

# Publikacje elektroniczne — przygotowanie tekstów

**Piotr Bolek**

[P.Bolek@oaza.org.pl](mailto:P.Bolek@oaza.org.pl)

**Adam Dawidziuk**

[A.Dawidziuk@oaza.org.pl](mailto:A.Dawidziuk@oaza.org.pl)

## 1. Wprowadzenie

Sieć internet staje się coraz bardziej popularna. Coraz więcej organizacji czy instytucji chce albo musi mieć własny serwis WWW. W serwisie tym mają być umieszczane teksty pisane specjalnie w tym celu oraz bardzo częsta także takie, które były już publikowane w inny, bardziej tradycyjny sposób: np. jako artykuł w czasopiśmie lub w formie książki. Coraz częściej publikowane są także dokumenty na płytach CD-ROM, zawierające np. zebrane roczniki czasopism, referaty z konferencji, encyklopedie, słowniki itp.

Elektroniczny sposób publikacji daje nowe, niezwykle możliwości. Przede wszystkim pozwala wyszukiwać interesujące czytelnika informacje, tworzyć różnego rodzaju zestawienia. Aby te nowe możliwości mogły być zrealizowane konieczne jest specjalne przygotowanie tekstów do publikacji elektronicznych. Nie wystarczą już tu techniki znane z DTP, ponieważ tam nastawiamy się na przygotowanie tekstu tylko w jeden i do tego ustalony raz na zawsze sposób — po wydrukowaniu tekstu w książce albo gazecie nic więcej nie można z nim zrobić.

Potrzebne są więc inne techniki pracy i „otwarte” formaty danych tekstowych umożliwiające wielokrotne używanie tych samych dokumentów w różnym kontekście i w różny sposób. W tym artykule chcemy opisać różne okoliczności związane z powstawaniem i redagowaniem tekstów przeznaczonych do publikacji elektronicznej oraz zwrócić uwagę na problemy jakie mogą wyniknąć w trakcie pracy. Chcemy także zaprezentować przykładowe rozwiązania, które można wykorzystać do efektywnego przetwarzania dokumentów elektronicznych.

Nie będziemy opisywać szczegółowych rozwiązań, omawiać wyczerpująco standardów, ani proponować konkretnych narzędzi. Tego typu sprawy są tak naprawdę drugorzędne i powinny być brane pod uwagę dopiero w konkretnych okolicznościach przez zespoły ludzi (specjalistów z różnych dziedzin) pracujących wspólnie nad konkretnym zastosowaniem tego o czym będziemy tutaj mówić. Chcemy dać ogólny obraz problematyki, zwracając uwagę na sprawy istotne i pokazując różnice w stosunku do tego co jest już znane, powszechnie dostępne, a czasem także niestety tylko modne.

## 2. Specyfika dokumentów elektronicznych

Dokumenty elektroniczne różnią się w sposób dosyć istotny od tradycyjnych publikacji papierowych. Materiały prezentowane w postaci elektronicznej mają bowiem najczęściej postać hipertekstu, czyli tekstu z dodanym „trzecim wymiarem”. Ten trzeci wymiar to możliwość wyróżniania fragmentów tekstu i definiowania go jako odnośnika do innej części tego samego dokumentu, innego dokumentu w tym samym serwisie, albo nawet do dokumentu znajdującego się na innym serwerze w dowolnie odległym zakątku świata.

Prezentacja informacji w postaci hipertekstu daje jeszcze inne możliwości. Pozwala mianowicie na tworzenie różnego rodzaju indeksów: tematycznych, chronologicznych, indeksów autorów, osób itp. Indeksy te mogą być realizowane „statycznie” albo „dynamicznie”. Indeks statyczny jest podobny do tradycyjnych indeksów w książkach, gdzie wszystkie hasła indeksu są wybrane przez autora (lub redaktora) i umieszczone w postaci uporządkowanej listy. Indeks dynamiczny (inaczej wyszukiwarka) jest generowany na żądanie — w momencie odwołania się do jakiegoś hasła przeszukiwane są bazy danych i generowane odnośniki do tekstów, które „pasują” do poszukiwanego przez czytelnika hasła. Oczywiście techniczna realizacja obu typów indeksów jest zupełnie inna, ale w obu przypadkach muszą być dostępne odpowiednio przygotowane teksty.

Podstawowym formatem, w jakim obecnie są publikowane dokumenty w WWW jest HTML. Format ten jest jednak stosunkowo prosty i ma dosyć poważne ograniczenia, więc często korzystne jest umieszczanie dokumentów także w innych formatach, np. PDF. Poza tym jak już wspomnieliśmy dokumenty elektroniczne nie muszą mieć wcale związku z WWW — HTML nie nadaje się np. absolutnie do publikacji technicznego czasopisma naukowego na płycie CD-ROM. Cały czas trwają także intensywne prace związane z opracowywaniem nowych standardów prezentacji dokumentów hipertekstowych. Dobrze byłoby więc aby teksty, które są przeznaczone do prezentacji w postaci elektronicznej, a ich aktualność nie jest ograniczona, mogły być w prosty sposób tłumaczone na różne, często obecnie jeszcze nie istniejące formaty.

