UNIWERSYTET GDAŃSKI – WYDZIAŁ ZARZĄDZANIA

Walenty Szczęsny
nr albumu: rr000000001

Monika A. Szczęsna-Woś
nr albumu: rr000000002

Dokumenty strukturalne: teoria i zastosowania

Praca magisterska na kierunku:
INFORMATYKA EKONOMICZNA
Promotor:
ks. prof. Jan Kowalski

Wejherowo 2001

Streszczenie

Aczkolwiek w ciągu ostatnich lat skład tekstu wspomagany komputerowo całkowicie wyeliminował stosowanie tradycyjnych technik drukarskich, to podobny proces w przypadku publikacji elektronicznych czyli publikacji, które w ogóle nie wykorzystują papieru, a nośnikem informacji staje się ekran komputera nie jest obserwowany.

Słowa kluczowe

SGML, XML, XSL, dokumenty elektroniczne, dokumenty strukturalne

Spis treści

Wprowadzenie

Aczkolwiek w ciągu ostatnich lat skład tekstu wspomagany komputerowo całkowicie wyeliminował stosowanie tradycyjnych technik drukarskich, to podobny proces w przypadku publikacji elektronicznych czyli publikacji, które w ogóle nie wykorzystują papieru, a nośnikem informacji staje się ekran komputera nie jest obserwowany.

Formatowanie wizualne, powstało z myślą o przygotowaniu publikacji do druku i dlatego nie może sprostać nowym potrzebom, które stwarza postęp techniki. Coraz większą rolę odgrywają dziś elektroniczne repozytoria, bazy danych, publikacje na CD-Romach oraz WWW. Wypływa też stąd ważny wniosek, że tworzenie dokumentów według paradygmatu WYSIWYG nie jest efektywne i stopniowo należy oczekiwać powstawania, wdrażania i rozpowszechniania się systemów opartych na paradygmacie *formatowania strukturalnego*.

W dokumencie formatowanym strukturalnie oznaczana jest struktura dokumentu a nie określany jego wygląd. Zwróćmy uwagę, że układ graficzny jest pochodną struktury, tj. nadajemy jednolity wygląd tytułom rozdziałów, śródtytułów, przypisów, jednakowo wyróżniamy wyliczenia itp. Układ graficzny jak już wspominano może ulegać zmianie (np. wraz z rozpowszechnianiem się nowych technologii wydawniczych) ale treść i struktura raczej nie, np. Biblia Gutenberga widziana z tej perspektywy nie zmieniła się wcale.

ROZDZIAŁ 1

Wprowadzenie do standardów SGML/XML

SGML [9] jest to *metajęzyk* służący do opisywania struktury i zawartości dokumentów (standard ISO 8879). Do podstawowych zalet takiego podejścia należy:

- jest to międzynarodowy standard dostępny na wielu platformach sprzętowo-systemowych;
- jest to język opisu *każdego* dokumentu, o praktycznie nieograniczonych możliwościach (*rozszerzalność*)
- umożliwia powtórne wykorzystywanie dokumentów, także w sposób inny od poprzedniego (np. tradycyjna książka i dokument multimedialny utworzony z tego samego dokumentu SGML-owego).

Standard służy jedynie do opisywania logicznej struktury dokumentów, nie determinuje ostatecznej formy prezentacji informacji, która może być docelowo przekształcana w najróżniejszy sposób. Dokument zakodowany z wykorzystaniem SGML-a może służyć jako postać wyjściowa do formatowania tej samej informacji w różny sposób i prezentacji z użyciem różnych mediów np. w formie drukowanej na papierze, w postaci hipertekstu albo tekstowej bazy danych. Pozwala to zminimalizować koszty, cały cykl wydawniczy dokonuje się na jednym dokumencie – pliku SGML-owym, a nie na wielu.

$$f(X|\Phi) = \sum_{i=1}^{K} p_k \phi(X|\theta_k) \tag{1.1}$$

albo:

$$t = \frac{b^{(1)} - b^{(2)}}{\sqrt{\frac{(n^{(1)}-1)^2}{n^{(1)}+n^{(2)}-2} sb^{2(1)} + \frac{(n^{(2)}-1)^2}{n^{(1)}+n^{(2)}-2} sb^{2(2)}} \sqrt{\frac{1}{n^{(1)}} + \frac{1}{n^{(2)}}}} \tag{1.2}$$

1.1. Elementy składowe systemu SGML/XML

Standard SGML/XML to specyfikacja techniczna metajęzyka. Zaś systemem SGML/XML nazywamy *zestaw narzędzi i środków niezbędnych do tworzenia, składowania i obróbki dokumentów z wykorzystaniem tego standardu*. Typowy proces produkcji dokumentów w standardzie SGML/XML podzielony jest na kilka części. Najważniejszymi elementami tego procesu są [15, s. 45–47]:

- Klasyfikacja tworzonych dokumentów w grupy i wynikający z tego dobór definicji typu dokumentu[1] (por. punkt 1.2).

