

## ВИКОРИСТАННЯ XML ДЛЯ ПІДГОТОВКИ СПЕЦІАЛІЗОВАНОЇ ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

У даній статті розглядаються питання створення спеціалізованої документації для програмного забезпечення. Розроблені інструментальні засоби базуються на широко поширених технологіях XML і DocBook.

The paper discusses the creation of specialized documentation for the software. The developed tools are based on widely accepted technologies, XML and DocBook.

### Вступ

Досить часто програмна документація має специфічні особливості. Наприклад, вистовуються спеціальні для даної предметної області шаблони опису, або є суттєві обмеження технологічного процесу підготовки документації.

У даній статті розглядаються різні системи документування програмного забезпечення і робиться вибір відповідного інструментарію. Також робиться вивід про необхідність доопрацювання готової системи документування і описуються розроблені для конкретного випадку технологія і програмні засоби, що дозволяють вирішити поставлену задачу більш економічним чином.

### Постановка задачі

У досить великий за обсягом комплекс програмного забезпечення входить підсистема «швидких звітів», побудованих на основі SQL-запитів. Підсистема повинна мати програмну документацію. Кількість звітів – досить велика, декілька сотень, готувати і, тим більше, супроводжувати опис звітів традиційними методами достатньо складно і дорого.

Треба вибрати найбільш придатний інструментарій для документування програмної підсистеми і, у разі потреби, внести необхідні доробки у технологію та програмні засоби.

### Принцип єдиного джерела

Складання програмної документації - дуже

важлива складова розробки програмного продукту [1 - 2]. Стандарт ISO/IEC 12207, що визначає процеси життєвого циклу програмного забезпечення, навіть передбачає спеціальний процес, присвячений вказаному питанню. При цьому на кожен програмний продукт повинна розроблятися документація двох типів: для користувачів різних груп і для розробників. Відсутність документації будь-якого типу для конкретного програмного продукту є недопустимою.

Модульний підхід укорінився в програмуванні багато років тому. Ніхто не набирає один і той же програмний код багато разів, а зловживання командами текстового редактора «Сору» і «Paste» програмісти, поза сумнівом, визнають грубою роботою. Якщо програма в різних ситуаціях повинна виконувати одні і ті ж дії, їх оформляють у вигляді функцій або процедур. Якщо дії однакові загалом, але розрізняються залежно від конкретної ситуації, то програміст передбачає можливість передати функції або процедурі певні аргументи.

У технічній документації повтор – явище не менш рідкісне, ніж у програмі. Доводиться описувати схожі функції програм і систем, схожі екранні форми, схожі дії користувачів. Описи одних і тих же об'єктів доводиться повністю або частково дублювати в різних документах. Оскільки матеріал, з якого складається технічна документація, – текст на природній мові, застосувати до нього модульний підхід виявляється складнішим, ніж до програмного коду. У тексті документації досить особливостей, обумовлених лексичними і граматичними виклю-

ченнями, традиціями, що склалися, міркуваннями стилістики і естетики. Багато розробників документації роблять те, що було б ганьбою для будь-якого програміста: «копипейстят» власний текст і при необхідності уручну правлять його, пристосовувавши до чергового контексту.

Принцип єдиного джерела і технології, що реалізують його, дозволяють застосувати модульний принцип до документування [3]. Завдяки цьому принципу ми можемо не дивлячись на всю своєрідність тексту «вносити за дужки» однакове, параметризувати майже однакове і формувати з цих блоків документи в тому вигляді, в якому їх вважає за краще отримувати їх аудиторія.