Modny w ostatnich latach paradygmat tworzenia publikacji o wdzięcznej nazwie WYSIWYG (ang. *What You See Is What You Get* — czyli „widzisz to co otrzymasz”), wykorzystywany głównie w technologii DTP oraz w większości edytorów tekstów i wordprocesorów nie bardzo sprawdza się w przypadku pracy nad tekstami przeznaczonymi do publikacji elektronicznej. Przede wszystkim w przypadku publikacji elektronicznych ostateczny wygląd dokumentu nie może być zwykle określony ani przez autora, ani nawet redaktora publikacji. Nawet gdy jest to możliwe, to zawsze może okazać się, że zajdzie kiedyś konieczność prezentacji dokumentu w postaci, której autor czy redaktor nie przewidział. Tworzenie dokumentu w sposób „wizualny” z nadawaniem mu od razu określonego wyglądu to

za mało. Aby dokument mógł być efektywnie wykorzystywany wielokrotnie i w różny sposób (różne formy i konteksty prezentacji), konieczne jest dodawanie do tekstu dodatkowej informacji związanej z jego strukturą, a także być może treścią, tematyką itp.

Ponieważ tematyka Seminarium związana jest z internetem więc w dalszej części będziemy skupiać się głównie na zagadnieniach związanych z publikacją za pomocą tego „medium”. Opisane techniki nie można z powodzeniem stosować także do tekstów, które będą wykorzystywane w inny sposób.

### **3. Charakter serwisu WWW a opracowanie tekstów**

Sposób przygotowania tekstów na potrzeby publikacji w sieci zależy w dużej mierze od rodzaju serwisu. Serwisy WWW możemy w najprostszy z możliwych sposobów podzielić na dwie kategorie: serwisy małe i duże. Serwis mały to serwis zawierający niewiele (kilka) stron, które zmieniają się rzadko (nie częściej niż co kilka miesięcy). Do tej kategorii można zaliczyć właściwie jedynie strony osobiste. Wszystkie inne serwisy: internetowe wydania czasopism, serwisy firm, organizacji, ruchów lub stowarzyszeń to zwykle serwisy duże. Zwłaszcza jeśli mają one zawierać np. dział „aktualności”, w których pojawiać się będą różne bieżące informacje. W przypadku czasopism wydawać się może początkowo, że internetowe wydanie to serwis niezbyt duży. I tak być może jest w przypadku jednego numeru czasopisma. Jeśli jednak w WWW mają być dostępne numery archiwalne (a jest chyba oczywiste, że tak być powinno), to serwis gwałtownie się rozrasta.

Innym kryterium podziału serwisów WWW może być ich zawartość. Mogą istnieć serwisy reklamowe (czy nawet internetowe sklepy), serwisy będące zbiorami dokumentów, internetowe wydania czasopism, serwisy ogłoszeniowe, serwisy informacyjne (zawierające relacje z aktualnych wydarzeń). Mimo różnic jakie mogą występować między tymi rodzajami serwisów, mają one coś wspólnego. Wszystkie powinny zawierać indeksy i możliwości różnorodnego wyszukiwania informacji. W każdym z tych przypadków ważne jest więc aby przygotowywane do publikacji teksty zawierały dodatkowe informacje „strukturalne”, umożliwiające łatwe włączenie ich w całości (albo we fragmentach) w różne struktury indeksowe i/lub działy tematyczne.

### **4. Struktura logiczna tekstu**

Wszystko o czym powiedzieliśmy do tej pory wyraźnie wskazuje, że istnieje potrzeba takiego przygotowania tekstów, aby oprócz treści „właściwej” (tego co autor ma do powiedzenia) zawierały także informacje dodatkowe. Te informacje

mogą być dwojakiego rodzaju: służące do automatycznego klasyfikowania dokumentu oraz określające wyraźnie i jednoznacznie strukturę logiczną dokumentu. Sytuacja jest nieco podobna do wymagań jakie stawiane są tekstom drukowanym tradycyjnie na papierze. Gotowa książka oprócz treści zawiera różne elementy wprowadzone w procesie składu (różne kroje i wielkości czcionek, paginy, światła itp). Tekst bez tych dodatkowych elementów, które nic przecież nie wnoszą do zawartości merytorycznej, byłby bardzo trudny w odbiorze dla czytelnika.

W przypadku tekstów, które mają być publikowane w postaci elektronicznej droga między autorem a czytelnikiem jest nieco inna i inne są wymagania na ostateczną (jeśli w ogóle da się taką zdefiniować) postać dokumentu. Ponieważ nie jest znana z góry ostateczna forma prezentacji informacji to ograniczone zastosowanie ma klasyczne podejście do redakcji technicznej tekstu. Nie da się raz na zawsze definitywnie przypisać elementom strukturalnym tekstu fizycznych atrybutów (takich jak krój czy wielkość czcionek, różnego rodzaju światła, czyli odstępy między różnymi częściami tekstu itp). Można co najwyżej, przewidując możliwe sposoby prezentacji tekstu, przygotować różne *szablony* (ang. *style sheets*) dla różnych mediów, uwzględniające ich możliwości i ograniczenia. Na przykład na kolorowym terminalu tekstowym nie ma możliwości wyróżnienia tytułu przez powiększenie czcionki, ale jest możliwość zastosowania innego koloru.

