УДК 621.391

© П. Михайленич[7], 2015

ЗАСТОСУВАННЯ СТРУКТУРИЗАЦІЇ ТЕКСТОВИХ ДОКУМЕНТІВ ДЛЯ ЗБЕРІГАННЯ СЛУЖБОВОЇ ІНФОРМАЦІЇ В ІНФОКОМУНІКАЦІЙНИХ СИСТЕМАХ

Стаття присвячена аналізу проблеми збереження та ефективної обробки текстової інформації в інфокомунікаційних системах. В ній показано роль і особливості збереження текстової інформації з використанням як примітивних так і розвинутих форматів. Для розв'язку задачі збереження текстової інформації в інфокомунікаційних системах запропоновано застосовувати розширювану мову розмітки документів.

Ключові слова: інфокомунікаційна система, обробка текстової інформації, формат, розширювальна мова.

The paper analyzes the problem of saving and efficient processing of text information in information communication systems. It shows the role and features of text information saving using both primitive and advanced file formats. Extensible markup language has been proposed for solving the problem of text information saving in information communication systems.

Keywords: information communication system, processing of text information, format, extensible language.

1. ВСТУП

В телекомунікаційних мережах обмін інформацією здійснюється з використанням різних протоколів та способів форматування. Інформація, яка передається протоколами зв'язку спочатку капсулюється в пакетах, а на приймальній стороні зберігається на жорсткому диску, або оперативній пам'яті комп'ютера. Відповідно, для забезпечення подальшої роботи з отриманою інформацією необхідно визначити спосіб та правила її обробки. В сучасних інфокомунікаційних системах для збереження та представлення інформації можуть використовуватися різні формати текстових документів. Перед адміністратором виникає проблема вибору та застосування певного методу форматування текстової інформації, який забезпечує високу ефективність функціонування інфокомунікаційної системи. З цією метою в роботі визначено перелік форматів, які можуть використовуватися для розв'язку задачі збереження текстової інформації та проведено дослідження їхніх можливостей.

7 Національний університет “Львівська політехніка”

2. ПРОБЛЕМИ ЗБЕРЕЖЕННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ

2.1 Примітивні формати тексту

На початку розвитку ери комп'ютерної техніки та телекомунікаційних технологій основним форматом збереження інформації був відкритий текст (plain text), тобто такий спосіб її представлення при якому не застосовувалися певні правила його форматування. Для кодування інформації в документах відкритого тексту першою використовувалася символьна таблиця ASCII, яка внаслідок забезпечення недостатньої кількості символів була замінена стандартом кодування символів Unicode та таблицями символів UTF-8, UTF-16. Особливістю застосування відкритого тексту є його незалежність від різних стандартів форматування, які визначають власні правила обробки інформації та вимагають застосування лише відповідних їм програмних продуктів. Це забезпечує можливість вільного обміну інформацією між різними системами. Текстові документи у відкритому форматі можуть оброблятися практично усіма текстовими редакторами та деякими спеціалізованими програмами. Він широко застосовується для створення вихідного коду під час розробки програмного забезпечення, налаштування роботи серверів інформаційних систем. Проте, формат відкритого тексту не забезпечує високу ефективність обробки текстової інформації, і внаслідок цього не може застосовуватися в інфокомунікаційних системах як основний метод її збереження. Це зумовлює необхідність застосування більш складної структури документу та правил обробки інформації.

В процесі розвитку інформаційних технологій, формат відкритого тексту став поступово витіснятися новими, які базуються на застосуванні більш складних правил форматування (richtext). Хоча формат відкритого тексту забезпечує простоту та доступність, його складно застосувати для обробки великих об'ємів текстової інформації, формування структури документів, що створює незручності у роботі з ним. Це зумовило необхідність розробки нових методів збереження інформації у текстових документах та появи складних стандартів форматування. Для опису процесу форматування текстової інформації почали вживати термін розмітка з метою позначення факту створення його структури і поділу на окремі частини з різним призначенням.