Застосування технології єдиного джерела вимагає підготовки. Спочатку проектується структура єдиного джерела, розробляються шаблони і стилі оформлення, встановлюється і настроюється інструментарій для формування документів. Ця стадія віднімає час на початку проекту і пред'являє досить високі вимоги до кваліфікації розробника технічної документації. Але потім починається рутинна робота по написанню тексту і його завантаженню в єдине джерело. У будь-який момент на основі введеного тексту можна сформувати документи, з більшою чи меншою мірою готовності. На цій стадії ми отримуємо віддачу від зроблених вкладень. Розділ, малюнок, таблиця, абзац, будь-який інший фрагмент, який повинен з'являтися в декількох місцях, при необхідності достатньо виправити одноразово в єдиному джерелі. Чудово, що це стосується не тільки тексту, але і структурних рішень. Одноразового внесення змін до шаблону, припустимо, керівництва користувача, достатньо, щоб всі документи цього типу після чергової автоматичної обробки були змінені потрібним чином.

Принцип єдиного джерела в документуванні, як і модульний принцип в програмуванні, допомагає організувати роботу колективу співавторів, розподіливши між ними більш-менш ізольовані підзадачі. Таким чином, єдине джерело – це не тільки технічне, але ще і організаційне рішення. Взаємодія між учасниками розробки технічної документації теж може бути автоматизованою. Крупні інтегровані середовища для автоматизації документування, такі, як AUTHORIT, SiberSafe, ROBOHELP надають

для цього різні засоби: управління правами доступу, контроль версій, планування робіт і тому подібне

### Існуючі технології підготовки програмної документації

Залежно від цілей і об'ємів документації, що готується, розробниками створюються різні технологічні ланцюжки з використанням наступних найбільш популярних програмних продуктів [4 - 6]:

- *Microsoft Word – PDF*

Текст пишеться з використанням текстового редактора Microsoft Word і потім конвертується у формат PDF. Технологічний ланцюжок дозволяє готувати паперові публікації, а також комплекти електронної документації у форматі PDF.

Використовується в тих випадках, коли вимоги до видавничої якості документації невисокі. Перевагами є легкість впровадження і можливість рецензування і редагування матеріалів всіма, хто має до них доступ. Недоліком є невисока якість верстки.

- *Adobe FrameMaker – PDF*

Текст верстається з використанням програмного комплексу Adobe FrameMaker і потім конвертується у формат PDF. Технологічний ланцюжок дозволяє готувати паперові публікації, а також комплекти електронної документації у форматі PDF. Можливо також перетворення в інші формати (RTF, HTML, різні застосування XML і т. д.).

Використовується для підготовки публікацій професійної видавничої якості. До недоліків відноситься необхідність витрат на впровадження і неможливість редагування матеріалів «на льоту».

- *Технології підготовки контекстної довідки*

Технологічні ланцюжки цього типу використовуються для створення контекстної довідки в одному з поширених форматів: WinHelp, HTML Help і так далі Ланцюжок може включати спеціалізовану програмну систему (ForeHelp, eHelp, RoboHelp) або

обходитися без неї.

Технології підготовки контекстної довідки вимагають витрат на впровадження і узгодження дій між технічним письменником і командою розробників.

- **Технологія єдиного джерела на основі DocBook**

Текст технічної документації спочатку пишеться на мові розмітки DocBook і використовується як єдине джерело. Два паралельні технологічні ланцюжки дозволяють отримати файл у форматі PDF (який надалі розповсюджується в електронному вигляді або служить основою для паперової публікації) або файл у форматі HTML Help.

Технологія єдиного джерела на основі DocBook є оптимальним рішенням в тих випадках, коли необхідно скласти і підтримувати технічну документацію як у формі паперових (або електронних) документів, так і у формі системи контекстної довідки. При цьому всі зміни вносяться до єдиного джерела, вихідні ж файли будь-якого формату при необхідності генеруються з єдиного джерела. Таким чином, їх зміст завжди відповідає актуальному стану початкового тексту, написаного на DocBook.

Технологія потребує значних витрат зусиль на впровадження.

