SL, тобто дизайнер безпосередньо описує ЦБ у 5SL-файлі (XML-реалізації моделі 5S).
3. Файл 5SL разом із пулом компонентів (наприклад, SAX, DOM, процедури MARIAN, OAI harvester) передається генератору ЦБ, який створює новий робочий екземпляр ЦБ відповідно до специфікації.

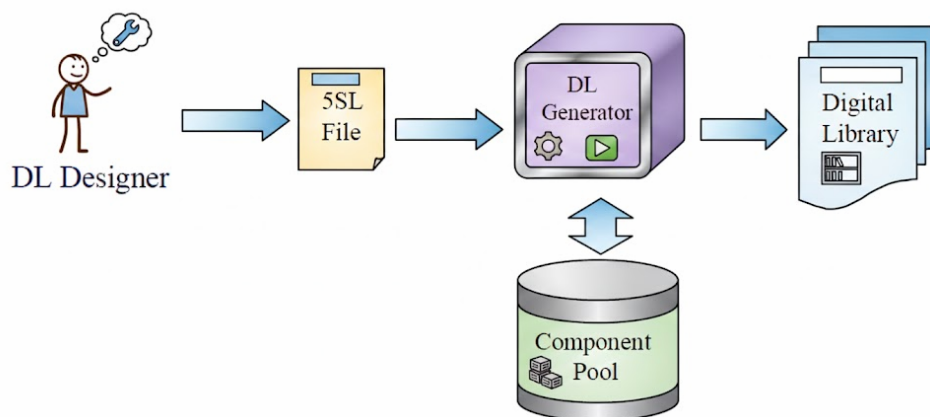


Рис. 2.3. Створення цифрових бібліотек за допомогою моделі 5S

Як було зазначено раніше, пряма розробка опису ЦБ у 5SL була обтяжливою і висувала надмірні вимоги до кваліфікації дизайнера:

- Дизайнеру необхідно було мати міцне та глибоке розуміння синтаксису та складної семантики 5SL.

- 5SL складається з п'яти окремих підмоделей, кожна з яких має унікальну синтаксичну та семантичну структуру. Це робило процес ручного створення повністю коректних файлів 5SL для всіх частин моделі надзвичайно складним.

- Через проблему узгодженості існували семантичні обмеження та залежності між підмоделями, що вимагало від дизайнерів додаткових зусиль для підтримки узгодженості (consistency) всієї моделі.

Для усунення цих недоліків було запропоновано впровадити додатковий проміжний етап — допоміжний інструмент між дизайнером ЦБ та генератором. Пропонований інструмент є візуальним інструментом моделювання, специфічним для домену, який забезпечує дружній графічний інтерфейс і автоматично генерує необхідні та коректні 5SL-файли.

Процес автоматичної генерації ЦБ за допомогою цього інструменту (рис. 2.4) трансформувався наступним чином:

1. Створення метамоделі експертом.

Експерт із цифрових бібліотек (який добре володіє моделлю 5S) створює метамодель на основі теорії 5S. Ця метамодель формально описує узагальнену цифрову бібліотеку.

2. Візуалізація мета моделі.

Інструмент моделювання обробляє цю метамодель, забезпечуючи візуалізацію її компонентів для кінцевого користувача.

3. Візуальне складання моделі дизайнером.

Дизайнер ЦБ використовує інструмент моделювання. Візуалізація метамоделі слугує керівництвом під час створення моделі, скорочуючи час навчання та допомагаючи зрозуміти структуру узагальненої ЦБ. Дизайнер

використовує ці візуалізовані компоненти для компоновання (складання) остаточної моделі власної ЦБ.

#### 4. Трансформація.

Після візуального опису, інструмент моделювання автоматично перетворює візуалізовану модель ЦБ на 5SL-файли, які відповідають синтаксису XML.

#### 5. Генерація робочого екземпляра.

Згенерований файл 5SL передається генератору ЦБ разом із пулом компонентів, що призводить до створення робочого екземпляра бажаної ЦБ.

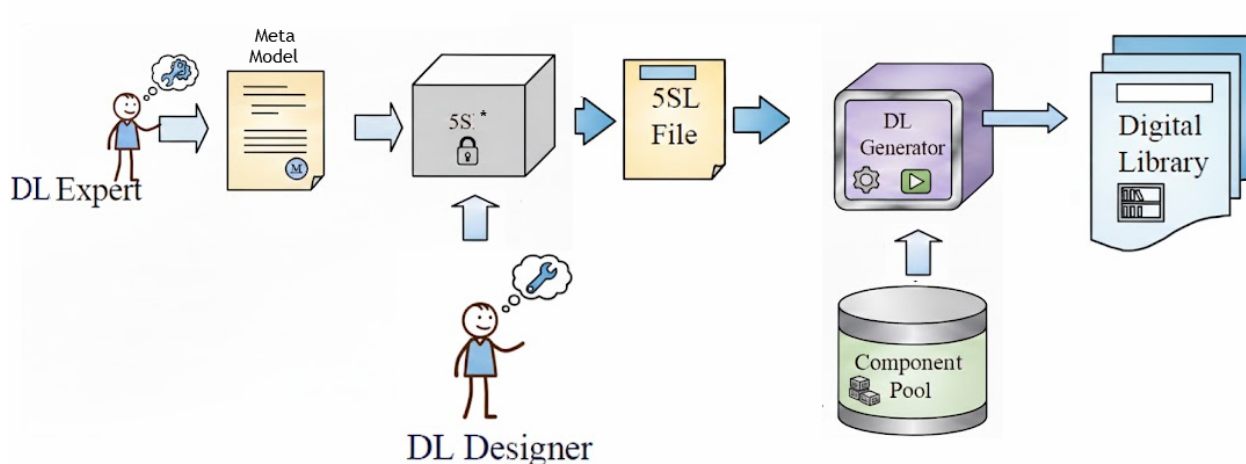


Рис. 2.4. Процес генерації цифрової бібліотеки з 5S\*

#### 2.2.2. Мінімізація складності та роль метамоделі

Завдяки візуальному інструменту моделювання (5S\*) складність роботи дизайнера мінімізується. Дизайнеру достатньо працювати лише з графічним інтерфейсом та об'єднувати візуальні компоненти, усуваючи необхідність запам'ятовувати деталі синтаксису та семантики 5SL. Це значно зменшує когнітивне навантаження, мінімізує помилки введення та забезпечує керівництво процесом моделювання. Крім того, коректність та узгодженість моделі можуть бути автоматично гарантовані функціоналом 5S\*.

У цьому підході використовується концепція метамоделі, яка описує узагальнену ЦБ. Модель для конкретної ЦБ є, по суті, екземпляром цієї

метамоделі, як показано на рисунку 2.5. Цей взаємозв'язок між метамоделлю та моделлю відповідає визначенням, встановленим у Meta Object Facility (MOF) OMG, хоча метамодель, використана в даній роботі, є специфічною для домену (домену створення ЦБ).

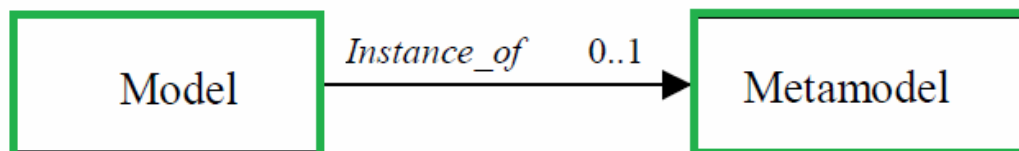


Рис. 2.5. Зв'язок між моделлю та метамоделлю

Слід підкреслити, що створення початкової метамоделі вимагає експерта з цифрових бібліотек, добре обізнаного з моделлю 5S. Роль цього експерта забезпечує гнучкість системи. Оскільки модель 5S все ще перебуває на ранній стадії розвитку, і очікуються майбутні зміни та доповнення, експерт з ЦБ відповідає за створення та модифікацію метамоделі відповідно до еволюції моделі 5S. Таким чином, отримуючи оновлену метамодель, інструмент 5S\* може бути використаний з майбутніми версіями теорії 5S.

### **2.3. Представлення концепції та архітектури інструменту моделювання цифрових бібліотек на основі домену**

Інструмент 5S\* є інструментом моделювання, специфічним для домену (domain-specific modeling tool), адаптованим для підтримки конструювання цифрових бібліотек (ЦБ).

Переваги моделювання:

- У моделюванні, специфічному для домену, модель складається з елементів, які належать виключно до конкретної предметної області, а не до сукупного універсального світу сутностей.

- Адаптація до мета моделі, тобто інструмент 5S\* спеціально адаптований для роботи з метамоделлю 5S, яка слугує його теоретичною основою.

- Оптимізація процесу, тобто можуть бути застосовані спеціалізовані методи, придатні лише для цього домену, з метою оптимізації процесу моделювання.

- Повторне використання в межах одного домену є більш реалістичним та ефективним, оскільки моделі в одній галузі демонструють більшу кількість спільних характеристик.

Модель 5S, як комплексна система, широко використовує існуючі стандартні мови опису (наприклад, UIML). Це обумовлено тим, що модель 5S охоплює багато підгалузей, для кожної з яких вже існують усталені специфікації.

Критично важливо, щоб мова специфікації 5SL могла ефективно взаємодіяти та інтегруватися з іншими стандартами. 5SL розроблена з урахуванням цієї можливості комбінування та інтеграції. Існує багато високорозвинених інструментів для різних підгалузей, інтегрованих у модель 5S. Метадані є ключовим елементом 5S. Існують загальноприйняті стандарти метаданих (наприклад, ETD-MS, Dublin Core), для яких доступні вже розроблені редактори метаданих, що можуть використовуватися для перегляду та редагування.

Частина Сценаріїв у моделі 5S може бути змодельована та описана за допомогою діаграм послідовностей UML. Для цієї мети можна використовувати існуючі інструменти моделювання UML.

Інструмент 5S\* спроектований як суперінструмент (super-tool). Це рішення дозволяє уникнути дублювання зусиль при виконанні завдань, для яких вже існують готові інструменти.

5S\* надає інфраструктуру, засновану на моделі 5S, і за потреби викликає існуючі зовнішні інструменти для виконання спеціалізованих функцій (наприклад, редагування метаданих чи моделювання сценаріїв).