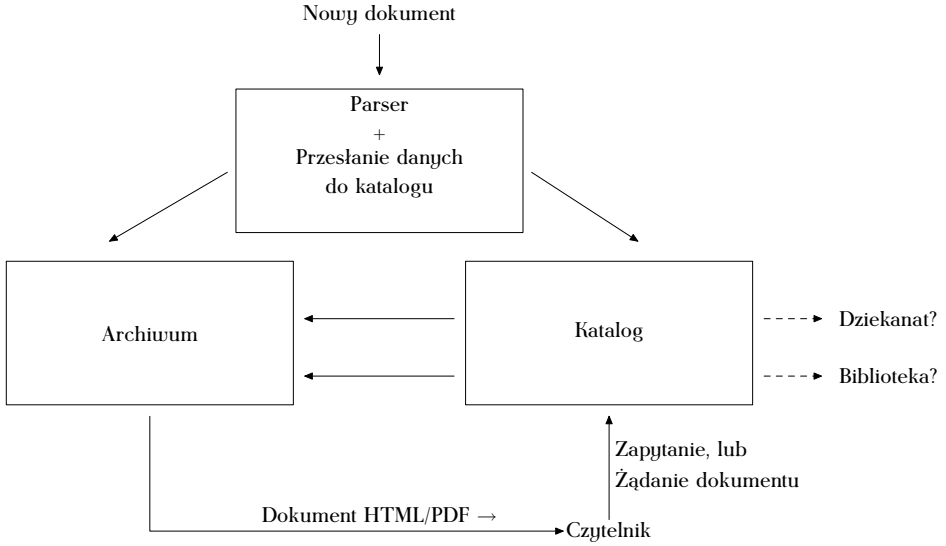
- Wybraniu odpowiednich narzędzi do tworzenia i modyfikowania dokumentów SGML/XML (edytory, konwertery, por. punkt 1.4, s. 8)[2].

- Sprawdzenie poprawności oznaczenia dokumentów (walidacja).

- Ustalenie metod składowania zbiorów oznakowanych dokumentów.

- Ustalenie sposobu prezentacji oznaczonych dokumentów i ich formatów wyjściowych – przygotowanie odpowiednich specyfikacji konwersji formatów i dobór właściwych do tego celu narzędzi (por. punkty 2.1 i 2.2).

Każdy z wyżej wymienionych problemów stanowi pewne zadanie do wykonania, które może być w dużym stopniu lub całkowicie zautomatyzowane dzięki zastosowaniu odpowiednich narzędzi, co jest jednym z celów i korzyści stosowania systemu opartego na standardzie SGML/XML.



Rysunek 1.1. Schemat archiwum

Źródło: Opracowanie własne

[1] Definicja typu dokumentu DTD to formalny opis pewnej klasy dokumentów, spis jej elementów składowych (znaczników i ich atrybutów) oraz zasad ich stosowania (hierarchii występowania czy możliwości skracania).

[2] Wybrany edytor winien w maksymalnym stopniu ułatwiać proces tworzenie dokumentów strukturalnych, różniący się od tego, jaki jest stosowany w narzędziach biurowych typu edytor Word.

1.2. Opracowanie typu dokumentu

Tworzenie DTD [15, 14, 19] powinno zacząć się od analizy posiadanych danych oraz potrzeb dotyczących dokumentów wynikowych i informacji. Rezultatem analizy dokumentów jest wyspecyfikowanie słownika takich elementów oraz, dla każdego typu dokumentu, zależności między tymi elementami. Z tego powodu stworzenie dobrego DTD jest zadaniem niełatwym. Ponadto DTD jest tylko częścią systemu SGML/XML; oprócz niego potrzebne są narzędzia do formatowania dokumentów, takie jak arkusze stylów DSSSL czy też XSL/XSLT.