Одним із найпростіших методів форматування текстових документів, які використовуються в інфокомунікаційних системах, є їхній поділ на окремі частини за допомогою розділових символів, наприклад символу табуляції, пробілу, коми, двокрапки, тощо. Такий метод називають форматуванням розділовими символами - DSV (Delimiter Separated Value). Згідно вимог форматування DSV, поділ

тексту реалізовується будь-якими символами, навіть тими, які зустрічаються в ньому [2]. Це призводить до появи помилок форматування. Прикладом реалізації форматування типу DSV є використання в якості розділового символу коми — CSV (Comma Separated Value). Формат CSV має достатньо простий набір правил форматування тексту і може використовуватися в якості нестандартизованої методики обробки невеликих за обсягом файлів. Принципи впорядкування тексту CSV є подібним до формату відкритого тексту, а основною його відмінністю є те, що групи символів логічно відокремлюються одна від одної спеціальним розділовим символом — комою. Весь текстовий файл в форматі CSV складається із послідовності рядків (записів інформації), які відокремлюються один від одного символом переносу. Рядок в свою чергу містить групи символів (інформаційні поля) розділені між собою символом коми. Таке форматування тексту дозволяє зберігати інформацію у вигляді умовної таблиці. Стовпці таблиці є інформаційними полями, а рядки — однотипними записами. Для отримання вмісту комірки таблиці необхідно знайти рядок та скопіювати вміст поля, що відповідає шуканому стовпцю. Приклад збереження інформації про працівників, за допомогою формату CSV, в інфокомунікаційній системі:

рік, місяць, день, країна, місто, прізвище, ім'я

1990, 05, 20, Україна, Львів, Коваль, Андрій

1987, 10, 04, Україна, Київ, Михайлик,Іван

Для формату CSV є характерною проблема відсутності засобів створення структури текстового документу, внаслідок чого ефективність виконання таких операцій як запис, пошук, зміна типу чи призначення інформації є низькою.

2.2 Форматування текстових документів розміткою

Для подолання проблем, які виникали в процесі використання простих типів форматування, покращення методів збереження та обміну інформацією в інфокомунікаційних системах, було розроблено стандарт узагальненої мови розмітки документів SGML (Standard Generalized Markup Language). Стандартом визначено два основні положення. Розмітка текстового документу повинна базуватися на визначені його структури, а не описі способу його обробки. Це положення забезпечує формування стандартизованої методики обробки текстових документів різними прикладними програмами, а саме: їхня обробка повинна полягати у аналізі типової структури, а не

у застосуванні правил, які додаються до них. Наступне положення полягає в тому, що розмітка текстового документу повинна відповідати строгим правилам. Виконання цього положення забезпечує можливість обробки тестових документів програмами, які підтримують заданий формат та здійснювати обмін між різними інформаційними системами без застосування додаткових правил форматування.

В стандарті SGML вперше запропоновано використання для опису формату текстового документу поняття “тип документа” DT (DocumentType) та “визначення типу документа” DTD (DocumentTypeDefinition) [3, 4]. Поняття “тип документа” задає його формат, або структуру. “Визначення типу документа” є описом його структури, тобто переліку всіх його елементів з атрибутами та поведінкою, які можуть зустрічатися в ньому. Таким чином, формат текстового документу задається шляхом опису його типу DTD, а корисна інформація зберігається за допомогою елементів. Це забезпечує універсальний підхід щодо створення текстових документів різного призначення на основі єдиного стандарту.

Структурними частинами документу SGML, в межах яких здійснюється однакове форматування тексту є елементи. Кожний елемент має власну назву, тобто узагальнений ідентифікатор (generic identifier), за допомогою якого здійснюється пошук інформації. Межі дії елементів позначаються мітками (tag), які містять його назву та знаки <, > і /. Елемент відкривається за допомогою початкової мітки (start tag), що складається із символів < та >, і закривається такою самою міткою (end tag) з символом /. Використання ієрархічної деревоподібної структури, яка містить пару кореневий та інформаційний елементи, забезпечує ефективну обробку текстових документів. З цією метою розробляється програмне забезпечення для пошук необхідного елементу в ієрархічній деревоподібній структурі текстового документу та одержання інформації, яку він містить. Цей процес відбувається значно швидше, ніж у випадку пошуку інформації документа у форматі відкритого тексту, або DSV.

2.3 Розробка структури документу

Розглянемо застосування розмітки текстових документів для розв'язку задачі збереження інформації про встановлені з'єднання в інфокомунікаційній системі. З цією метою розроблено структуру текстового документу (рис. 1).