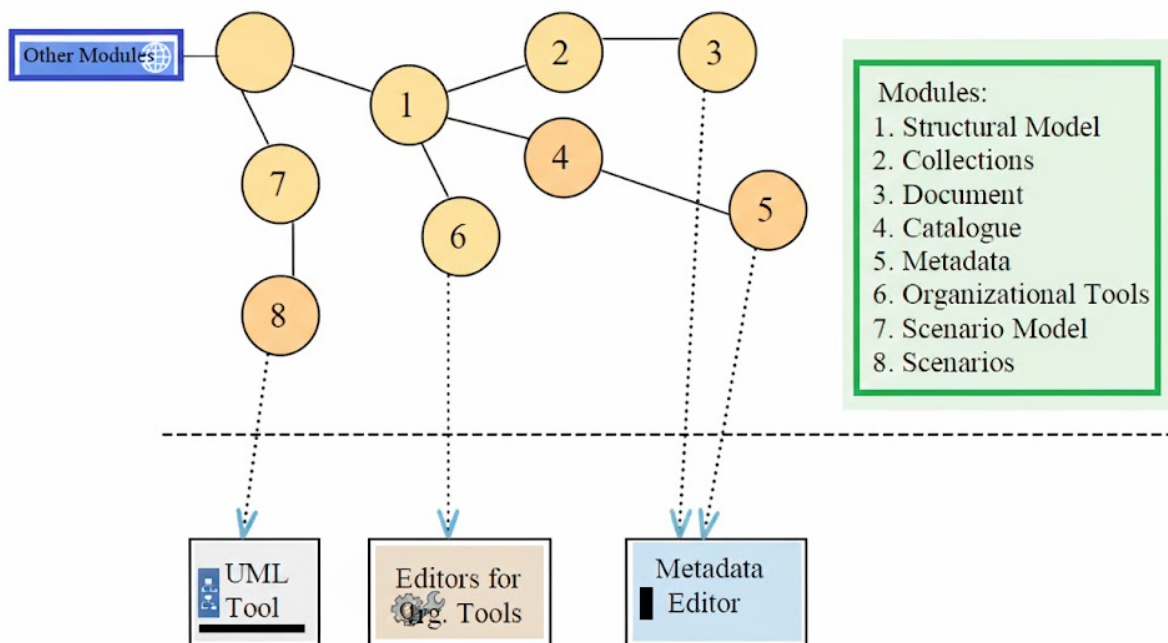


Рис. 2.6. Архітектура інструменту для моделювання бібліотек 5S\*

Оскільки описана робота є початковим етапом розвитку, поточна версія інструменту зосереджена лише на реалізації базової інфраструктури, залишаючи функціонал, пов'язаний із викликом зовнішнього програмного забезпечення, для майбутньої роботи.

## 2.4. Огляд формальної моделі 5S/5SL

У цьому розділі представлено опис ключових концепцій та взаємозв'язків моделі 5S та її мови специфікації 5SL. З огляду на значну складність моделі 5S, поданий огляд не є вичерпним, але охоплює основні компоненти та критичні залежності, на яких ґрунтується метамодель, використана в даній роботі.

Модель 5S забезпечує опис цифрових бібліотек (ЦБ) з п'яти різних, але взаємодоповнюючих перспектив. Різні S-моделі оперують різними примітивами та слугують різним цілям. Важливо відзначити, що ці моделі не є повністю незалежними; примітив в одній S-моделі може мати залежність

від примітива в іншій S-моделі. На рисунку 2.7 наведено загальний огляд моделі 5S.

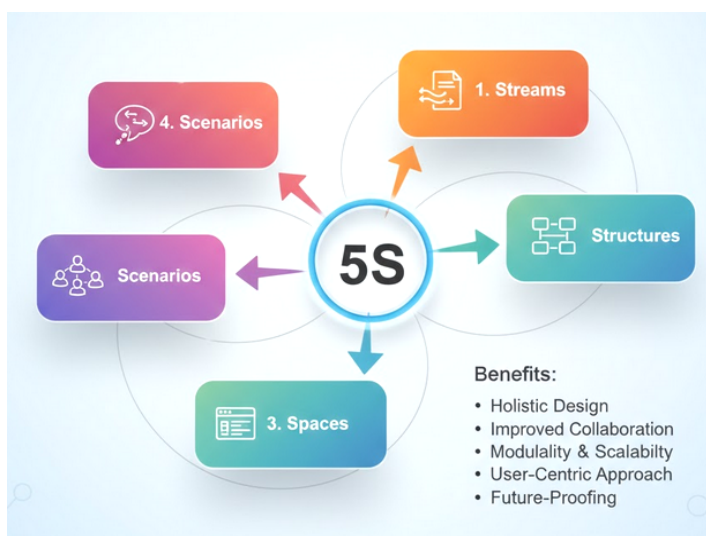


Рис. 2.7. Огляд фреймворку 5S для побудови цифрових бібліотек

#### 2.4.1. Модель потоків

Модель потоків визначає властивості цифрового вмісту, який підтримується ЦБ. Цифровий вміст може мати різну природу (наприклад, текст, відео, аудіо, зображення, програмні програми). Їхньою спільною характеристикою є можливість представлення як послідовності більш елементарних одиниць (наприклад, символів, бітів тощо). Поточна ієрархія моделі потоків представлена на рис. 2.8.

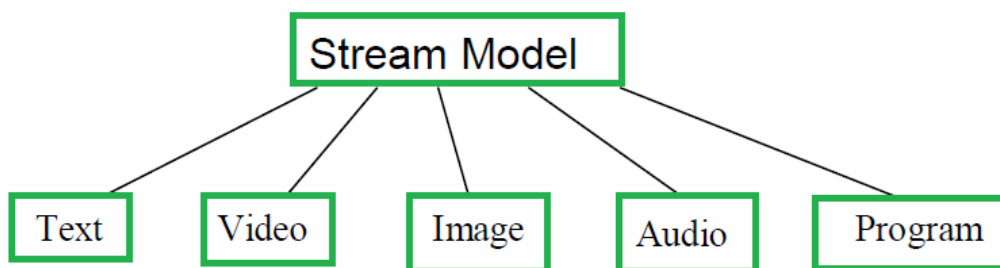


Рис. 2.8. Ієрархія моделі потоків

Властивості потоків включають кодування та формат послідовності. Для забезпечення стандартизації та взаємодії адаптовано офіційний інтернет-

стандарт MIME (багатоцільові розширення інтернет-пошти). Поля заголовків MIME використовуються як XML-елементи у 5SL.

Таблиця 2.1.

Опис потоків у 5SL

Поле заголовка MIME	Опис потоку	5SL приклад
Content-Type: text/plain; charset=ISO-8859-1	Текстовий потік в ETD (електронна теза)	<stream name='ETDText'> <MIME>text/plain</MIME> <charset>ISO-8859-1</charset> </stream>
Content-type: image/gif	Зображення в ETD	<stream name='ETDImage'> <MIME>image/gif</MIME> </stream>

#### 2.4.2. Структурна модель

Структурна модель визначає всі види структур у моделі 5S, які включають внутрішню структуру цифрового об'єкта, стандарти метаданих, властивості колекцій та каталогів, а також інструменти організації знань. Поточна версія ієрархії показана на рис. 2.9.

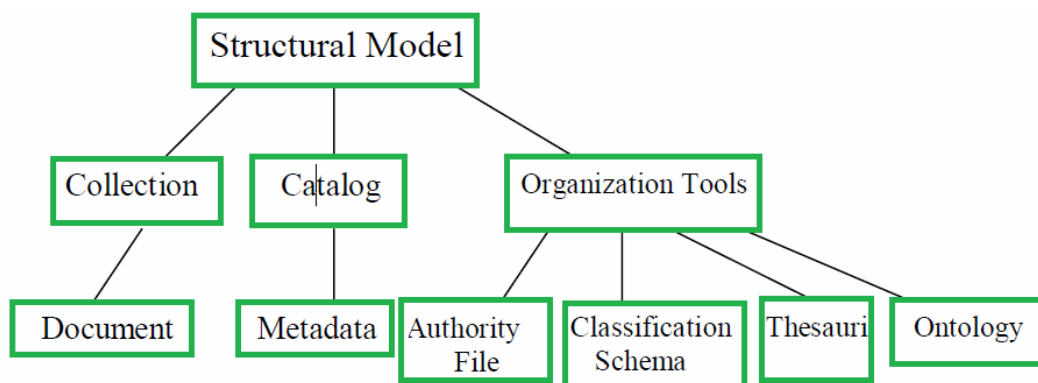


Рис. 2.9. Ієрархія структурної моделі

Цифрові об'єкти (документи) є фундаментальними будівельними блоками ЦБ. Формальне визначення цифрових об'єктів у моделі 5S полягає в накладенні структур на потік(и) або використанні структур для організації потоків у цифровий об'єкт. Формальне визначення:

$$\text{Цифровий об'єкт} = (h, SM, ST, StructuredStreams)$$

де:

$h \in H$  ( $H$  — множина універсально унікальних ідентифікаторів);

$SM = \{sm1, sm2, \dots, smn\}$  — множина потоків;

$ST = \{st1, st2, \dots, stn\}$  — множина структурних метаданих;

$StructuredStreams = \{stsm1, stsm2, \dots, stsmn\}$  — множина функцій  $StructuredStream$ , визначених на потоках ( $SM$ ) та на структурах ( $ST$ ).

$StructuredStream$  визначає поєднання структури та потоку. Приклад  $StructuredStream$  для електронної дисертації або тези (ETD) ілюструється на рисунку 2.10. Це текстовий потік, якому надано структуру.

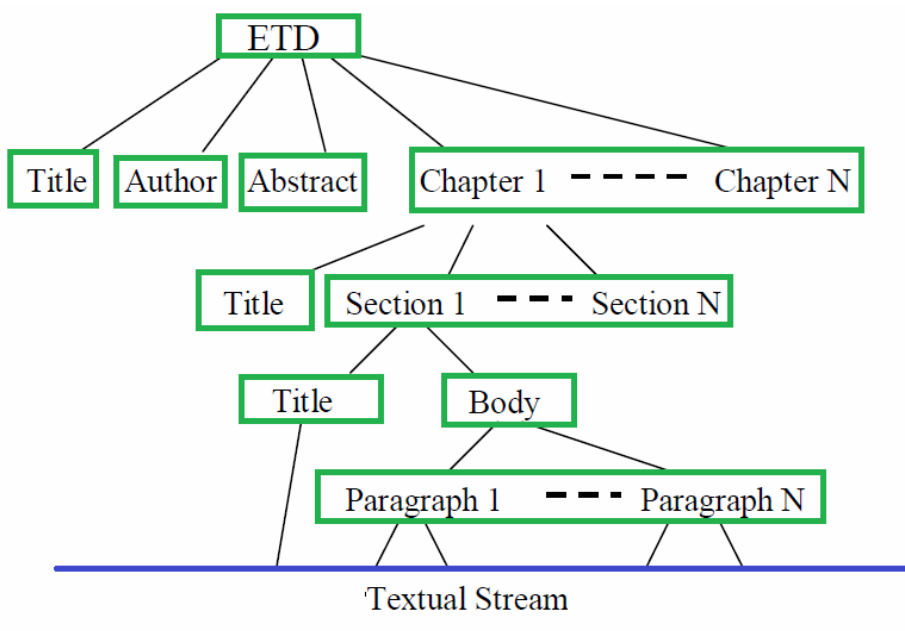


Рис. 2.10. StructuredStream для ETD

Приклад опису документа (ETD) у 5SL:

```
<document name='ETD'>
  <stream_enum>
    <stream> ETDText</stream>
    <stream> ETDAudio </stream>
    <stream> PDF </stream>
  </stream_enum>
  <Structured_stream> XMLSchema</structured_stream>
</document>
```

Семантичні обмеження:

- Потоки у визначенні цифрового об'єкта (документа) повинні бути попередньо визначені у Моделі потоків.

- Колекція складається з різних видів документів. Каталог описує колекцію, оскільки він агрегує всі адміністративні або описові метадані, що стосуються колекції. Таким чином, каталог залежить від колекції.