- **Технологія DITA**

Основний постулат технології DITA полягає в тому, що текст технічної документації складається з фрагментів (так званих топиків), що типізуються, причому кожному типу відповідає певний план опису. При цьому технологія DITA дозволяє авторові декларувати нові типи топиків для потреб конкретного проекту і задавати для кожного типу план опису.

У комплекті інструментарію DITA вже є готові стилі, що реалізують усі найбільш важливі функції формування документів приблизно на тому ж рівні, на якому це роблять стилі DocBook. Технологія DITA дозволяє, з одного боку, створювати нові типи топиків, а з іншою, використовувати для їх обробки вже наявні стилі. Це досягається завдяки прийому, який в об'єктно-орієнтованому програмуванні відомий як успадкування.

Технологія DITA є більш складною у порівнянні з DocBook і потребує ще більш витрат на впровадження.

Зі всіх розглянутих технологій найбільш прийнятною для початкового впровадження принципу єдиного джерела здається технологія DocBook. Вона є у достатній мірі гнучкою та потужною і водночас не такою складною, як DITA. До того ж DocBook є відкритою системою (Open Source проект).

### **Технологія DocBook/XML**

Основою технологічної платформи DocBook/XML служить однойменна проблемно-орієнтована мова розмітки [7]. Вона призначена для запису тексту технічної документації на програми, алгоритмічні мови, комп'ютерне устаткування і інші рішення в області інформаційних технологій, чим принципово відрізняється від більшості форматів зберігання текстових даних (але не XML-мов!). У мові DocBook/XML передбачені засоби опису фрагментів, властивих технічній документації, наприклад, спеціальними елементами належить виділяти назви елементів інтерфейсу, позначення клавіш, назви змінних, терміни, різні врізання (зауваження, підказки, попередження), лістинги, описи виконуваних користувачем процедур. Розмітка, що задається мовою DocBook/XML, носить переважно функціональний характер: автор указує роль, яку той або інший фрагмент відіграє в тексті, а не спосіб його зовнішнього оформлення. Такий підхід скоує автора, зате дозволяє добитися явної незалежності змісту і оформлення вихідного документа та уніфікувати деякі важливі якості стилю викладу при роботі декількох авторів в одному проекті.

Для набору тексту вхідного документа можуть застосовуватися різні програми, від звичайного «Блокнота» до розвинених XML-редакторів. Оскільки вимоги до формату вхідного документа визначаються специфікаціями використовуваних мов розмітки, вибір редактора перестає бути важливим питанням, яке необхідно вирішити на рівні проекту. Кожен автор може працювати в тому редакторі, який йому зручний.

По відношенню до завдання набору тексту у форматі DocBook/XML редактори, що існують сьогодні, можна розділити на наступні групи:

- звичайні текстові редактори;

- XML-редактори з інтерфейсом текстового процесора;
- XML-редактори, що спрощують набір розмітки.

Вимоги до оформлення вихідних документів в різних проектах істотно різні, тому спроби випустити вичерпний набір XSLT-стилів приречені на невдачу. Створювати XSLT-стили наново в кожному проекті теж не можна, тому що терміни і вартість цієї роботи вийдуть неприйнятними. На практиці зазвичай застосовується вільно поширюваний комплект стандартних стилів DocBook XSL [7]. Він підтримує основні елементи макетів і цільові формати, а також добре пристосований до доопрацювання для потреб конкретного проекту. Завдяки архітектурі XSLT адаптація стандартних стилів не вимагає модифікації їх коду. Нові правила, що розташовуються в окремих файлах, доповнюють або заміщають стандартні. Виконані доопрацювання не прив'язані до конкретної копії стандартних стилів, отже:

- доопрацювання не конфліктують один з одним, і ними легко обмінюватися;
- можливість оновлення версій DocBook XSL обмежена тільки їх зворотною сумісністю.