W ten sposób dochodzimy do sedna sprawy. Ponieważ nie da się złożyć tekstu w sensie tradycyjnym, a tylko przygotować go do „włożenia w pewne szablony” lub późniejszej automatycznej obróbki, to wszystkie *logiczne* elementy tekstu muszą być właściwie zidentyfikowane i odpowiednio oznaczone. Takie przygotowanie tekstu oprócz umożliwienia automatycznego formatowania dokumentu, pozwala także dokonywać różnych niezbędnych operacji na tekście. Jeśli np. chcemy opublikować książkę czy inny duży tekst w WWW to oczywiste jest, że należy podzielić go na „kawałki” umieszczając każdy rozdział w innym pliku. Ale to nie wszystko. Każdy rozdział (poza być może pierwszym i ostatnim) powinien zawierać odnośniki (ang. *link*) do rozdziału poprzedniego i następnego. Tego typu podział i powiązania mogą być łatwo zrealizowane w sposób automatyczny (czyli za pomocą odpowiedniego programu), ale tylko wtedy kiedy informacja o granicach i zawartości rozdziałów będzie obecna w dokumencie źródłowym. I nie będzie oczywiście żadnym problemem zastosowanie tego schematu do dokumentu bardziej skomplikowanego — zawierającego więcej niż jeden poziom rozdziałów (np. rozdziały, podrozdziały itd.). Oczywiście w takim przypadku na każdej stronie zawierającej podstawową jednostkę podziału naszego tekstu (np. podrozdział) powinny znaleźć się odnośniki do sąsiednich podrozdziałów oraz do początku bieżącego rozdziału.

Co więcej tak przygotowany tekst można przetwarzać w różny sposób, bez konieczności wprowadzania jakichkolwiek zmian do jego zawartości „źródłowej”. Dokument staje się więc „otwarty”, jego format przestaje być przeszkodą w po-

nowym wykorzystaniu (jak to bardzo często niestety bywa — spróbujmy np. przenieść dokument zapisany w formacie edytora TAG na stronę WWW), a odwrotnie umożliwia dostosowywanie formy prezentacji do aktualnych wymagań i możliwości odbiorcy oraz pozwala na wielokrotne używanie tekstu w różny sposób i do różnych celów.

W tym miejscu warto zwrócić uwagę na jeden fakt: tekst zapisany w formacie uwzględniającym jego strukturę logiczną zwykle nie bardzo nadaje się do bezpośredniej prezentacji. Aby można go było wykorzystać (przeczytać, umieścić na serwerze zawierającym zbiór dokumentów itp.) musimy go przekształcić na inną postać albo zdefiniować odpowiedni proces konwersji oparty np. na wykorzystaniu szablonów. Ujawnia się więc w tym momencie podział na dwa typy formatów tekstów: format *strukturalny* i format *prezentacyjny*.

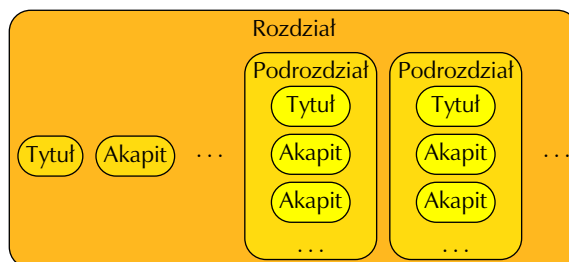
Dokumenty w formacie strukturalnym znakomicie nadają się do przetwarzania, indeksowania, klasyfikowania, umieszczania w bazach danych. Ponieważ nie zawierają informacji o atrybutach typograficznych czy zależnościach geometrycznych między różnymi elementami dokumentu, to aby można było je „wygodnie” przeczytać muszą być przetransformowane do jakiegoś formatu prezentacyjnego. I nie ma z tym żadnego problemu. Wystarczy z każdym z elementów logicznych dokumentu źródłowego związać pewien zestaw atrybutów fizycznych i zakodować je za pomocą odpowiednich poleceń formatujących. Takie przypisanie parametrów fizycznych do logicznych części dokumentu nazywane jest często szablonem. Przejście w drugą stronę, czyli od dokumentu w formacie prezentacyjnym do dokumentu strukturalnego nie jest możliwe bez udziału człowieka, który musi „zinterpretować” sformatowany tekst. Co więcej taka „interpretacja” wcale nie musi być jednoznaczna. . .

Formaty prezentacyjne nie muszą oczywiście odwoływać się bezpośrednio do fizycznych atrybutów, czy możliwości urządzeń wyjściowych. Współczesne formaty prezentacyjne takie jak PostScript, PDF (ang. *Portable Document Format*), RTF czy HTML to języki bardzo wysokiego poziomu. Służą one jednak głównie do definiowania atrybutów typograficznych i zależności geometrycznych między różnymi elementami dokumentu w czasie prezentacji. Posiadanie takich właśnie możliwości jest kryterium zakwalifikowania danego formatu do rodziny formatów prezentacyjnych.

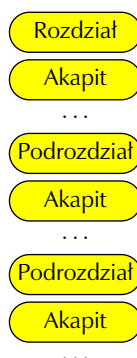
## 5. Narzędzia do strukturalizacji tekstów

Wiadomo już jakie wymagania musi spełniać tekst aby można go było efektywnie i wielokrotnie wykorzystywać. Pojawia się jednak pytanie kto, kiedy i jak powinien dokonywać „strukturalizacji” tekstu? Odpowiedź jest z pozoru prosta: powinno to być robione na możliwie najwcześniejszym etapie opracowania tekstu,

czyli już w momencie jego powstawania u autora, który używa odpowiedniego strukturalnego edytora tekstu. Jest jednak dosyć poważny problem. Popularne i modne edytory nie wspomagają tworzenia tekstów „strukturalnych”. We wszystkich właściwie edytorach tekstu i systemach składu istnieją co prawda tzw. „style” (ang. *styles*) ale ich możliwości są mocno ograniczone i niedostosowane do potrzeb strukturalizacji dokumentów.