#### *2.4.3. Просторова модель*

Просторова модель описує компоненти ЦБ, які містять базовий метричний простір та реляційні операції над цим простором. Примітиви Просторовій моделі охоплюють індексування ЦБ для колекцій та каталогів, моделі пошуку та інтерфейс користувача (ІК).

Інтерфейс користувача є складною компонентою. Для його опису обрано існуючу модель на основі UIML (User Interface Markup Language), яка є XML-мовою. Теги UIML описують структуру елементів ІК (графічна кнопка, текстове поле), їхні властивості (колір, розмір) та поведінку при взаємодії. Планується, що в майбутньому інтерфейси в UIML будуть автоматично генеруватися зі сценаріїв.

#### *2.4.4. Модель сценаріїв*

Модель сценаріїв описує поведінку послуг цифрової бібліотеки. Примітиви включають послугу (service), подію (event), умову (condition) та дію (action). Діаграма послідовностей UML вважається ефективним візуальним інструментом для вираження різних сценаріїв.

#### *2.4.5. Модель суспільств*

Модель суспільств базується на концепції "спільноти" (community), що визначається як група людей або комп'ютерів, які мають спільні характеристики та поведінку. Модель складається з типів спільнот, функціональності спільнот та взаємозв'язків між ними.

Загалом, існують два основні типи спільнот: менеджери (Managers) та актори (Actors). Менеджери зазвичай представлені електронними агентами; відповідають за адміністративні та функціональні послуги, такі як інтерфейс, індекс, репозиторій, пошук, перегляд. Актори беруть участь та використовують будь-які послуги, визначені в моделі сценаріїв.

Семантичні обмеження:

- Послуги, в яких бере участь актор, повинні бути послугами, визначеними у моделі сценаріїв.

- Послуги, якими керує менеджер, також повинні бути послугами, визначеними у моделі сценаріїв.

### **Висновки до розділу**

Другий розділ дозволив дослідити інструменти та методи моделювання цифрових бібліотек, що забезпечують формалізоване проектування інформаційних систем цього класу. Аналіз існуючих рішень показав, що більшість інструментів зорієнтовані на підтримку повного життєвого циклу цифрової бібліотеки — від моделювання структур до генерації кінцевої реалізації. Особливу увагу приділено моделі 5S, яка дала змогу визначити чітку послідовність формування бібліотеки та виявити критичні аспекти оптимізації цього процесу.

Дослідження початкової генерації бібліотеки показало, що формалізація доменних сутностей суттєво зменшує складність проектування. Розглянута роль метамоделі продемонструвала, що вона є ключовим інструментом у мінімізації надмірності та забезпеченні структурної узгодженості. Аналіз формальної моделі 5S/5SL дав змогу детально описати можливий простір структур, потоків, сценаріїв та суспільних взаємодій у цифрових бібліотеках. Виявлено, що використання такого формального підходу забезпечує передбачуваність системної архітектури та підтримує багаторазове застосування модельних компонентів. Презентована концепція

інструменту моделювання, орієнтованого на домен, довела свою перевагу у створенні адаптивних бібліотечних платформ.

Розділ також окреслив технічні принципи, що сприяють автоматизації процедур генерації бібліотек. У підсумку встановлено, що поєднання формальних моделей та інструментальних засобів забезпечує високий рівень структурної і функціональної чіткості цифрових бібліотек.

## РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ МЕТАМОДЕЛЕЙ ДЛЯ ІНСТРУМЕНТІВ МОДЕЛЮВАННЯ ЦИФРОВИХ БІБЛІОТЕК

### 3.1. Специфікація метамоделі та її відображення в 5S\*

У цьому розділі представлено опис специфікації метамоделі та механізму її відображення в інструменті моделювання цифрової бібліотеки 5S\*.

#### 3.1.1 Вимоги до метамоделі

Метамодель є центральним і необхідним елементом функціонування інструменту 5S\*. Без неї інструмент функціонував би лише як порожня структура для ієрархічних каркасів без змістовного наповнення.

Метамодель виконує функцію опису узагальненої цифрової бібліотеки, надаючи будівельні блоки та встановлюючи зв'язки між ними. Вона базується на теорії 5S і має бути розроблена експертом із цифрових бібліотек, який володіє глибоким розумінням цієї теорії.

Вхідні дані від експертів до 5S\* охоплюють не лише формальну інформацію для моделювання, але й конфігураційну інформацію, необхідну для візуального представлення (наприклад, асоціації іконок з типами). Ця конфігураційна інформація також інтегрована у специфікацію метамоделі.

Узагальнюючи, метамодель повинна містити наступну інформацію:

- Типи компонентів (DataType);
- Властивості (Property) для кожного типу;
- Асоціації іконок з відповідними типами;
- Ієрархічну структурну інформацію (зв'язки між вузлами);
- Обмеження кардинальності (maxOccurs);
- Семантичні обмеження.

### 3.1.2. Огляд прикладу метамоделі

Метамодель 5S\* визначає два фундаментальні типи елементів: `DataType` (Тип Даних) та `DataSet` (Набір Даних). Кореневим елементом специфікації завжди є `DLMetaModel`. Для коректної обробки всі елементи `DataSet` повинні бути визначені перед елементами `DataType`.

На рисунку 3.1 наведено огляд прикладу метамоделі, представлений у форматі XML.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DLMetaModel>
  <Text_Content type="DataSet"/>
  <lang_content type="DataSet"/>
  <char_content type="DataSet"/>
  <Audio_Content type="DataSet"/>
  <Video_Content type="DataSet"/>
  <Image_Content type="DataSet"/>
  <Application_Content type="DataSet"/>
  <MetaData_Content type="DataSet"/>
  <RenderingContent type="DataSet"/>
  <RetrievalSpaceContent type="DataSet"/>
  <StemmingAlgContent type="DataSet"/>

  <Digital_Library type="DataType"/>

  <Stream_Model type="DataType"/>
  <Struct_Model type="DataType"/>
  <CollectionSet type="DataType"/>
  <Space_Model type="DataType"/>
  <Scenario_Model type="DataType"/>
  <Society_Model type="DataType"/>

  <Collection type="DataType"/>
  <Document type="DataType"/>
  <CatalogSet type="DataType"/>
  <Catalog type="DataType"/>
  <MetaDataFormat type="DataType"/>
  <Org_Tool type="DataType"/>
  <AuthorityFile type="DataType"/>
  <ClassificationSchema type="DataType"/>
  <Thesaurus type="DataType"/>
  <Ontology type="DataType"/>

  <UI type="DataType"/>
  <IR type="DataType"/>
  <Rendering type="DataType"/>
  <Index type="DataType"/>
  <Repository_MGR type="DataType"/>
</DLMetaModel>
```

Рис. 3.1. Огляд фрагменту метамоделі

Приклад демонструє, як ці елементи структуровані:

1. DataSet (набір даних) - визначає фіксовані, дискретні набори значень (переліки), які можуть бути використані як значення властивостей для елементів DataType. Наприклад:

- Text\_Content: містить дозволени типи MIME для текстового вмісту (наприклад, text/plain, text/html).

- MetaData\_Content: містить перелік підтримуваних стандартів метаданих (наприклад, Dublin Core, MARC).

2. DataType (тип даних) - визначає будівельні блоки ЦБ, що відповідають примітивам моделі 5S. Кожен DataType може містити:

- SubNodes встановлює ієрархічні зв'язки (композицію) з іншими DataType. Наприклад, Digital\_Library включає Stream\_Model, Struct\_Model і т. д. Атрибут maxOccurs (як у CollectionSet) задає обмеження кардинальності.

- Icon - встановлює асоціацію (наприклад, <Icon source="d\_lib.gif" />).

- Property - визначає атрибути та їхні типи. Тип choice використовує посилання на попередньо визначений DataSet (наприклад, <Property name="encoding" type="choice" dataset="char\_content" />).

Детальна структура, атрибути та візуальні позначення для елементів DataType та DataSet будуть детально пояснені у наступних розділах, включаючи приклади сценаріїв для кращого розуміння цих концепцій.

## 3.2. Специфікація елементів метамоделі

У цьому розділі детально описано два основні типи елементів, що використовуються у специфікації метамоделі для інструменту 5S\*: DataSet та DataType.

### 3.2.1. Елемент DataSet

DataSet являє собою діапазон дискретних значень, з яких можуть бути обрані допустимі варіанти. DataSet функціонує як джерело даних для елементів DataType і не має незалежної візуалізації в інструменті.

Атрибут `type="DataSet"` вказує на його приналежність до цього типу елементів. Елемент містить набір дочірніх елементів `<item>`, кожен з яких має атрибут `value`, що визначає допустиме значення.

Елемент `MetaData_Content` ілюструє визначення набору стандартів мета даних.

```
<MetaData_Content type="DataSet">
  <item value="Dublin Core" />
  <item value="IMS" />
  <item value="MARC" />
  <item value="RFC1807" />
  <item value="IEEE LTSC-LOM" />
</MetaData_Content>
```

Рис. 3.2. `MetaData_Content`

Цей приклад встановлює, що `MetaData_Content` містить значення `Dublin Core`, `IMS`, `MARC`, `RFC1807` та `IEEE LTSC-LOM`. Інші приклади `DataSet` можуть представляти набори типів зображень, аудіо-типів тощо, як показано нижче.

```
<Image_Content type="DataSet">
  <item value="image/gif" />
  <item value="image/jpeg" />
  <item value="image/png" />
</Image_Content>
<Audio_Content type="DataSet">
  <item value="audio/basic" />
  <item value="audio/wav" />
  <item value="audio/x-aiff" />
</Audio_Content>
```

Рис. 3.3. `Image_Content` та `Audio_Content`

### 3.2.2. Елемент `Data_Type`

`Data_Type` (тип даних) являє собою компонент метамоделі, який визначає тип, що може бути інстанційований у множину екземплярів у моделі, створюваній користувачем.

Наведемо приклад. До DataType відносяться такі ключові сутності, як digital\_library (цифрова бібліотека), stream\_model (модель потоків), struct\_model (структурна модель) та document (документ).

Елемент DataType візуалізується в інструментарії 5S\* як іменованний прямокутник. Кожен елемент DataType складається з трьох основних компонентів:

- SubNodes (Підвузли)
- Icon (Іконка)
- Property (Властивість)

У наступних підрозділах детально описано візуалізацію та функціональність SubNodes та Icon.

Приклад оголошення Digital\_Library та його візуальне позначення подано нижче.