Duże koszty implementacji przemawiają często za rozważeniem wykorzystania, czy też adaptacji w systemie SGML/XML udostępnionych publicznie gotowych DTD. W omawianym projekcie postąpiono podobnie, ewentualne opracowanie własnego typu dokumentu odkładając na później. Rozważano wstępnie wykorzystanie trzech publicznie dostępnych typów dokumentów: `etd` zdefiniowanego na potrzeby projektu *Electronic Theses and Dissertations* w Virginia Polytechnic Institute and State University, `TEI` opracowanego w ramach projektu *Text Encoding Iniciative* [1] oraz DocBook opracowany przez konsorcjum firm, głównie sektora IT, używających technologii SGML/XML do tworzenia dokumentacji do swoich produktów [19].

1.3. DocBook

DocBook DTD to typ dokumentu definiujący strukturę i zawartość zbioru znaczników SGML, służących do dokumentacji oprogramowania i różnego rodzaju dokumentacje techniczne [19, s. 123]. Posiada ona bardzo rozbudowaną hierarchię znaczników do budowy struktur książek, dokumentacji technicznej itp dokumentów. DTD dla tego typu dokumentu jest dostępne zarówno w standardzie SGML jak i XML. Dostępna jest także uproszczona wersja DTD o nazwie SimpleDocbook. Zaletą używania wersji uproszczonej jest dużo łatwiejsze posługiwanie się nią, z uwagi na znacznie mniejszą liczbę znaczników, por. tabela 1.1.

Typ dokumentu	Liczba elementów	Liczba encji
Docbook	357	1814
SimpleDocBook	93	234
TEI	brak danych	3
HTML	98	234

Tabela 1.1. Porównanie wielkości popularnych definicji typu dokumentu

Źródło: Obliczenia własne

DocBook jest używany obecnie w większości zarówno komercyjnych jak i niekomercyj-

nych projektach tworzenia dokumentacji komputerowej (np. dokumentacja dla takich projektów jak: LDP – *Linux Documentation Project*, *RedHat Gnome Desktop*, *Postgres SQL RDBMS*, *PHP3 Hypertext Preprocesor* czy dokumentacja do systemu FreeBSD).

1.4. Edytory sterowane składnią

Pierwszym elementem systemu SGML/XML jest edytor strukturalny, zwany też edytorem sterowanym składnią. W tradycyjnych edytorach działających według paradygmatu WYSIWYG autor wprowadzając tekst określa na bieżąco jego wygląd, mając zarówno w jednym jak i drugim przypadku dużą swobodę. Taki sposób pracy w przypadku masowego tworzenia dokumentów jest zupełnie nieefektywny. W szczególności dotyczy to pracy wielu autorów nad jednym lub wieloma dokumentami o zuniformizowanym formacie, czy też tworzenia dokumentów, o których z góry wiadomo, że będą prezentowane w różny sposób.

Pewnym rozwiązaniem opisywanych wyżej problemów jest wykorzystanie szablonów, w które, od jakiegoś czasu są wyposażane wszystkie popularne edytory biurowe. Posługując się szablonami, autor nie definiuje wyglądu samodzielnie tylko wypełnia treścią zdefiniowany z góry szablon dokumentu. Stanowiąc niewątpliwy krok do przodu takie podejście nie rozwiązuje jednak wszystkich problemów.

Komercyjne edytory SGML, są w chwili obecnej ciągle bardzo drogie, zaś oprogramowanie rozprowadzane na różnego rodzaju licencjach *Open Source* nie oferuje jeszcze wygody pracy, do której przyzwyczaili się użytkownicy edytorów biurowych.

1.5. Emacs i psgml

Konstatując brak tanich i funkcjonalnych edytorów do pracy z dokumentami SGML/XML należy wspomnieć o jedynym dostępnym w chwili obecnej, efektywnym, tanim (darmowy) i powszechnie dostępnym na wielu platformach systemowo-sprzętowych środowisku do tworzenia dokumentów strukturalnych jakim jest edytor Emacs z pakietem psgml. Środowisko to wspomaga autora poprzez: kolorowanie składni dokumentów SGML i aplikacji SGML, automatyczne uzupełnianie brakujących znaczników, automatyczną kontrolę jakie w danym kontekście można wstawiać znaczniki i atrybuty, wyświetlanie informacje odnośnie możliwych i domyślnych wartości. Wszystkie funkcje są dostępne za pomocą odpowiednie skrótów klawiszowych a także dostępne poprzez wybór wskaźnikiem myszy odpowiedniej pozycji z menu. Pakiet ten umożliwia również wywołanie zewnętrznego parsera SGML celem weryfikacji poprawności dokumentu.