Rysunek 1. Fragment struktury książki



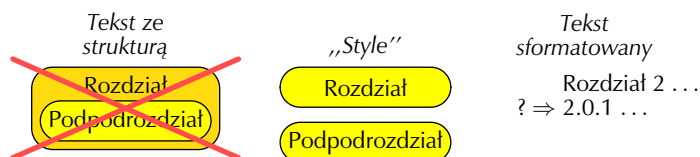
Rysunek 2. Użycie „stylów” do zdefiniowania struktury książki

Style można bowiem wiązać tylko z akapitami, co powoduje powstawanie płaskiej struktury logicznej tekstu, nie oddającej rzeczywistych logicznych zależności występujących w prawdziwych dokumentach. Rozdział książki składa się przecież np. z tytułu, po którym następuje, powiedzmy kilka akapitów wprowadzających, i dalej z podrozdziałów, znowu zawierających tytuł oraz akapity itd. Na rysunku 1 pokazany jest schematycznie fragment struktury książki. Widać tutaj hierarchię elementów logicznych struktury oraz relację zawierania elementów niższego poziomu w elementach wyższego poziomu.

Na rysunku 2 przedstawiony jest efekt próby wykorzystania mechanizmu stylów do zakodowania tej samej struktury książki. Nie istnieje możliwość tworzenia hierarchii stylów, w której style wyższego poziomu zawierałyby style niższego pozo-

mu — cała struktura się spłaszcza i przestaje być hierarchiczna. Co jeszcze gorsze — nie ma nawet miejsca na niektóre elementy: „Rozdział” czy „Podrozdział” zdefiniowany za pomocą stylów to w rzeczywistości tylko tytuły odpowiednio rozdziału i podrozdziału. Nie ma możliwości zdefiniowania elementów takich jak „Rozdział” (w sensie właściwym, a nie „stylowym”), które są tylko (a może aż) „pojemnikami” zawierającymi elementy niższego poziomu. Mogą istnieć tylko takie elementy, które zawierają bezpośrednio jakiś tekst. Są to bardzo poważne ograniczenia, które właściwie dyskwalifikują cały mechanizm stylów dostępny w edytorach czy systemach składu do strukturalizacji dokumentów.

Oczywiście nawet tak słabe narzędzie jak style mogłoby nieco usprawnić pracę. Ale jest jeszcze jeden problem z nimi związany. Nie ma możliwości sprawdzania logicznej „poprawności” dokumentu w taki sposób definiowanego. Wynika to po prostu, ze spłaszczenia całej struktury (braku możliwości zagnieżdżania elementów i tworzenia ich hierarchii). Bo przecież pożyteczna byłaby możliwość zagwarantowania unikania pewnych oczywistych błędów w strukturze. Na przykład nigdy nie jest dozwolone pominięcie jednego poziomu podrozdziałów umieszczenie elementu <Podpodrozdział> bezpośrednio w elemencie <Rozdział> (rys. 3)



Rysunek 3. Błąd w strukturze logicznej dokumentu

Na rysunku 3 obrazowo przedstawione są efekty próby stworzenia „niepoprawnego” dokumentu. W przypadku wykorzystywania strukturalizacji, sytuacja taka nie może się zdarzyć. Dokument, w którym nie są zachowane pewne ustalone zależności między elementami jest przez programy, służące do jego przetwarzania traktowany jako błędny i wymaga poprawienia. Jeśli użyjemy mechanizmu stylów, nie mamy możliwości weryfikacji poprawności struktury dokumentu. Może to prowadzić do sytuacji pokazanej z prawej strony, kiedy w wyniku niedopatrzania pominięty został jeden poziom podrozdziałów i w numeracji pojawiło się zero.

Narzędzia, jakie są naprawdę potrzebne to edytory strukturalne, czyli takie, które pozwalają oprócz samej treści i informacji związanej z formatowaniem dokumentu, wprowadzać oraz weryfikować także jego strukturę. Edytory strukturalne udostępniają także narzędzia umożliwiające efektywne poruszanie się po strukturze dokumentu: np. możliwe jest wyświetlenie wszystkich tytułów podrozdziałów rozdziału 2, albo przeniesienie całego logicznego fragmentu tekstu np. rozdziału wraz całą zawartością, bez konieczności jawnego sprawdzenia gdzie on się kończy,

w inne miejsce. Edytory strukturalne mogą także (choć nie muszą) umożliwiać definiowanie i wykorzystywanie szablonów na prywatne potrzeby autora.

## 6. Teksty strukturalne w świadomości użytkowników

Problem z dokumentami pisanymi przez większość autorów jest taki, że prawie nikt nie podejmuje żadnych prób umieszczenia w nich informacji na temat struktury, ułatwiających pracę ludzi, do których później trafiają ich teksty przed publikacją. I to nie dlatego, że nie mają odpowiednich narzędzi, a raczej dlatego, że nie są świadomi znaczenia jaką we współczesnej technice publikowania odgrywa logiczna struktura dokumentów. Wszyscy fascynują się możliwościami najnowszych wersji modnych edytorów tekstu, a nie potrafią przygotować tekstu tak, żeby ludzie publikujący ich dzieła mogli to robić sprawnie bez domyślania się „co jest czym” w tekście autorskim.