```
<DataType name="Digital_Library">  
  <SubNodes>  
    <SubNode name="Stream_Model" />  
    <SubNode name="Struct_Model" />  
    <SubNode name="Space_Model" />  
    <SubNode name="Scenario_Model" />  
    <SubNode name="Society_Model" />  
  </SubNodes>  
  <Icon source="digital_library.gif" />  
</DataType>
```

Рис. 3.4. Digital\_Library та його візуальне позначення

Атрибут type підтверджує, що елемент Digital\_Library є DataType. Елемент SubNodes встановлює ієрархічну композицію, містячи інші елементи DataType, які є його дочірніми вузлами. У цьому прикладі, SubNodes елемента Digital\_Library складаються з п'яти ключових моделей 5S: Stream\_Model, Struct\_Model, Space\_Model, Scenario\_Model та Society\_Model.

Елемент Icon визначає графічне зображення, асоційоване з даним типом. Це зображення відображається ліворуч від прямокутника кожного

компонента в інструментарії та робочій області, сприяючи швидкій ідентифікації типу компонента користувачем.

### 3.3. Опис сценаріїв моделювання

Ці сценарії ілюструють, як дизайнер цифрової бібліотеки взаємодіє з елементами DataТуре метамоделі та як інструмент 5S\* застосовує правила кардинальності.

#### *Сценарій 1. Створення базової моделі ЦБ*

Мета: Продемонструвати використання елемента DataТуре у моделі користувача та механізм синхронізації.

Користувач розпочинає створення власної цифрової бібліотеки. Для цього він додає до своєї робочої області екземпляр Digital\_Library.

Перейменування: Користувач натискає на цей екземпляр та змінює його ім'я на My\_DL.

Синхронізація: Натискання на екземпляр активує механізм синхронізації інструменту. Інструмент розгортає тип Digital\_Library і відображає дочірні вузли цього типу (визначені в метамоделі) у панелі інструментів (toolbox).

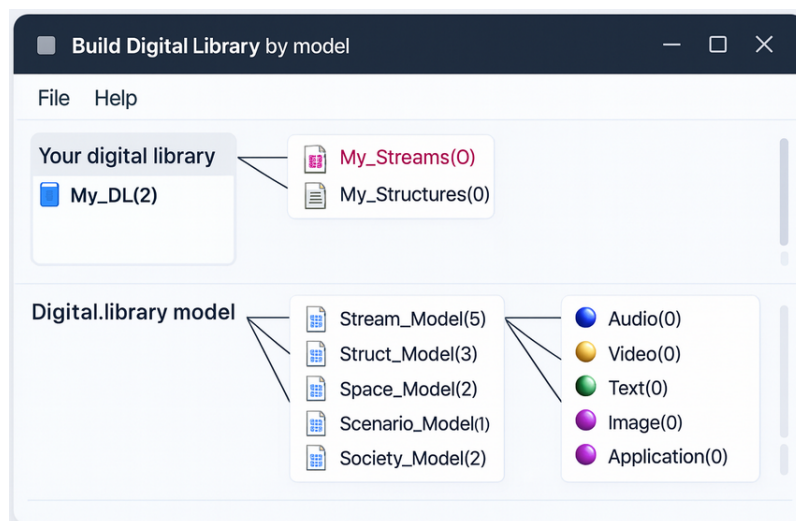


Рис. 3.6. Знімок екрана для сценарію 1

Додавання Вузлів: Тепер користувач може додавати необхідні дочірні вузли до своєї моделі. Він додає Stream\_Model (Модель Потоків) та Struct\_Model (Структурна Модель), перейменовуючи їх на My\_Streams та My\_Structures відповідно.

### Сценарій 2. Використання кардинальності для потоків

Мета: пояснити, як застосовується кардинальність (обмеження на кількість екземплярів) в елементі DataType.

Користувач вирішує деталізувати потоки у своїй Stream\_Model — My\_Streams.

Синхронізація: коли користувач натискає на My\_Streams, інструмент синхронізується, відображаючи дочірні вузли типу Stream\_Model у панелі інструментів. Це дозволяє користувачеві побачити, які типи потоків може містити його Stream\_Model.

Додавання екземплярів: користувач додає три види аудіо-потоків: Basic\_Audio, Wav\_Audio та Aiff\_Audio.

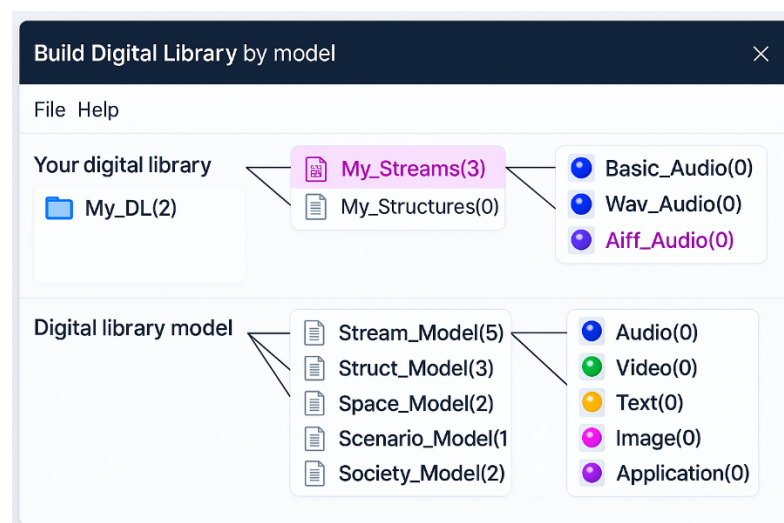


Рис. 3.7. Знімок екрана для сценарію 2

Слід звернути увагу на три аспекти:

1. Необмежена кількість.

Користувач може додавати будь-яку кількість аудіо-потоків. Це можливо, оскільки кардинальність елемента Audio у Stream\_Model

метамоделі вказує, що екземпляр Audio може з'являтися необмежену кількість разів в одного батьківського вузла.

## 2. Відображення кардинальності.

Кардинальність позначається лише в панелі інструментів (тобто в метамоделі), але не відображається безпосередньо в моделі користувача.

## 3. Ідентичність іконок.

Іконки всіх доданих екземплярів (Basic\_Audio, Wav\_Audio, Aiff\_Audio) є однаковими, оскільки вони належать до одного і того ж типу (Audio).

### 3.4. Опис властивостей елемента DataType у метамоделі 5S\*

Компонент Property (властивості) елемента DataType у метамоделі 5S\* використовується для специфікації атрибутів, якими повинен володіти цей елемент. Усі специфіковані властивості відображаються у таблиці властивостей (property sheet), асоційованій з кожним екземпляром DataType. Метамодель може як визначати специфічні властивості для елемента, так і не визначати жодних.

#### 1. Властивості за замовчуванням

Якщо для елемента не визначено жодних специфічних властивостей, він матиме лише властивості за замовчуванням у своїй таблиці властивостей.

Приклад Digital\_Library. Для елемента Digital\_Library не специфіковано жодних унікальних властивостей.

```
- <Digital_Library type="DataType">
- <SubNodes>
  <Stream_Model />
  <Struct_Model />
  <Space_Model />
  <Scenario_Model />
  <Society_Model />
</SubNodes>
  <Icon name="dl.gif" />
  <Property />
</Digital_Library>
```

Рис. 3.8. Digital\_Library та його таблиця властивостей

У цьому випадку, таблиця властивостей Digital\_Library містить лише властивості за замовчуванням, які притаманні кожному елементу DataType: 'Node Type' (тип вузла) та 'Enter a new name' (введіть нове ім'я).

## 2. Специфікація властивостей

Властивості, які можуть бути специфіковані в елементі Property, мають формат елементів інтерфейсу: випадаючий список (Combo Box) та текстове поле (Text Field).

### А. Випадаючий Список (Combo Box)

Використовується для вибору значення з попередньо визначеного DataSet. Приклад Audio: Елемент Audio (Аудіо) використовує ComboBox, щоб дозволити користувачеві вибрати тип вмісту з набору Audio\_Content.

```
<Audio_Content type="DataSet">
  <item value="audio/basic" />
  <item value="audio/wav" />
  <item value="audio/x-aiff" />
</Audio_Content>
.....
<Audio type="DataType">
  <Icon />
  <Property>
    <ComboBox name="content-type" src="Audio_Content" />
  </Property>
</Audio>
```

Рис. 3.9. Audio та його таблиця властивостей

Таблиця властивостей Audio міститиме три частини: 'Node Type', 'Enter a new name', та ComboBox з назвою "content-type". Вміст цього ComboBox отримується з DataSet "Audio\_Content".

### Б. Текстове поле (Text Field)

Використовується для введення довільного текстового значення.

Приклад Catalog: DataType Catalog має чотири визначені властивості, три з яких є текстовими полями.

```

- <Catalog type="DataType">
- <SubNodes>
  <MetaDataFormat constraint="*" />
</SubNodes>
<Icon />
- <Property>
  <TextField name="Description" />
  <TextField name="Creator" />
  <TextField name="Maintainer" />
  <ComboBox name="Collection" src="CollectionSet" />
</Property>
</Catalog>

```

Рис. 3.10. Catalog та його таблиця властивостей

Три текстові поля — “Description”, “Creator” та “Maintainer” — додаються до таблиці властивостей Catalog.

#### В. Атрибут завантаження (Load)

Якщо текстове поле має атрибут 'Load' зі значенням 'true', це означає, що значення властивості може бути завантажене з певного файлу на диску.

Приклад Ontology подано нижче.

```

- <Ontology type="DataType">
  <SubNodes />
  <Icon />
- <Property>
  <TextField name="Load_Ontology" load="true" />
</Property>
</Ontology>

```

Рис. 3.11. Ontology та його таблиця властивостей

### 3.5. Процес керування обмеженнями при моделюванні цифрової бібліотеки

Аналіз теорії 5S (розділ 2) виявив наявність певних внутрішніх семантичних обмежень у її ієрархічній структурі. Семантичні обмеження в

5S поділяються на дві категорії: обмеження значень (value constraint) та обмеження асоціацій (association constraint).

Обмеження значень специфікує діапазон можливих значень елемента.

Приклад: актор може брати участь лише у послугах, визначених у моделі сценаріїв.

Обмеження асоціацій визначає взаємозв'язки між різними компонентами.

Приклад: каталог повинен мати відношення 1:1 з Колекцією.

Інструмент 5S\* здатний реалізовувати та керувати цими обмеженнями. Наведені нижче приклади демонструють, як метамодель специфікує ці обмеження.

### 1. Обмеження значень

Приклад: актор може брати участь лише у послугах, визначених у моделі сценаріїв.

Оголошення actor (актор) та services (послуги) подано нижче.