ROZDZIAŁ 2

Narzędzia i standardy pokrewne

Systemy SGML, ze względu na mnogość funkcji jakie spełniają i ich kompleksowe podejście do oznakowywania i przetwarzania dokumentów tekstowych, są bardzo skomplikowane. Możemy wyróżnić dwa podejścia do budowy takich systemów. Z jednej strony, buduje się systemy zindywidualizowane, oparte o specyficzne narzędzia tworzone w takich językach, jak: C, C++, Perl czy Python. Edytory strukturalne, filtry do transformacji formatów czy parsery i biblioteki przydatne do konstrukcji dalszych narzędzi, tworzone są według potrzeb określonych, pojedynczych systemów.

Z drugiej strony, twórcy oprogramowania postanowili pójść krok dalej i połączyć te różne narzędzia w jedną całość. Tą całość miał stanowić DSSSL lub jego XML-owy odpowiednik – standard XSL. Ze względu na oferowane możliwości można twierdzić, że tworzenie i używanie narzędzi implementujących standard DSSSL/XSL, jest najwłaściwszym podejściem. Przemawiają za tym różne argumenty, ale najważniejszym z nich jest to, że mamy tu możliwość stworzenia niezależnego od platformy programowej i narzędziowej zbioru szablonów – przepisów jak przetwarzać dokumenty SGML/XML.

2.1. Przetwarzanie dokumentów SGML – standard DSSSL

DSSSL (*Document Style Semantics and Specification Language*) – to międzynarodowy standard ściśle związany ze standardem SGML. Standard ten, można podzielić na następujące części:

- język transformacji (*transformation language*). To definicja języka służącego do transformacji dokumentu oznaczonego znacznikami zgodnie z pewnym DTD na dokument oznaczony zgodnie z innym DTD.
- język stylu (*style language*) opisujący sposób formatowania dokumentów SGML.
- język zapytań (*query language*) służy do identyfikowania poszczególnych fragmentów dokumentu SGML.

Opisane główne części składowe standardu DSSSL dają obraz tego, jak wiele aspektów przetwarzania zostało zdefiniowanych i jak skomplikowany jest to problem. Jest to głównym powodem tego, że mimo upływu kilku lat od zdefiniowania standardu nie powstały ani komercyjne ani wolnodostępne aplikacje wspierające go w całości. Istnieją natomiast *nieliczne* narzędzia realizujące DSSSL w ograniczonym zakresie, głównie w części definiującej

język stylu, który odpowiada za opatrzenie dokumentu czysto strukturalnego w informacje formatujące. Daje to możliwość publikacji dokumentów SGML zarówno w postaci elektronicznej, hipertekstowej czy też drukowanej.

2.2. Przetwarzanie dokumentów XML – standard XSL

Tak jak XML jest *uproszczoną* wersją standardu SGML, tak XSL jest uproszczonym odpowiednikiem standardu DSSSL. W szczególności, wyróżnić można w tym standardzie następujące części składowe:

- język transformacji (XSLT) To definicja języka służącego do transformacji dokumentu.
- język zapytań (XPath) służy do identyfikowania poszczególnych fragmentów dokumentu.
- język stylu deefiniujący sposób formatowania dokumentów XML.

ROZDZIAŁ 3

Przegląd dostępnych narzędzi

W celu wykorzystania standardu SGML/XML do przetwarzania dokumentów, niezbędne jest zebranie odpowiedniego zestawu narzędzi. Narzędzi do przetwarzania dokumentów SGML/XML jest wiele. Są to zarówno całe systemy zintegrowane, jak i poszczególne programy, biblioteki czy skrypty wspomagające.

3.1. Narzędzia do przeglądania dokumentów SGML/XML

Do tej kategorii oprogramowania zaliczamy przeglądarki dokumentów SGML/XML oraz serwery sieciowe wspomagające standard SGML/XML, przy czym rozwiązań wspierających standard XML jest już w chwili obecnej dużo więcej i są dużo powszechniejsze.

Jeżeli chodzi o przeglądarki to zarówno Internet Explorer jak i Netscape umożliwiają bezpośrednie wyświetlenie dokumentów XML; ponieważ jednak nie wspierają w całości standardu XML, prowadzi to ciągle do wielu problemów[1].