DocBook успішно використовується при розробці і підтримці технічної документації, особливо коли мова йде про об'єми в тисячі і десятки тисяч сторінок. Технологія дозволяє вирішити ряд типових завдань, що виникають при розробці технічної документації, таких як:

- розробка і оформлення документації в суворій відповідності з галузевими і державними стандартами (ЕСКД, ЕСПД, КСАС);
- автоматизація процедур нормоконтролю документації, що готується;
- підготовка документації в різних електронних форматах, призначених для друку і перегляду на екрані монітора: PDF, HTML, HTML Help, Java Help, Unix Man Pages;
- підтримка багатокористувачевого режиму підготовки документації.

Технологія DocBook чудово підходить підприємствам, чия продукція супроводиться технічною документацією, підготовленою та оформленою відповідно до вимог ДСТ 2, 19 і 34

серії (ЕСКД, ЕСПД, КСАС).

Проте, навіть така розвинена технологія, як DocBook, потребує деякого доопрацювання при конкретному її застосуванні.

### **Підготовка спеціалізованої програмної документації**

Великий за обсягом програмний комплекс «Лабораторні інформаційні системи» призначений для ведення інформаційних систем для обслуговування лікарень і медичних лабораторій.

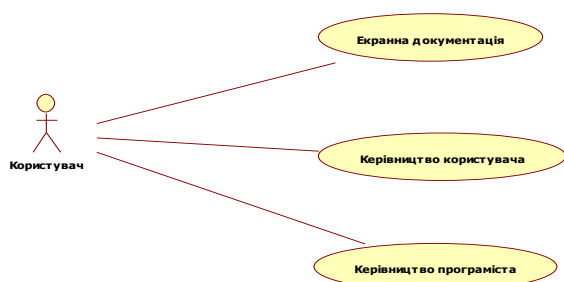
Однією з підсистем цього комплексу є підсистема «Швидкі звіти» SQL, яка служить для видачі даних про таблиці-довідники системи на підставі SQL запитів.

Підсистема «Швидкі звіти» SQL повинна мати повноцінну програмну документацію, яка включає розділи керівництва користувача і керівництва програміста (див. рис. 1).

До складу програмної документації також входить екранна допомога користувачеві звітів.

Підготовка програмної документації для підсистеми «Швидкі звіти» SQL має ряд важливих особливостей:

- SQL запити можуть достатньо часто змінюватися під час експлуатації системи, разом з цим також змінюється зовнішній вид звітів, і, отже, повинна змінюватися документація;
- програмна документація повинна підтримуватися у всіх вказаних вище видах – для кінцевого користувача, для програміста, екранна допомога;
- документація для користувача має бути представлена в декількох форматах – у форматі набору HTML-сторінок і форматі PDF-файлу ;
- програмна документація у вигляді екранної допомоги реалізується за допомогою спеціальної програмної системи і тому вихідна інформація для неї має бути представлена у вигляді XML-файлів спеціального формату;
- під час експлуатації системи можуть з'являтися нові звіти SQL і видалятися існуючі.



**Рис. 1. Діаграма використання документації підсистеми**

Всі ці чинники приводять до того, що система підготовки програмної документації, а особливо – система супроводу програмної документації мають бути дуже динамічними і повинні дозволяти швидку підготовку різних форматів документації з одного джерела.

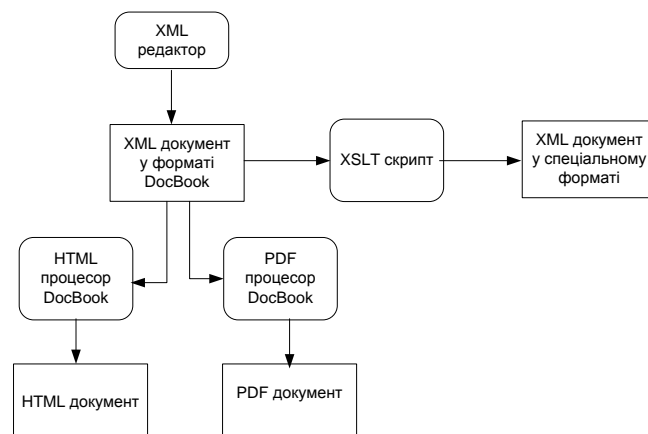
Як основна система документування, на підставі проведеного аналізу, вибрана система документування DocBook (реалізація для Windows).