```
- <Services type="DataType">
- <SubNodes>
  <Scenario constraint="*" />
</SubNodes>
  <Icon name="strM.gif" />
- <Property>
  <TextField name="Load_Services" load="true" />
</Property>
</Services>
- <Actor type="DataType">
  <SubNodes src="Services" />
  <Icon />
  <Property />
</Actor>
```

Рис. 3.12. Оголошення Actor та Services

У цьому оголошенні, частина SubNodes елемента Actor містить атрибут src зі значенням "Services". Це означає, що дочірні вузли Actor походять із джерела "Services". Таким чином, лише існуючі екземпляри Services можуть стати дочірніми вузлами Actor.

### Сценарій 3. Підтримка обмежень значень

Мета: продемонструвати, як 5S\* підтримує обмеження значень.

Ініціалізація актора: дизайнер створює модель ЦБ NDLTD. У моделі суспільств він додає новий екземпляр Actor під назвою “Students”.

Блокування додавання: На цьому етапі він не може додати послуги до “Students”, оскільки в моделі сценаріїв ще не визначено жодної послуги. Це підтримує обмеження значень.

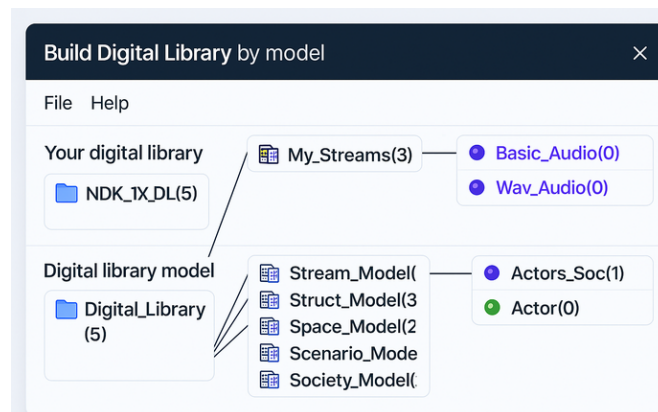


Рис. 3.13. Знімок екрана 1 для сценарію 3

Для визначення послуг дизайнер переходить до моделі сценаріїв і створює п'ять послуг: `fulltext_search`, `metadata_search`, `browsing`, `submission` та `training`.

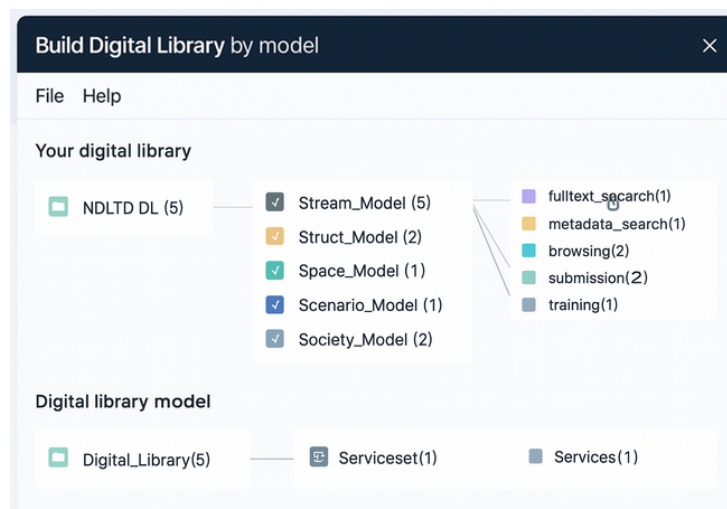


Рис. 3.14. Знімок екрана 2 для сценарію 3

Процес динамічного оновлення, коли дизайнер повертається до “Students”, він бачить, що п'ять послуг автоматично додалися до метамоделі під вузлом Actor. 5S\* динамічно підтримує цю залежність для користувача.

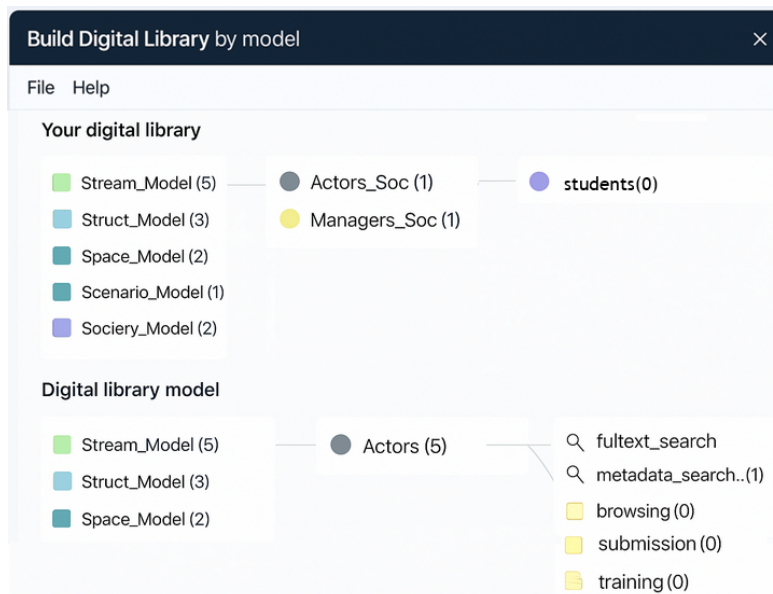


Рис. 3.15. Знімок екрана 3 для сценарію 3

Щодо додавання послуг, то дизайнер додає всі п'ять послуг до “Students”, що означає, що студенти можуть брати участь у всіх п'яти послугах, які надає ЦБ NDLTD.

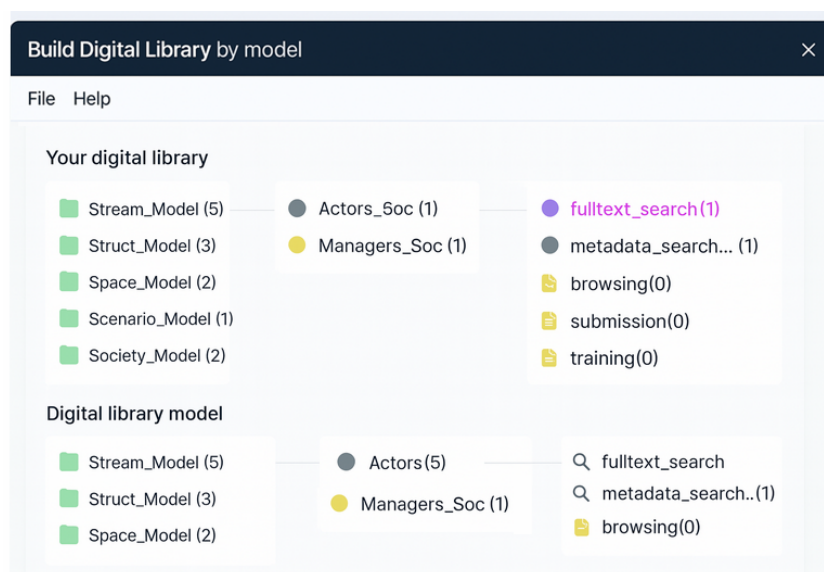


Рис. 3.16. Знімок екрана 4 для сценарію 3

## 2. Обмеження асоціацій

Приклад: каталог містить описові метадані для цифрових об'єктів у певній колекції. Тому Catalog повинен мати відношення 1:1 з Collection. Це означає, що Catalog не є незалежним і повинен залежати від однієї існуючої Collection.

Оголошення Catalog:

```
- <Catalog type="DataType">
- <SubNodes>
  <MetaDataFormat constraint="*" />
</SubNodes>
<Icon />
- <Property>
  <TextField name="Description" />
  <TextField name="Creator" />
  <TextField name="Maintainer" />
  <ComboBox name="Collection" src="CollectionSet" />
</Property>
</Catalog>
```

Рис. 3.17. Оголошення Catalog

У оголошенні ComboBox атрибут src посилається на CollectionSet. Якщо src посилається на DataSet, вміст ComboBox є фіксованим. Якщо src посилається на DataType (як у цьому випадку CollectionSet), вміст ComboBox залежить від вмісту екземпляра цього DataType. Вміст ComboBox динамічно змінюється відповідно до моделі користувача.

У даному випадку, вміст ComboBox залежить від того, які колекції дизайнер розмістив у CollectionSet.

Сценарій 4. Використання обмежень асоціацій

Мета: проілюструвати, як використовується обмеження асоціацій.

Створення колекцій: дизайнер створює модель ЦБ CITIDEL. Він створює дев'ять колекцій у своїй структурній моделі.

У таблиці властивостей Catalog поле ComboBox змушує його вибрати колекцію з існуючих колекцій для цього каталогу. Він обирає Virtual\_Union\_Collection і називає Catalog як Union\_Catalog.

Отже, наведені приклади підтверджують, що використання 5S\* робить керування семантичними обмеженнями (обмеженнями значень та асоціацій) дуже простим і зручним. Ця особливість 5S\* значно підвищує зручність використання інструменту (usability) та звільняє користувача від необхідності ручного запам'ятовування та підтримки обмежень.

### **3.6. Архітектурні рішення та дизайн інтерфейсу для моделювання**

Спостереження та аналіз свідчать, що деревоподібна структура у поєднанні з набором семантичних обмежень є достатньою для ефективного опису взаємозв'язків між основними компонентами моделі 5S. З огляду на це, для представлення як метамоделі, так і екземплярної моделі в інструменті 5S\* було обрано дерева.

Графічне представлення (наприклад, UML-діаграми) було відхилено, оскільки воно додає непотрібну складність компонування в інструменті. Вибір деревоподібної структури також зумовлений тим, що метамодель і модель специфікуються у форматі XML. Дослідження підтверджують, що деревоподібні структури добре підходять для відображення XML-екземплярів [25].

Основне завдання дизайну 5S\* полягало у визначенні того, як інструмент має допомагати користувачеві створювати модель, яка є коректним екземпляром метамоделі. Ключова вимога полягає в тому, що кожен компонент в екземплярній моделі повинен бути екземпляром певного типу, визначеного в метамоделі. Це породжує два наслідки:

- Екземпляр компонента повинен мати всі властивості відповідного типу метамоделі.
- Екземпляри компонентів повинні зберігати ті самі структурні взаємозв'язки, що й їхні відповідні типи в метамоделі.

Системи концептуальних карт [26] - ці системи використовують узагальнені візуальні компоненти та зв'язки з різними мітками для представлення різних концепцій та взаємозв'язків.

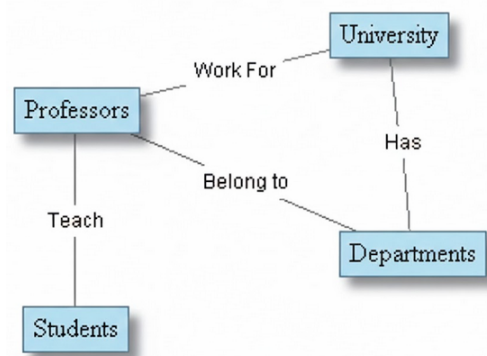


Рис. 3.18. Приклад концептуальної карти

Недоліком є те, що без попередньо визначених типів і структур користувач змушений створювати компоненти "з нуля" та вручну підтримувати коректні структурні взаємозв'язки, що є трудомістким і вимагає потужного механізму перевірки типів.

Системи з попередньо визначеними компонентами (наприклад, BeanBox, Khoros). Більшість інструментів моделювання використовують інструментарій для відображення всіх доступних візуальних компонентів, які можна вибирати та додавати до моделі користувача.

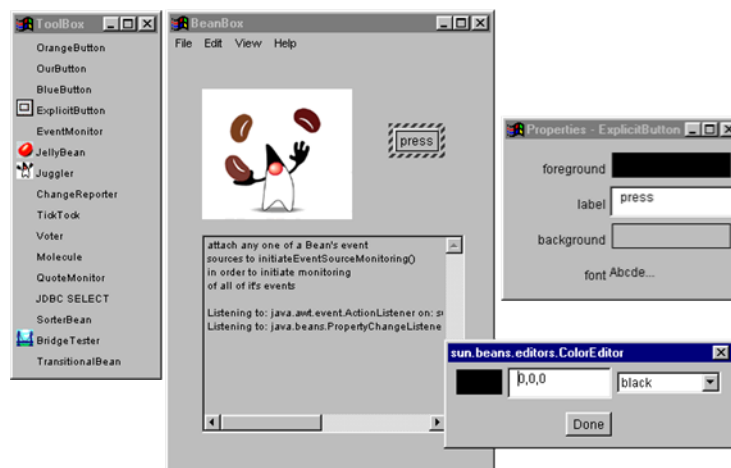


Рис. 3.19. Інструментарій BeanBox

Недоліком є те, що у таких інструментаріях структурні взаємозв'язки між візуальними компонентами не є видимими. Користувачеві потрібні глибокі знання метамоделі для підтримки правильних структурних взаємозв'язків, що вимагає підходу "проб і помилок".

Для подолання недоліків неструктурованих інструментаріїв у 5S\* впроваджено структурований інструментарій. Він не лише надає всі візуальні компоненти метамоделі, але й явно відображає структурні взаємозв'язки між цими компонентами.

Інструмент поділений на дві частини (рис. 3.20). Нижня частина - структурований інструментарій, який є візуалізацією метамоделі. Верхня частина - робоча область, де користувачі створюють свої екземплярні моделі.

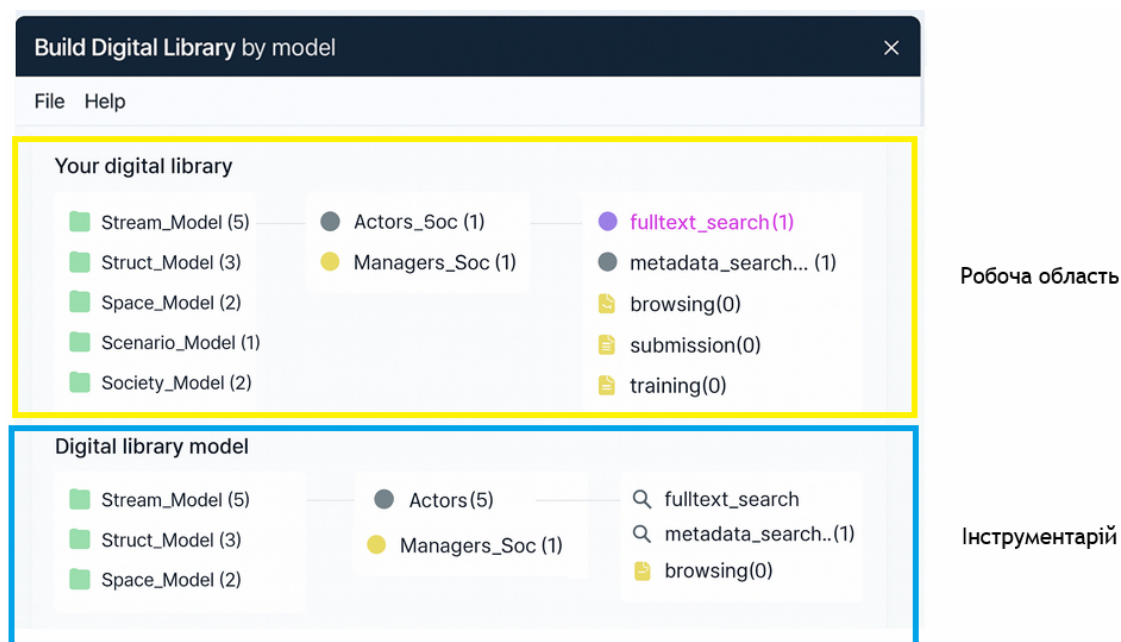


Рис. 3.20. Робоча область та структурований інструментарій у 5S\*

Усі візуальні компоненти в інструментарії та робочій області мають однакову форму (прямокутник). Це спрощує стратегію компонування та забезпечує узгоджені візуальні ефекти, оскільки сам інструмент не має попередньо визначених асоціацій форми/типу до отримання метамоделі. Імена екземплярів можуть змінюватися, але іконки залишаються незмінними,

що допомагає користувачеві підтримувати зв'язок між екземпляром та його типом. Компонент в інструментарії також містить індикатор кардинальності, який вказує, скільки екземплярів цього компонента можна додати до батьківського вузла в робочій області.

Аналіз структури моделі 5S свідчить про те, що її метамодель може бути ефективно представлена як деревоподібна структура, доповнена набором семантичних обмежень, специфічних для домену. Для візуалізації цієї ієрархії було розглянуто два основні підходи.

### 1. Підхід "Вузол та Зв'язок"

Традиційний метод візуалізації деревоподібних структур — це представлення за допомогою вузлів та зв'язків. Ця структура є кореневим орієнтованим графом, як показано на прикладі Java JTrees на рисунку 3.21.

Переваги:

- Підхід є простим для розуміння та інтуїтивно зрозумілим.
- Абсолютну глибину структури легко спостерігати, оскільки вузли на одному рівні мають однакову глибину.
- Зручність маніпулювання елементами.

Недоліки:

- Марнується значна частина доступного екранного простору.
- Погано масштабується; для великих структур необхідне прокручування. Наприклад, у JTree глибокі вузли можуть призводити до надмірного вертикального розширення.

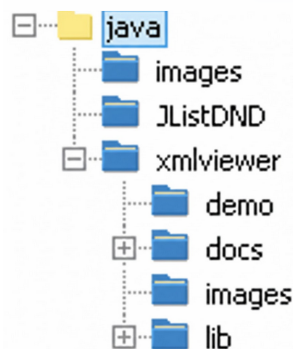


Рис. 3.21. Представлення "вузол-зв'язок" за допомогою JTree

Для подолання проблем візуалізації традиційних дерев були розроблені варіанти, такі як гіперболічне дерево та конусне дерево. Однак ці варіанти покращують візуалізацію за рахунок зменшення зручності маніпуляції.

## 2. Вкладений підхід

Треетар [27] використовує вкладений метод відображення і є абсолютно відмінним від підходу "вузол-зв'язок".

Дерева відображаються за допомогою двовимірного заповнення простору, де вузли представлені прямокутниками, а дочірні вузли вкладені всередину прямокутника свого батьківського вузла.

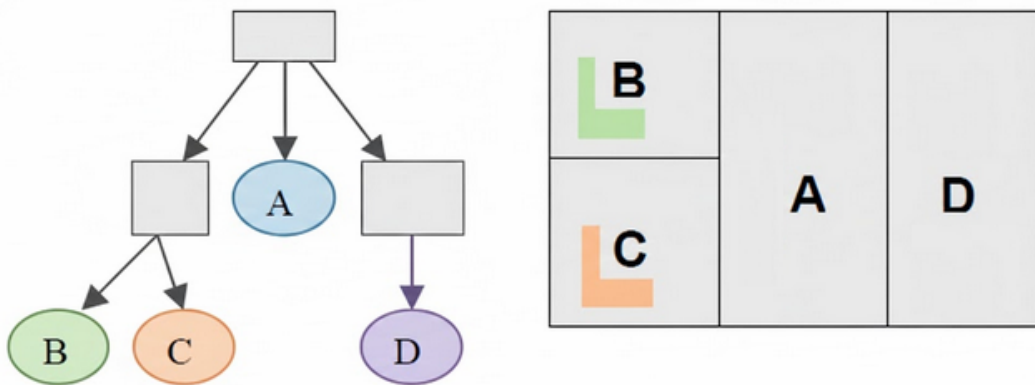


Рис. 3.22. Представлення "вузол-зв'язок" та Треетар дерева

Переваги Треетар:

- Краще використання екранного простору.
- Здатність візуалізувати великі та складні ієрархії.

Недоліки:

- Глибокі вузли можуть бути дуже малими, що вимагає функціоналу масштабування.
- Інформація про глибину структури не є очевидною.

Для 5S\* було обрано традиційне представлення "вузол та зв'язок". Ключова причина цього вибору полягає в чіткості інформації про глибину

структури, тоді як вкладений підхід Треетар не передає цю інформацію так виразно.

Представлення "вузол та зв'язок" (див. рис. 3.23) чітко демонструє відповідність рівнів між деревом метамоделі та деревом екземплярної моделі користувача.

#### 1. Вимога коректності.

Вимагається, щоб рівень екземпляра компонента в його екземплярному дереві відповідав рівню відповідного компонента типу в дереві метамоделі.

#### 2. Пріоритет зрозумілості.

Оскільки для дизайнера інструменту є критично важливим мати чіткий та зрозумілий огляд інформації про глибину обох дерев, було надано перевагу представленню "вузол та зв'язок".

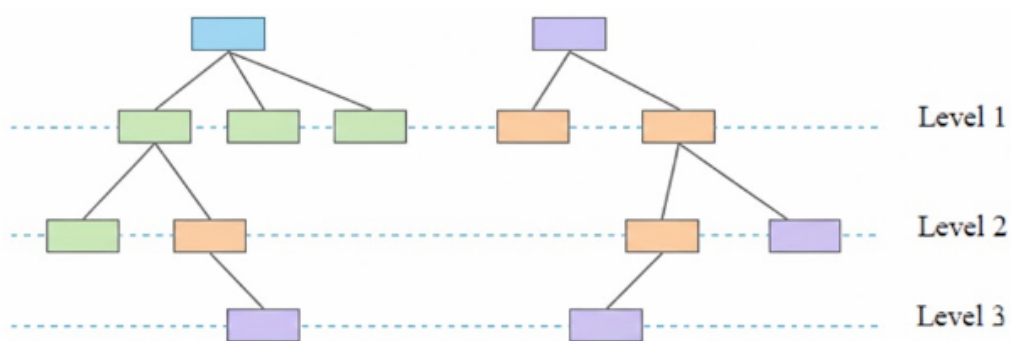


Рис. 3.23. Представлення "вузол-зв'язок" має чітку інформацію про глибину

### 3.7. Специфікація метамоделі та її роль у моделюванні цифрових бібліотек

Специфікація метамоделі та відповідні сценарії, представлені в цьому розділі, містять вичерпні відповіді на наступні ключові питання щодо метамоделі:

#### 1. Типи в метамоделі (Types).

Визначено фундаментальні елементи, що становлять основу узагальненої цифрової бібліотеки, а саме Data Type (типи даних) та DataSet (набори даних).

## 2. Властивості типів (Properties).

Встановлено, які атрибути повинні мати ці типи, які відображаються у таблиці властивостей (property sheet). Ці властивості реалізуються через інтерфейсні елементи, такі як ComboBox та Text Field.

## 3. Візуальні асоціації (Icons).

Вказано, які графічні зображення (іконки) повинні використовувати типи для візуальної ідентифікації в інструменті.

## 4. Ієрархічні структурні зв'язки (Hierarchical Relationships).

Описано взаємозв'язки композиції між типами за допомогою елемента SubNodes.

## 5. Кардинальність (Cardinality).

Визначено обмеження щодо кількості екземплярів, які може мати тип, що відображається в панелі інструментів (наприклад, через maxOccurs).

## 6. Представлення семантичних обмежень (Semantic Constraints).

Показано, як інструмент представляє та керує внутрішніми семантичними обмеженнями, поділеними на обмеження значень (value constraints) та обмеження асоціацій (association constraints).

## 7. Створення екземпляра моделі (Instance Model Creation).

Продемонстровано, як на основі метамоделі можна створити конкретну модель ЦБ (екземпляр моделі користувача) за допомогою інструменту 5S\*.

На основі детальної специфікації метамоделі, інструмент 5S\* виконує не лише відображення ієрархічної структури метамоделі та моделі користувача, але й забезпечує дотримання внутрішніх семантичних обмежень. Це гарантує узгодженість (consistency) та коректність (correctness) моделі, створеної користувачем.

Отже, було представлено візуальний інструмент моделювання, специфічний для домену, 5S\* створений для цілей моделювання та побудови

цифрових бібліотек на основі формальної моделі 5S. Розробка 5S\* є важливим кроком у проєкті, спрямованому на швидку генерацію цифрових бібліотек. Процес швидкої генерації вимагає від дизайнерів опису їхніх ЦБ за допомогою специфічної мови опису (5SL) на першому етапі. 5S\* суттєво зменшує складність цього початкового етапу для дизайнерів, які не є експертами у 5SL. Наскільки нам відомо, наразі не було розроблено жодного іншого подібного інструменту моделювання, орієнтованого на цю конкретну мету.