3.2. Parsery SGML/XML

Program `nsgmls` (z pakietu `SP` Jamesa Clarka) jest doskonałym parserem dokumentów SGML, dostępnym publicznie. Parser `nsgmls` jest dostępny w postaci źródłowej oraz w postaci programów wykonywalnych przygotowanych na platformę MS Windows, Linux/Unix i inne. Oprócz analizy poprawności dokumentu parser ten umożliwia również konwersję danych do formatu ESIS, który wykorzystywany jest jako dane wejściowe przez wiele narzędzi do przetwarzania i formatowania dokumentów SGML. Dodatkowymi, bardzo przydatnymi elementami pakietu `SP` są: program `sgmlnorm` do normalizacji, program `sx` służący do konwersji dokumentu SGML na XML oraz biblioteki programistyczne, przydatne przy tworzeniu specjalistycznych aplikacji służących do przetwarzania dokumentów SGML.

W przypadku dokumentów XML publicznie dostępnych, parserów jest w chwili obec-

[1] Z innych mniej popularnych rozwiązań można wymienić takie aplikacje, jak: HyBrick SGML/XML Browser firmy Fujitsu Limited, Panorama Publisher firmy InterLeaf Inc, DynaText firmy Inso Corporation czy darmowy QWeb. W przypadku serwerów zwykle dokonują one transformacji „w locie" żądanych dokumentów na format HTML (rzadziej bezpośrednio wyświetlają dokumenty XML). Ta kategoria oprogramowania ma, z punktu widzenia projektu, znaczenie drugorzędne.

nej kilkadziesiąt. Do popularniejszych zaliczyć można Microsoft Java XML Parser firmy Microsoft, LT XML firmy Language Technology Group, Exapt oraz XP (James Clark)

3.3. Wykorzystanie języków skryptowych

3.4. Wykorzystanie szablonów XSL

Stosując wersję XML typu DocBook można wykorzystać szablony stylów przygotowane w standardzie XSL (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF) oraz hipertekstowej (HTML).

Podobnie jak w przypadku szablonów DSSSL, szablony stylów XSL są sparametryzowane i udokumentowane i dzięki temu łatwe w adaptacji. Do zamiany dokumentu XML na postać prezentacyjną można wykorzystać jeden z dostępnych publicznie procesorów XSLT (por. tabela 3.1).

Nazwa	Autor	Adres URL
`sablotron`	Ginger Alliance	`http://www.gingerall.com`
`Xt`	J. Clark	`http://www.jclark.com`
`4XSLT`	FourThought	`http://www.fourthought.com`
`Saxon`	Michael Kay	`http://users.iclway.co.uk/mhkay/saxon`
`Xalan`	Apache XML Project	`http://xml.apache.org`

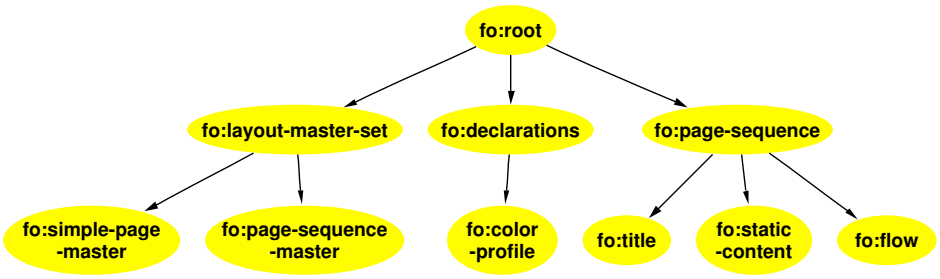
Tabela 3.1. Publicznie dostępne procesory XLST

Źródło: Opracowanie własne

Wykorzystując szablony, na podstawie dokumentu XML otrzymujemy bezpośrednio dokument HTML. W przypadku wersji drukowanych sprawa jest bardziej skomplikowana (por. rys. 3.1).

XSL:FO jest skomplikowanym językiem o dużych możliwościach, zawierającym ponad 50 różnych „obiektów formatujących", począwszy od najprostszych, takich jak prostokątne bloki tekstu poprzez wyliczenia, tabele i odsyłacze. Obiekty te można formatować wykorzystując przeszło 200 różnych właściwości (*properties*), takich jak: kroje, odmiany i wielkości pisma, odstępy, kolory itp. W tym dokumencie przedstawione jest absolutne miniumum informacji na temat standardu XSL:FO.