Co gorsza problemu nie dostrzega także wielu redaktorów i innych pracowników wydawnictw. Mało kto z nich wie cokolwiek o znaczeniu logicznej struktury tekstu (zwłaszcza w przypadku publikacji elektronicznych), a już niemal nikt nie słyszał o czymś takim jak SGML. Komu z czytelników tego tekstu ten skrót coś mówi, albo mówił zanim usłyszeli o nim na naszym Seminarium? A nawet jeśli ktoś coś o tym wie, to czy zetknął się z tym w praktyce, czy widział SGML w działaniu, czy słyszał o jakichkolwiek kursach związanych z tą tematyką? Pozytywnie na to pytanie nie odpowie chyba nikt. I jest to poważny problem. Bo na naszych oczach (które nie bardzo chcemy otworzyć) w dziedzinie opracowania i publikowania tekstów dokonuje się rewolucja na miarę Gutenberga, a nikt w Polsce tego nie zauważa.

Co ciekawe — DTP, o którym słyszeli wszyscy, to wynalazek młodszy i, jak się okazało, mniej użyteczny niż SGML. SGML (ang. *Standard Generalized Markup Language*) został formalnie opublikowany w październiku 1986 r. jako standard ISO-8879, ale jego historia jest znacznie dłuższa i sięga końca lat sześćdziesiątych. Czy ktoś w tym czasie przewidywał „rewolucję” DTP, która zaczęła się dopiero w roku 1985? A tymczasem minęła już największa fala zachłyśnięcia się możliwościami DTP i systemów typu WYSIWYG. Okazało się bowiem, że takie podejście, mające oczywiście wiele niewątpliwych zalet, ma także niezwykle bolesne ograniczenia. Ujawniły one się wtedy gdy pojawiła się potrzeba wielokrotnego opracowywania tych samych tekstów, które muszą być prezentowane za pomocą różnych mediów i od których wymaga się czegoś więcej niż tylko estetycznego wyglądu.

## 7. Co to jest SGML?

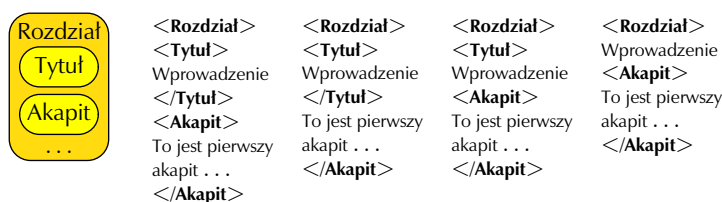
Pora więc zacząć opowiadać na postawione pytania. Częściowa odpowiedź na pytanie postawione w tytule tej części została udzielona już powyżej. Wszystko co powiedzieliśmy wcześniej na temat struktury logicznej tekstów obejmuje swoim zasięgiem standard SGML. SGML umożliwia kodowanie tekstów w sposób strukturalny. Nie jest to co prawda, żaden program, ani nawet konkretny format kodowania tekstu, ale coś bardziej ogólnego. Jest to mianowicie „metajęzyk” (czy „metaformat”) pozwalający definiować języki (formaty) służące do opisywania logicznej struktury dokumentów. SGML (nieco wbrew swojej nazwie) nie jest językiem w klasycznym sensie, bo nim być nie może, ze względu na cel jakiemu ma służyć. Nie da się bowiem (a nawet jeśli by się dało, to nie ma to sensu) za pomocą jednego ściśle określonego języka (formatu) opisać struktury wszystkich możliwych dokumentów. Popatrzmy jak różne części składowe mogą zawierać rodzaje publikacji:

- list: <nadawca>, <adresat>, <adres>, <data>, <powitanie>, <treść>, <pozdrowienia>, <podpis>;
- kryminał: <autor>, <tytuł>, <spis treści>, <rozdział>, <akapit>;
- artykuł naukowy: <autor>, <tytuł>, <streszczenie>, <sekcja>, <wzór>, <tabela>, <rysunek>, <twierdzenie>, <dowód>, <wniosek>, <definicja> itd.;

SGML nie służy więc bezpośrednio do kodowania struktury dokumentów, ale umożliwia tworzenie formalnych definicji różnych typów dokumentów. *Definicja Typu Dokumentu* (ang. *Document Type Definition*; DTD) jest jednym z podstawowych pojęć związanych z SGML-em. DTD jest to sformalizowany opis struktury dokumentów pewnego typu, zawierający spis ich elementów składowych oraz hierarchię elementów właściwą dla danego typu dokumentu. Można powiedzieć, że dopiero DTD jest definicją języka (czy formatu) do opisywania struktury danej klasy dokumentów.

Do oznaczania elementów w dokumentach służą *znaczniki* (ang. *tags*). Każdemu elementowi przypisany jest znacznik otwierający (<nazwa>) i znacznik zamykający (</nazwa>). Oprócz definicji elementów i znaczników z nimi związanych DTD zawiera także reguły ich używania: np. czy możliwe jest opuszczenie znacznika zamykającego lub otwierającego dany element, jeśli z kontekstu możliwe jest wywnioskowanie faktu jego istnienia.

Oto przykład: założmy, że element <rozdział> składa się obowiązkowego tytułu, po którym może nastąpić jeden lub więcej akapitów. Na rysunku 4 pokazany jest schemat zależności między elementami oraz cztery równoważne sposoby zaznaczenia struktury fragmentu tekstu.



Rysunek 4. Opuszczanie znaczników

Ogromną zaletą SGML-a jest także prawdziwa, a nie jedynie deklarowana, „otwartość” — dokumenty zakodowane z wykorzystaniem tego standardu nie są związane z jakąkolwiek platformą sprzętową, systemem operacyjnym, czy konkretnym systemem oprogramowania. W dobie popularności różnych edytorów, które umożliwiają podobno konwertowanie mnóstwa najróżniejszych formatów, a nie są w stanie poprawnie odczytać tekstów w swoich rodzimych formatach, tyle tylko, że przygotowanych w systemie z zainstalowanym innym zestawem czcionek, wydaje się to niemożliwe — ale tak jest naprawdę.