Ключові характеристики:

- 5S\* ґрунтується на метамоделі, яка формально описує узагальнену цифрову бібліотеку з використанням теорії 5S.

- Інструмент допомагає дизайнерам створювати їхні екземплярні моделі (описи ЦБ) у форматі 5SL без необхідності знати деталі, синтаксис та семантику самої мови 5SL.

- 5S\* відображає метамодель у вигляді структурованого інструментарію (structured toolbox). Це забезпечує візуалізацію як окремих компонентів, так і взаємозв'язків між ними, а також надає дизайнерам спеціалізовані візуальні компоненти, специфічні для домену.

### **Висновки до розділу**

У третьому розділі було розкрито процес імплементації метамodelей, які слугують основою для інструментального моделювання цифрових бібліотек. Дослідження специфікації метамodelі продемонструвало, що формалізація структур, типів даних і взаємодій значно полегшує процес проєктування складних інформаційних систем. Розгляд вимог до метамodelі дав змогу визначити ключові критерії, що включають гнучкість, розширюваність та відповідність доменній логіці цифрової бібліотеки. Аналіз прикладу метамodelі дозволив показати, як абстрактні структурні елементи

трансформуються у формальні об'єкти, здатні відобразити реальні бібліотечні процеси. Окрема увага була приділена елементам DataSet і DataType, які є центральними для визначення інформаційних структур у межах проєктованого інструментарію моделювання. Розділ детально описав сценарії моделювання, що продемонстрували послідовність створення бібліотеки у межах запропонованої метамоделі. Також було розкрито механізми керування обмеженнями, що забезпечують логічну цілісність і правильність побудови моделі.

Дослідження архітектурних рішень і принципів дизайну інтерфейсу показало, що ефективний інструмент моделювання потребує поєднання формальної строгості й зручної інтерактивності. Розділ підкреслив значення метамоделі у забезпеченні автоматизації, структурної узгодженості та можливості повторного використання компонентів. У підсумку встановлено, що імплементація метамоделі створює передумови для побудови універсальних і масштабованих інструментів проєктування цифрових бібліотек.

## ВИСНОВКИ

Магістерське дослідження, присвячене засобам та інструментам моделювання цифрових бібліотек, дало змогу всебічно проаналізувати теоретичні підходи, методологічні засади та практичні механізми побудови сучасних цифрових бібліотечних систем. У роботі комплексно досліджено концептуальні основи цифрових бібліотек, проаналізовано інструментальні засоби для їх створення й запропоновано метамодельний підхід, що забезпечує формалізацію, уніфікацію та оптимізацію процесів моделювання.

У першому розділі здійснено аналіз предметної області цифрових бібліотек. Показано, що цифрова бібліотека є складною інформаційною системою, яка поєднує структурні, функціональні, технологічні та соціальні компоненти. Визначено еволюційний шлях розвитку концепції цифрових бібліотек: від репозитаріїв електронних документів — до комплексних інтерактивних платформ, що підтримують управління знаннями, інформаційний пошук, автоматизовану організацію даних та соціальну взаємодію користувачів. Теоретично обґрунтовано застосування різних архітектурних моделей цифрових бібліотек, включно з моделюванням на основі парадигми 5S/5SL, яка описує систему через п'ять взаємопов'язаних вимірів: простори (Spaces), потоки (Streams), структури (Structures), сценарії (Scenarios) та суспільства (Societies). Доведено, що моделювання цифрових бібліотек потребує комплексного підходу, який дозволяє узгодити концепцію, архітектуру, інформаційну модель та функціональність у єдиному формалізованому середовищі.

У другому розділі зосереджено увагу на інструментах і методах моделювання, що застосовуються у процесі створення цифрових бібліотек. Проаналізовано існуючі платформи та програмні рішення, їхню архітектуру, функціональні можливості та обмеження. Особливу увагу приділено процесу генерації цифрової бібліотеки на основі моделі 5S, який є формальним механізмом опису структури та поведінки інформаційного середовища.

Показано, що оптимізація цього процесу можлива через використання метамodelей, які мінімізують складність представлення елементів, забезпечують повторюваність та узгодженість інструментальних рішень. У дослідженні уточнено зміст кожної компоненти моделі 5S/5SL, визначено зв'язки між ними та окреслено можливості застосування моделей для створення адаптивних та масштабованих цифрових бібліотек. Запропоновано архітектурний підхід до побудови інструменту моделювання, орієнтованого на доменну специфікацію та формальну логіку побудови цифрової бібліотеки.

У третьому розділі представлено результати імплементації метамodelьного підходу для інструментів моделювання цифрових бібліотек. Побудовано специфікацію метамodelі, яка розширює класичну 5S-архітектуру та підтримує доменні сутності, що є критично важливими для опису цифрових бібліотек. Розроблено формальні визначення ключових елементів метамodelі, зокрема DataSet та DataType, що дозволяють стандартизувати структури даних, типи ресурсів та їх поведінкові характеристики. Описано сценарії моделювання, які демонструють практичне застосування метамodelі під час проєктування складних бібліотечних систем. Особливо наголошено на механізмах керування обмеженнями — правил, які забезпечують коректність моделі та запобігають порушенню логічної цілісності системи. Також сформовано архітектурні рішення для інтерфейсу моделювання, що орієнтовані на забезпечення інтуїтивності, гнучкості та підтримки масштабованості. Доведено, що використання метамodelі підвищує рівень автоматизації, забезпечує адаптивність бібліотечних платформ і сприяє формалізованому підходу до побудови складних цифрових інфраструктур.

Узагальнюючи результати дослідження, слід зазначити, що створення цифрових бібліотек у сучасних умовах вимагає чіткої методологічної основи, інструментальної підтримки та формальних моделей, які забезпечують коректність, системність і масштабованість. Запропонований у роботі підхід

на основі моделі 5S, доповнений метамодельними специфікаціями, забезпечує можливість створення уніфікованих інструментів моделювання, які здатні описувати складні інформаційні системи на різних рівнях абстракції та реалізації. Результати дослідження формують теоретичний і практичний фундамент для подальшої розробки інструментів моделювання, автоматизації процесів генерації цифрових бібліотек, інтеграції формальних моделей у системи підтримки проектування та створення адаптивних інформаційних платформ нового покоління.

Таким чином, магістерська робота не лише узагальнює існуючі наукові підходи до моделювання цифрових бібліотек, але й пропонує власне комплексне рішення, що може бути використане як основа для розробки сучасних систем цифрової бібліотечної інженерії.

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. XML: Validating SML Dialogs Created in XML - 69validate.pdf . - <https://www.microimages.com/documentation/TechGuides/69validate.pdf>
2. A User-Oriented Approach to Scheduling Collection Building in Greenstone. - [https://www.researchgate.net/publication/33051964\\_A\\_User-Oriented\\_Approach\\_to\\_Scheduling\\_Collection\\_Building\\_in\\_Greenstone](https://www.researchgate.net/publication/33051964_A_User-Oriented_Approach_to_Scheduling_Collection_Building_in_Greenstone)
3. A Framework for Building Open Digital Libraries. - [https://www.researchgate.net/publication/228367358\\_A\\_Framework\\_for\\_Building\\_Open\\_Digital\\_Libraries](https://www.researchgate.net/publication/228367358_A_Framework_for_Building_Open_Digital_Libraries)
4. MARIAN: Searching and Querying across Heterogeneous Federated Digital Libraries. - [https://www.researchgate.net/publication/2374495\\_MARIAN\\_Searching\\_and\\_Querying\\_across\\_Heterogeneous\\_Federated\\_Digital\\_Libraries](https://www.researchgate.net/publication/2374495_MARIAN_Searching_and_Querying_across_Heterogeneous_Federated_Digital_Libraries)
5. Fox, Edward A., Gonçalves, M.A. and Shen, Rao. Book Review: Theoretical foundations for digital libraries. San Raphael, CA: Morgan & Claypool Publishers, 2012. XXII, 158 p. ISBN 978-1-60845-910-0 - <https://informationr.net/ir/reviews/revs455.html>
6. Extending the 5S Digital Library (DL) Framework: From a Minimal DL towards a DL Reference Model. - [https://www.researchgate.net/publication/240954341\\_Extending\\_the\\_5S\\_Digital\\_Library\\_DL\\_Framework\\_From\\_a\\_Minimal\\_DL\\_towards\\_a\\_DL\\_Reference\\_Model](https://www.researchgate.net/publication/240954341_Extending_the_5S_Digital_Library_DL_Framework_From_a_Minimal_DL_towards_a_DL_Reference_Model)
7. Gonçalves, M. A., Fox, E. A., Watson, L. T., & Kipp, N. A. (2003). Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries. Virginia Polytechnic Institute & State University, Department of Computer Science, Technical Report TR-03-04.
8. Zhu, Q. (2002). 5SGraph: A Modeling Tool for Digital Libraries. Master's Thesis, Virginia Tech.
9. Murthy, U., Gorton, D., Torres, R. da S., Gonçalves, M. A., Fox, E. A., & Delcambre, L. M. L. (2007). Extending the 5S Digital Library (DL)

- Framework: From a Minimal DL Towards a DL Reference Model. Paper presented at the First Digital Library Foundations Workshop, held in conjunction with the 2007 Joint Conference on Digital Libraries, Vancouver, Canada, June 18–24, 2007.
10. Shen, R. (2006). Applying the 5S Framework to Integrating Digital Libraries. Virginia Tech.
  11. Isah, A., Mutshewa, A., Serema, B. C., & Kenosi, L. (2013). Digital Libraries: Analysis of DELOS Reference Model and 5S Theory. *Journal of Information Science Theory and Practice*, 1(4), 38–47. <https://doi.org/10.1633/JISTaP.2013.1.4.3>
  12. Alzahrani, A., Mahmud, I., Ramayah, T., & Alfarraj, O. (2017). Modelling Digital Library Success Using the DeLone and McLean Information System Success Model. *Journal of Librarianship and Information Science*, 51(2). <https://doi.org/10.1177/0961000617726123>