Cały dokument XSL:FO zawarty jest wewnątrz elementu `fo:root`. Element ten zawiera (w podanej niżej kolejności):



Rysunek 3.1. Ogólna struktura dokumenu XSL-FO

Źródło: Opracowanie własne

- dokładnie jeden element `fo:layout-master-set` zawierający szablony określające wygląd poszczególnych stron oraz sekwencji stron (te ostatnie są opcjonalne, ale typowo są definiowane);
- zero lub więcej elementów `fo:declarations`;
- jeden lub więcej elementów `fo:page-sequance` zawierających treść formatowanego dokumentu wraz z opisem jego sformatowania i podziału na strony.

Zakończenie

Możliwości, jakie stoją przed archiwum prac magisterskich opartych na XML-u, są ograniczone jedynie czasem, jaki należy poświęcić na pełną implementację systemu. Nie ma przeszkód technologicznych do stworzenia co najmniej równie doskonałego repozytorium, jak ma to miejsce w przypadku ETD. Jeżeli chcemy w pełni uczestniczyć w rozwoju nowej ery informacji, musimy szczególną uwagę przykładać do odpowiedniej klasyfikacji i archiwizacji danych. Sądzę, że język XML znacznie to upraszcza.

DODATEK A

Tytuł załącznika jeden

Treść załącznika jeden. Treść załącznika jeden. Treść załącznika jeden Treść załącznika jeden. Treść załącznika jeden. Treść załącznika jeden.

DODATEK B

Tytuł załącznika dwa

Treść załącznika dwa. Treść załącznika dwa. Treść załącznika dwa Treść załącznika dwa. Treść załącznika dwa. Treść załącznika dwa.

Bibliografia

[1] Burnard L., Sperberg-McQueen C. M.: *TEI Lite: An Introduction to Text Encoding for Interchange*, 1995. `http://www.uic.edu/orgs/tei/intros/teiu5.html`.

[2] Adobe Systems Inc: *Portable Document Format Reference Manual Version 1.2*, Addison-Wesley, 1996.

[3] Beebe N. H. F.: „Bibliography Prettyprinting and Syntax Checking", *TUGBoat* 14/4, 1995, s. 395–419.

[4] Braams J. L.: The Status of Babel, w: *Proceedings of the IXth European TeX Conference* (W. Dol red.), Arnhem 1995, s. 17–26.

[5] Bray T., Paoli J., Sperberg-McQueen C. M.: *Extensible Markup Language (XML) 1.0*, 1998. `http://www.w3.org/TR/1998/REC-xml-19980210`.

[6] Bryan M.: *SGML and HTML explained*, Addison-Wesley 1997.

[7] Clark J.: *XSL Transformations (XSLT) Version 1.0*, 1999. `http://www.w3.org/TR/xslt`.

[8] Flynn P.: *Understanding SGML and XML tools*, Kluwer Academic Publishers 1998.

[9] Goldfarb C.: *The SGML Handbook*, Oxford University Press.

[10] Goossens M., Rahtz S.: *The LaTeX Web Companion*, Addison-Wesley 1999.

[11] Grayson J.: *Python and Tkinter programming*, Manning Publications.

[12] Van Herwijnen E.: *Practical SGML*, Kluwer Academic Publishers 1990.

[13] Lamport L.: *LaTeX: a document preparation system*, Addison-Wesley 1994.

[14] Maler E.: *SGML exceptions and XML*, 1998. `http://www.arbortext.com/Think_Tank/XML_Resources/SGML_Exceptions_and_XML`.

[15] Maler E., El Andaloussi J.: *Developing SGML DTDs. From Text to Model to Markup*, Prentice Hall 1996.

[16] North S.: *XML dla każdego*, Helion 2000.

[17] Pepper S.: *Whirlwind Guide to SGML Tools and Vendors*. `http://www.infotek.no/sgmltool/index.htm`.

[18] Wall L., Christiansen T., Schwartz R.: *Programming Perl*, O'Reilly & Associates 1996.

[19] Walsh N., Muellner L.: *DocBook: The Definitive Guide*, O'Reilly & Associates October 1999. Dokument dostępny także w `http://www.docbook.org/`.

Spis tabel

Spis rysunków

Skorowidz

Oświadczenie

Ja, niżej podpisany(a) oświadczam, iż przedłożona praca dyplomowa została wykonana przeze mnie samodzielnie, nie narusza praw autorskich, interesów prawnych i materialnych innych osób.

.....	.....
data	podpis