Wszystko to chyba wydaje się dziwnie znajome. Identyczną notację, takie same konwencje dotyczące możliwości opuszczania znaczników i niezależność (przynajmniej częściową) od oprogramowania i sprzętu znamy z języka HTML. Jest tak dlatego, że HTML to w rzeczywistości najbardziej znana *aplikacja* SGML. Niestety w język HTML, mimo iż oparty na standardzie SGML, należy do formatów prezentacyjnych i nie bardzo się nadaje do kodowania struktury dokumentów. Jest to język stosunkowo prosty i zorientowany na prezentację dokumentu w okienku przeglądarki WWW. Możliwości jakie oferuje są bardziej zbliżone (choć dużo uboższe) do mechanizmów „stylu” znanych z WYSIWYG, niż do prawdziwego kodowania strukturalnego.

Tworzone są już nowe standardy, które mają zastąpić bądź uzupełnić HTML. Jednym z nich jest XML — kolejna aplikacja SGML, mająca zachować (niestety niezupełnie) zgodność „w dół” z językiem HTML. Uproszczenie w stosunku do „pełnego” standardu SGML polega przede wszystkim na możliwości zrezygnowania z definiowania DTD, ponieważ jego opracowanie jest sprawą złożoną. W wielu praktycznych zastosowaniach, zwłaszcza związanych z elektroniczną publikacją dokumentów hipertekstowych o prostej strukturze, nie jest to nawet konieczne. XML to jednak jeszcze przyszłość — narzędzia do przygotowywania oraz prezentacji tekstów dopiero powstają, a i sama definicja języka XML nie jest jeszcze ostatecznie ustalona.

## 8. Teksty, ludzie i narzędzia

Co zatem można zrobić, aby skorzystać z możliwości jakie dają odpowiednio wykorzystane narzędzia i standardy? Jak zorganizować pracę redaktorów i autorów tekstów, które mają być (albo mogą kiedyś być, a to już dotyczy niemal wszystkich tekstów) udostępniane i prezentowane w różny sposób?

Przed wszystkim należy uświadomić sobie znaczenie właściwego przygotowania tekstów nie tylko pod względem merytorycznym, ale także „strukturalnym”. Ważne jest zrozumienie zysków i nowych możliwości jakie daje odpowiednie opracowanie tekstów.

Konieczne trzeba mieć także świadomość, że dla sprawnej pracy redakcji przygotowującej teksty do publikacji elektronicznej niezbędna jest współpraca różnych specjalistów. Potrzebni są oczywiście kompetentni autorzy. Są oni zwykle stosunkowo najluźniej związani z redakcją. Ale teksty rodzą się przecież właśnie u nich i dobrze byłoby aby już w czasie powstawania otrzymywały także odpowiednią formę. Dlatego koniecznie należy zmienić przyzwyczajenia większości autorów, którzy nie dbają zupełnie o techniczną jakość swoich tekstów. Nie jest ważne jakiego edytora używa, czy w jakim systemie pracuje autor, ważne jest aby ludzie pracujący później nad dokumentem nie musieli zgadywać co jest czym. W większości wypadków wystarczą rozwiązania bardzo proste. Nawet zwykły HTML (byłe rozsądnie używany i nie generowany automatycznie z modnego edytora tekstów) może być czasem, pomimo swoich ograniczeń, bardzo rozsądnym rozwiązaniem.

Trzeba także pamiętać, że elektroniczne publikowanie tekstów nie jest zwyczajną działalnością wydawniczą. Nie wystarczą typowe narzędzia i techniki „klasycznego” DTP. Nie istnieją na razie (i nie bardzo jeszcze wiadomo kiedy i czy w ogóle będą istniały) standardowe narzędzia do obróbki elektronicznych tekstów podobne do programów służących do składania tekstu. Sytuacja w przypadku publikacji elektronicznych jest bardziej złożona niż przy klasycznym składzie, gdzie wymagania co do ostatecznej formy i wyglądu tekstu są dobrze określone. Tekst ma być złożony i proces składu jest ostatnią operacją jaka będzie na nim wykonywana. Wydrukowane czasopismo albo książka to ostateczny i nigdy już nie zmieniany efekt pracy. Możliwe jest więc stworzenie odpowiednich programów realizujących w sposób mniej czy bardziej doskonały proces składu tekstu.

Inaczej jest w przypadku publikacji elektronicznych. Ważną cechą tego typu udostępniania informacji jest „otwartość” czyli możliwość innego zorganizowania (np. posortowania, wybrania, wyszukania) informacji. Poza tym nowe możliwości sprzętowe oraz prezentacyjne (nowe standardy prezentacji informacji) sprawiają, że nie da się określić raz na zawsze schematu przetwarzania dokumentu. Przed wszystkim takich schematów może być wiele. Efekt końcowy jednego z nich może być całkiem klasyczny — wydrukowanie tekstu na papierze. Ale to nie jest wszystko. Teksty elektroniczne mogą być umieszczane w bazach danych indek-

sowane, przetwarzane „w locie” do formatów prezentacyjnych. I każdy z tych sposobów przetwarzania, czy wykorzystywania informacji wymaga odpowiedniego oprogramowania. Oprogramowanie to może nawet nie istnieć w momencie powstawania tekstu. Trudno więc wyobrazić sobie tutaj jakieś standardowe narzędzia „do wszystkiego” jakie znamy z technologii DTP.