Перше рішення було таким: підготувати вихідну документацію у форматі DocBook і засобами DocBook отримати результуючу документацію у форматі HTML (у вигляді ієрархічної структури сторінок) і у форматі PDF.

Для отримання необхідної документації екранної допомоги у вигляді XML-файлу, структура якого визначена зовнішньою системою підготовки екранної допомоги, було вирішено розробити скрипт конвертації засобами XSLT (мова перетворення XML-документів). Схема підготовки документації для підсистеми «Швидкі звіти» SQL придбала вигляд, зображений на рис. 2.

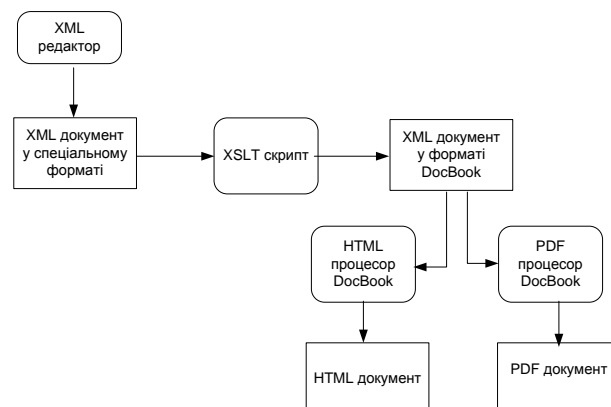
Проте, підготовка описів у форматі DocBook для великого числа звітів виявилася вельми трудомісткою (через гнучкість, об'ємність і складність цього формату), і тоді було вирішено готувати початкові описи за значно простішим шаблоном «XML в спеціальному форматі» (для системи допомоги). Модифікована схема підготовки документації приведена на рис. 3.

Така невелика зміна напряму потоку даних в робочому процесі привела зрештою до економії багатьох годин роботи операторів (програмістів) по підготовці вихідних описів звітів.



**Рис. 2. Схема підготовки документації (варіант 1)**

Такий процес підготовки документації є еквівалентним добре відомому в програмуванні процесу компіляції і збірки великого програмного продукту.



**Рис. 3. Схема підготовки документації (варіант 2)**

## Висновки

В процесі дослідження розглянуто різні системи документування програмного за безпечення та обґрунтовано вибір технології DocBook. Наведений приклад доопрацювання готової системи документування в конкретному випадку завдання підготовки спеціалізованої програмної документації (системи видачі програмних звітів) вирішене більш економічним чином.

**Перелік посилань**

1. Липаев В.В. Документирование и управление конфигурацией программных средств / В.В. Липаев. — М.: СИНТЕГ, 1998. — 220 с.
2. Липаев В.В. Качество программных средств / В.В. Липаев. — М: Янус-К, 2002. 400 с.
3. Технология DocBook и принцип единого источника [Электронный ресурс]. Режим доступа к ресурсу: <http://glossterm.org/index.php>
4. Белайчук А. Кратчайший путь к DocBook [Электронный ресурс]. Режим доступа к ресурсу: <http://docbook.ru/doc/sw/>
5. Технический писатель. Базовые компетенции специалиста [Электронный ресурс]. Режим доступа к ресурсу: <http://philosoft.ru/twskills.zhtml>
6. Технология DITA. Обзор возможностей и основные преимущества [Электронный ресурс]. Режим доступа к ресурсу: <http://www.philosoft.ru/dita-intro.zhtml>
7. Н.Уолш. DocBook: The Definitive Guide [Электронный ресурс]. Режим доступа к ресурсу: <http://www.docbook.org/tdg/en/html/docbook.html>