  13. Krafft, D. B., Birkland, A., & Cramer, E. J. (2008). NCore: Architecture and Implementation of a Flexible, Collaborative Digital Library. In Proceedings of [preprint] — National Science Digital Library (NSDL) project report.
  14. Al-Natsheh, H. T., Martinet, L., Muhlenbach, F., Rico, F., & Zighed, D. A. (2018). Metadata Enrichment of Multi-Disciplinary Digital Library: A Semantic-based Approach.
  15. Tiurkedzhy, N. S., Davydova, I. O., Marina, O. Yu., & Marin, S. O. (2022). Accessibility Analysis of Digital Libraries and Specialized Library Resources. *University Library at a New Stage of Social Communications Development. Conference Proceedings*, No. 7, 218–231.
  16. Yaroshenko, T. O., & Iaroshenko, O. I. (2024). Digital Curation: Opportunities and Challenges for Ukrainian Libraries. *University Library at a New Stage of Social Communications Development. Conference Proceedings*, No. 9, 114–119.

17. Ranjna Assistant Librarian. (2014). A Reference Quality Model and Framework (5S) for Good Digital Libraries: Issues and Applications. *Global Journal of Multidisciplinary Studies*, 3(11).
18. Moreira, B. L., Golçalves, M. A., & Fox, E. A. (2007). What is a Good Digital Library? — A Quality Model for Digital Libraries. *Information Processing & Management*, 43.
19. Lomachinska, I., Lomachynskyi, B., & Rykhlitska, O. (2024). Digital Culture in Scientific Libraries. *Skhid*, 6(1), 43–49. [skhid.kubg.edu.ua](http://skhid.kubg.edu.ua)
20. Tiurkedzhy, N. S., Davydova, I. O., Marina, O. Yu., & Marin, S. O. (2022). Analysis of Digital Libraries and Specialized Library Resources. *University Library at a New Stage of Social Communications Development*.
21. Agosti, M., Ferro, N., Fox, E. A., Gonçalves, M. A., & Lagoeiro, B. (2007). Towards a Reference Quality Model for Digital Libraries. (ПРОЕКТ DELOS Reference Model)
22. Fox, E. A., & Gonçalves, M. A. (2004). Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Digital Library Framework and Its Applications. Dissertation, Virginia Tech.
23. (Assorted) Authors. (2025). First Workshop on Building Innovative Research Systems for Digital Libraries (BIRDS 2025).
24. Krafft, D. B., Birkland, A., & Cramer, E. J. (2008). NCore: Architecture and Implementation of a Flexible, Collaborative Digital Library. arXiv preprint.
25. Lagoze, C., Arms, W., Gan, S., Hillmann, D., Ingram, C., Krafft, D., Marisa, R., Phipps, J., Saylor, J., Terrizzi, C., Hoehn, W., Millman, D., Allan, J., Guzman-Lara, S., & Kalt, T. (2002). Core Services in the Architecture of the National Digital Library for Science Education (NSDL).
26. Van de Sompel, H., Bekaert, J., Liu, X., Balakireva, L., & Schwander, T. (2005). aDORe: a modular, standards-based Digital Object Repository.

27. Lagoze, C., Van de Sompel, H., Nelson, M. L., Warner, S., Sanderson, R., & Johnston, P. (2008). Object Re-Use & Exchange: A Resource-Centric Approach.
28. Kostelic, C. S. (2017). Applying the Levels of Conceptual Interoperability Model to a Digital Library Ecosystem — A Case Study. In Proceedings of the International Conference on Dublin Core and Metadata Applications, 2017. doi:10.23106/dcmi.952137857.
29. Suleman, H. (2018). The design and architecture of digital libraries. In G. G. Chowdhury & Foo Schubert (Eds.), *Digital Libraries and Information Access* (Chapter 2). Cambridge University Press.
30. Anonymous authors. (n.d.). Digital Libraries: Analysis of Delos Reference Model and 5S Theory. *Journal of Information Science Theory and Practice*.
31. Novitsky, A. V., & Reznichenko, V. A. (2020). Model of information object for digital library and its verification. *Problems in Programming*, 2–3, 31–38.
32. Novytskyi, O. V. (n.d.). Integrated open source metadata aggregation environment to support scientific research. *Problems in Programming*.
33. Tamaro, A. M., Castelli, D., Candela, L., Athanasopoulos, G., El Raheb, K., Innocenti, P., Ioannidis, Y., Katifori, A., Nika, A., Vullo, G., Ross, S., & others. (2011). *Digital Library Reference Model Concepts and Relations*. DL-org.
34. Aparac-Jelušić, T. (2017). Digital Libraries for Cultural Heritage: Development, Outcomes, and Challenges from European Perspectives. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 4, i–175. Springer. doi:10.2200/s00775ed1v01y201704icr058.
35. Haslhofer, B., & Isaac, A. (2011). A Linked Data-based Semantic Interoperability Framework for Digital Libraries.
36. Patra, C. (2006). Interoperability in Semantic Digital Library. *Information Systems Research Journal*.

37. Digital Information and Library Services in ODDE — Towards a Collaborative Digital Library Model. In Handbook of Open, Distance and Digital Education. 2023. - doi:10.1007/978-981-19-0351-9\_45-1.
38. Khoeini, S., Noruzi, A., Naghshineh, N., & Sheikhshoei, F. (2024). Developing a Model of Digital Transformation of University Libraries Based on Meta-Synthesis. *The Electronic Library*, 42(4), 681–699.
39. Khoeini, S., Noruzi, A., Naghshineh, N., & Sheikhshoei, F. (2024). Designing the Digital Transformation Model of Public University Libraries in Iran Based on Delphi Method. *Digital Library Perspectives*, 41(1), 45–73.
40. Novitsky, A. V., & Reznichenko, V. A. (2020). Model of Information Object for Digital Library and Its Verification. *Problems in Programming*, 2–3, 31–38.
41. Novytskyi, O. V. (2024). Integrated Open Source Metadata Aggregation Environment to Support Scientific Research. *Problems in Programming*.
42. Al-Natsheh, H. T., Martinet, L., Muhlenbach, F., Rico, F., & Zighed, D. A. (2018). Metadata Enrichment of Multi-Disciplinary Digital Library: A Semantic-based Approach. arXiv preprint arXiv:1806.08202.
43. Indrák, M., & Pokorná, L. (2021). Analysis of Digital Transformation of Services in a Research Library. *Global Knowledge, Memory and Communication*, 70(1/2), 154–172

## **ДОДАТКИ**

## Додаток А

### Фрагменти кодів створення інструментарію

#### index.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="utf-8" />
<title>Digital Library Model</title>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<style>

/* ----- змінні ----- */
:root{
  --bg:#f5f7fa;
  --card:#ffffff;
  --text:#1f2937;
  --muted:#6b7280;
  --accent:#3b82f6;
  --radius:10px;
  --shadow:0 2px 6px rgba(0,0,0,.06);
  --font:'Inter',system-ui,-apple-system,Segoe
UI,Roboto,Helvetica,Arial;
}

/* ----- база ----- */
*{box-sizing:border-box;margin:0;font-family:var(--font)}
body{background:var(--bg);color:var(--text);display:flex;min-
height:100vh;line-height:1.4}
a{color:inherit;text-decoration:none}

/* ----- ліве меню ----- */
aside{width:260px;background:var(--card);padding:28px 20px;box-
shadow:var(--shadow)}
aside h2{font-size:1.25rem;margin-bottom:1.2rem}
.tree{list-style:none;padding-left:0;font-size:.95rem}
.tree ul{list-style:none;padding-left:22px;margin:4px 0}
.tree li{margin:6px 0;display:flex;align-
items:center;gap:8px;cursor:pointer}
.tree li:hover{color:var(--accent)}
.tree .badge{background:var(--accent);color:#fff;font-
size:.7rem;padding:2px 7px;border-radius:999px;margin-left:auto}

/* ----- основна зона ----- */
main{flex:1;padding:32px 40px 40px}
main h1{font-size:1.8rem;margin-bottom:1.5rem}
```

```

.grid{display:grid;grid-template-columns:repeat(auto-
fill,minmax(170px,1fr));gap:20px}
.card{background:var(--card);border-radius:var(--radius);box-
shadow:var(--shadow);padding:18px 16px;display:flex;align-
items:center;justify-content:space-between;transition:.2s}
.card:hover{transform:translateY(-2px);box-shadow:0 4px 12px
rgba(0,0,0,.08)}
.card span:first-child{font-size:1.05rem}
.card .badge{background:var(--muted);color:#fff;font-
size:.7rem;padding:3px 8px;border-radius:999px}

/* ----- мобільна адаптація ----- */
@media(max-width:700px){
  body{flex-direction:column}
  aside{width:100%;box-shadow:none;border-bottom:1px solid #e5e7eb}
  main{padding:24px 20px}
}
</style>
</head>
<body>

<!-- ліве меню -->
<aside>
  <h2>Your digital library</h2>
  <ul class="tree">
    <li>📁 fulltext_sgcarch<span class="badge">1</span></li>
    <li>📁 NDLTD DL<span class="badge">5</span></li>
    <li>📁 Stream_Model<span class="badge">5</span></li>
    <li>📁 metadata_search<span class="badge">1</span></li>
    <li>📁 Struct_Model<span class="badge">2</span></li>
    <li>📁 browsing<span class="badge">2</span></li>
    <li>📁 Space_Model<span class="badge">1</span></li>
    <li>📁 submission<span class="badge">2</span></li>
    <li>📁 Scenario_Model<span class="badge">1</span></li>
    <li>📁 training<span class="badge">1</span></li>
    <li>📁 Society_Model<span class="badge">2</span></li>

    <li style="margin-top:20px;font-weight:600">Digital library
model</li>
  <ul>
    <li>Serviceset<span class="badge">1</span></li>
    <li>Services<span class="badge">1</span></li>
    <li>Digital_Library<span class="badge">5</span></li>
  </ul>
</ul>
</aside>

<!-- основна зона -->

```

```

<main>
  <h1>Digital Library Model</h1>
  <div class="grid">
    <div class="card"><span>fulltext_sgcarch</span><span
class="badge">1</span></div>
    <div class="card"><span>NDLTD DL</span><span
class="badge">5</span></div>
    <div class="card"><span>Stream_Model</span><span
class="badge">5</span></div>
    <div class="card"><span>metadata_search</span><span
class="badge">1</span></div>
    <div class="card"><span>Struct_Model</span><span
class="badge">2</span></div>
    <div class="card"><span>browsing</span><span
class="badge">2</span></div>
    <div class="card"><span>Space_Model</span><span
class="badge">1</span></div>
    <div class="card"><span>submission</span><span
class="badge">2</span></div>
    <div class="card"><span>Scenario_Model</span><span
class="badge">1</span></div>
    <div class="card"><span>training</span><span
class="badge">1</span></div>
    <div class="card"><span>Society_Model</span><span
class="badge">2</span></div>
    <div class="card"><span>Serviceset</span><span
class="badge">1</span></div>
    <div class="card"><span>Services</span><span
class="badge">1</span></div>
    <div class="card"><span>Digital_Library</span><span
class="badge">5</span></div>
  </div>
</main>

</body>
</html>

```