Oprogramowanie wykorzystywane np. do indeksowania tekstów w bazie danych musi być „świadome” typów dokumentów jakich może się spodziewać. A przecież różnorodność typów dokumentów jest praktycznie nieograniczona. Nie ma sensu zwykle uwzględniać wszystkiego. Ale z drugiej strony nie można zakładać, że na naszym serwerze zawsze będą umieszczane dokumenty tylko takich typów jakie już tam są. Trzeba także pamiętać, że dokumenty elektroniczne są zwykle między sobą powiązane, poza tym niektóre z nich muszą być „generowane w locie”. Indeks musi się np. zmieniać (rozszerzać) w miarę przybywania dokumentów dostępnych na serwerze. Formaty prezentacyjne także się zmieniają. A musi istnieć np. możliwość prezentacji „starych” dokumentów w „nowych” formatach.

Wszystko co powiedzieliśmy powyżej wyraźnie wskazuje, że w przypadku dokumentów elektronicznych i narzędzi związanych z ich obróbką i prezentacją pojawia się konieczność skorzystania z pomocy informatyków. I pomoc ta nie może się jak już wiemy ograniczyć tylko do jednorazowego napisania odpowiednich programów. Tak jak redakcją merytoryczną musi zajmować się odpowiednio wykształcony redaktor, tak systemem informatycznym, jakim jest serwer zawierający bazę dokumentów musi zajmować się odpowiednio przygotowany informatyk. I podobnie jak w przypadku redaktora, ingerencja informatyka może być konieczna tylko na pewnych etapach. W przypadku redaktora jest to np. korekta, w przypadku informatyka pojawienie się potrzeby prezentacji dokumentów w nowym formacie, albo zastosowania nowych sposobów czy kryteriów wyszukiwania.

## 9. Przykład

Wszystkie referaty prezentowane na tym Seminarium zostały opracowane z wykorzystaniem możliwości, o których tu piszemy. Od początku zakładaliśmy, że teksty będą dostępne w różnych postaciach i to zarówno przed, w trakcie, jak i po Seminarium. Przed były dostępne dla organizatorów (w tym wszystkich prelegentów) w postaci elektronicznej, ale nadającej się do wydrukowania (Postscript, PDF). Ponieważ materiały dochodziły z różnych stron, w różnych formatach i w różnym czasie, to każdy z nich był wstępnie w zasadzie samodzielną całością. Forma była dostosowana do potrzeb np. jako czcionka podstawowa został wybrany standardowy Times-Roman, ponieważ jest to czcionka wbudowana i dostępna we wszystkich drukarkach postscriptowych i programach do przeglądania i druko-

wania dokumentów PDF. Jako powszechnie dostępna nie musi być dołączana do dokumentów, dzięki czemu są one mniejsze.

Na potrzeby wydrukowania materiałów w drukarni trzeba było wszystkie teksty potraktować jako całość i złożyć razem, generując przy okazji np. spis treści. Materiały będą (są) także dostępne w postaci elektronicznej w sieci WWW. Są one dostępne w zwyczajnym formacie HTML oraz w formacie PDF. Zostaną także przygotowane wersje materiałów dostępne na płytach CD-ROM (PDF, HTML).

Wszystkie te postacie wynikowe dokumentów generowane są automatycznie z wersji źródłowych zakodowanych z wykorzystaniem SGML. Użyte DTD to DocBook opracowany w amerykańskim wydawnictwie O'Reilly and Associates. Część materiałów była od razu pisana w tym formacie z wykorzystaniem edytora strukturalnego, który potrafi korzystać z informacji zawartych w DTD, pokazując np. autorowi elementy dopuszczalne w danym kontekście. Edytorem tym był Emacs pracujący w specjalnym trybie psgml.

Część dokumentów była konwertowana z innych formatów (Tag, Word,  $\LaTeX$ ). Nie korzystano z żadnych standardowych filtrów, ani konwerterów dostępnych w różnych edytorach tekstów. Na potrzeby konwersji tworzone były odpowiednie filtry w języku perl. Część pracy została wykonana „ręcznie” przy użyciu edytorów Emacs i vi.

Do weryfikacji poprawności dokumentów zakodowanych w SGML użyty został parser nsgmls autorstwa Jamesa Clarka. Parser ten wykorzystany został także przy przetwarzaniu dokumentów do formatów prezentacyjnych. Szablony służące do konwersji na formaty prezentacyjne zostały napisane w języku perl z wykorzystaniem specjalizowanej biblioteki SGMLSpm oraz pakietu sgmlspl autorstwa Davida Megginsa.

Do opracowania wersji drukowanej i oraz wersji w formacie PDF, wykorzystaliśmy program pdftex z formatem  $\LaTeX$ .