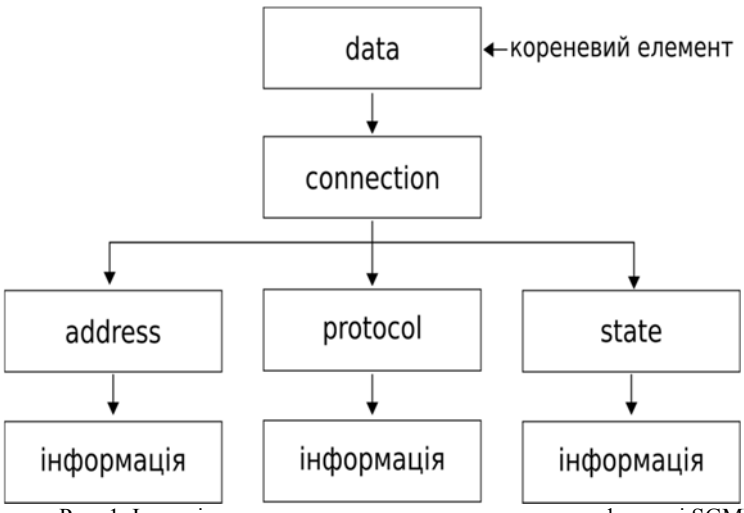

Рис. 1. Ієрархічна структура текстового документу в форматі SGML

Документ повинен містити кореневий елемент data, в межах якого дозволяється застосування одного елементу connection. Цей елемент в свою чергу зберігатиме інформацію про встановлені з'єднання. Призначення решти елементів є таким:

• address — використовується для збереження IPv4-адреси віддаленого вузла;

• protocol — містить інформацію про тип протоколу передавання інформації, наприклад TCP, HTTP, FTP, тощо;

• state — зберігає стан з'єднання, наприклад встановлене (open), або завершене (close).

Після розробки структури, необхідно описати “визначення типу документа” згідно стандарту SGML. Для заданої структури воно повинно бути таким:

```
<!DOCTYPE data>
[
<!ELEMENT connection (address, protocol, state)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT protocol (#PCDATA)>
<!ELEMENT state (#PCDATA)>
]>
```

В результаті інформація про встановлені з'єднання в інфокомунікаційній системі зберігатиметься у документі таким чином:

```
<data>
        <connection id=17>
                <address>172.16.2.14</address>
                <protocol>FTP</protocol>
                <state>open</state>
        </connection>
        <connection id=21>
                <address>192.168.11.3</address>
                <protocol>HTTP</protocol>
                <state>close</state>
        </connection>
        <connection id=32>
                <address>10.10.8.236</address>
                <protocol>HTTPS</protocol>
                <state>close</state>
        </connection>
</data>
```

Використання ієрархічної деревоподібної структури документу забезпечує ефективну обробку інформації про стан роботи інфокомунікаційної системи, наприклад: збереження інформації про встановлені з'єднання, аналіз типу протоколів, пошук IP-адрес, які використовувалися у запитах до системи, тощо.

3. ВИСНОВКИ

Перевагами узагальненої мови розмітки є те, що вона стандартизована, (має офіційну підтримку і розробку), забезпечує повний опис структури та вмісту документів у єдиному файлі, а створення ієрархічної структури документу забезпечує кращі можливості читання чи збереження тексту в інфокомунікаційних системах. Основним недоліком узагальненої мови розмітки є складність реалізації програмного забезпечення для форматування документів. Для формування та перегляду документів в форматі SGML необхідні програми, які можуть забезпечити одночасну обробку його структури, визначення типу документу та додаткових файлів стилів. Методика форматування текстової інформації може використовуватися для розв'язку різних задач в інфокомунікаційних системах, наприклад: ведення журналів подій в системі, організація баз даних чи перетворення в інші формати.

1. DeRose S.J. The SGML FAQ Book: Understanding the Foundation of HTML and XML. Kluwer Academic Publishers, 1997. - 250 p. 2. DuCharme B. SGML CD. Prentice Hall PTR, 1998 – 353 p. 3. Flynn P. Understanding Sgml and Xml Tools: Practical Programs for Handling Structured Text. Springer-Verlag New York Incorporated, 2012. - 464 p. 4. Herwijnen E. Practical Standard generalized markup language/-2nd ed. Kluwer Academic Publishers, 1994. - 288 p. 5. Powell G. Beginning XML Databases. Wiley Publishing, Inc., 2007. - 479 p. 6. Raymond E.S. The art of UNIX programming. Pearson Education, Inc., 2004. - 526 p. 7. http://www.is-thought.co.uk/sgml.htm. 8. http://www.w3schools.com