Poniżej, jako przykład dokumentu źródłowego zakodowanego z wykorzystaniem SGML, pokazujemy fragment tekstu ks. Krzysztofa Goneta:

```
<article>
<arheader>
<title>Wiara, biblioteki i~Internet<comment>Wypowiedź na "Warsztatach
  nt tworzenia serwisów WWW" zorganizowanych przez Radę Ruchów
  Katolickich w~Warszawie w~dniu 15 marca 1998 r.</comment></title>
<author>
<honorific>Ks.</honorific>
<firstname>Krzysztof</firstname>
<surname>Gonet</surname>
<authorblurb>
<para>kgon@wmsd.edu.pl</para>
</authorblurb>
</author>
</arheader>
```

```
<sect1>
<title>Świat globalnego porozumienia w roku 2000 a "pełnia czasu"
w roku 0</title>

<para>"Świat globalnego porozumienia", który budujemy na
przełomie drugiego i trzeciego tysiąclecia przypomina warunki,
które nastąpiły na przełomie "starej" i "nowej ery" w roku 0.
Św. Paweł określił tamten moment "pełnią czasu". [Gal 4,3]</para>

<para>Ta pełnia miała jednak nie tylko religijny wymiar. Ten
czas, gdy Bóg wszedł w ludzką historię stając się człowiekiem,
był jednak niezwykle również w całym materialnym wymiarze.</para>
<itemizedlist>
<listitem>
<para>Oto świat cywilizacji Europejskiej został zjednoczony w jeden
organizm państwowy Cesarstwa Rzymskiego.</para>
</listitem>
<listitem>
<para>Na jego terenie panował względny pokój ("Pax Romana").</para>
</listitem>
<listitem><para>Na całym terenie Cesarstwa można się było
porozumieć w jednym języku --- po grecku.</para></listitem>
<listitem><para>Rozległe tereny były połączone wspaniałymi, jak na
owe czasy, drogami rzymskimi i bezpiecznym (po opanowaniu
piractwa) transportem morskim.</para></listitem>
<para>
Ks. Marek Starowieyski: "Praeparatio Evangelica". W: Pierwsi
świadkowie. Wybór najstarszych pism chrześcijańskich. Kraków:
ZNAK, 1988, s. 5--14</para>
</listitem>
</itemizedlist>
<para>Opisane warunki dawały niezwykle możliwość międzyludzkiej
komunikacji. W ówczesnych warunkach, dwa tysiące lat temu tak
właśnie wyglądał "świat globalnego porozumienia".</para>
....
</sect1>
....
</article>
```

Przytoczyliśmy tu dosyć duży fragment tekstu, ale chodzi nam o to aby zwrócić uwagę na kilka spraw:

- Mimo być może skomplikowanego na pierwszy rzut oka wyglądu, dokument zakodowany w SGML-u jest czytelny dla człowieka. Chodzi przede wszystkim o to, że dokumenty SGML to dokumenty tekstowe. Do tworzenia bądź modyfikowania nie wymagają żadnych specjalizowanych programów typu wordprocesory, albo wymyślne edytory tekstów. Każdy najprymitywniejszy

nawet edytor tekstowy wystarczy do odczytania, bądź poprawienia tekstu dokumentu.

- Dokument jest wyraźnie podzielony na dwie części: nagłówek i część właściwą. W nagłówku umieszczone są oczywiście informacje na temat autora i ogólne informacje o dokumencie. Nagłówek jaki widać tutaj jest dosyć prosty. Potencjalnie może on zawierać dużo więcej informacji np. słowa kluczowe, informacje bibliograficzne, katalogowe (ISBN, ISSN) itd.
- Na powyższym fragmencie tekstu widać także jeszcze jedną cechę SGML-a, o której wcześniej nie wspominaliśmy. Jest to mianowicie możliwość stosowania skrótów. W SGML-u istnieje wiele mechanizmów pozwalających na zmniejszanie znakowania niezbędnego do prawidłowej interpretacji dokumentu. Nie będziemy wdawać się w szczegółowy opis wszystkich dostępnych mechanizmów. Chcemy tylko na jednym małym przykładzie pokazać istniejące możliwości.

Popatrzmy na cudzysłowy. Cudzysłowy jakie znajdują się w tekście źródłowym są „typu komputerowego” ("). Nie ma różnicy między cudzysłowem otwierającym i zamykającym. Oczywiście często autorom jest wygodnie stosować taką konwencję. Muszą zadbać tylko o właściwe ich sparowanie. W rzeczywistości znaki cudzysłowu są ogranicznikami i w tym przypadku para znaków " jest równoważna parze etykiet elementu służącego do zaznaczania cytatów (<quote>, </quote>). Fragment tekstu "pełnia czasu" jest więc równoważny z <quote>pełnia czasu</quote>. Mechanizm interpretacji cudzysłowów pokazujemy tutaj bez szczegółowego omówienia tylko jako przykład pokazujący, że mechanizm skrótów dostępny w SGML-u jest w niezwykle silnym narzędziem (umożliwia np. rozróżnianie znaczenia tych samych symboli w zależności od kontekstu).

W przytoczonym fragmencie występują jeszcze inne rodzaje skrótów ('~' i '---'), ale ich znaczenie jest inne. Znak '~' to twarda (niełamliwa) spacja czyli miejsce, w którym nie wolno zakończyć wiersza, Trzy minusy '---' służą do zaznaczenia znaku myślnika (pauzy '—', ang. *m-dash*), który jest inny niż znak przenoszenia (-) czy znak matematycznej operacji odejmowania (-). Niestety nie wszędzie da się tę różnicę zachować — na stronie WWW w HTML-u może być problem z rozróżnieniem tych znaków, ale to już jest sprawa formatu prezentacyjnego — w dokumencie źródłowym tego typu znaki muszą być rozróżnione.

- W przykładowym fragmencie tekstu wyraźnie widać także różnicę w stosunku do kodowania tekstów na potrzeby DTP, o której wspominaliśmy wcześniej. Niektóre elementy takie jak <sect1>, <itemizedlist>, <listitem>, <footnote>, nie zawierają bezpośrednio tekstu (treści) — są tylko (albo aż) „pojemnikami” na elementy niższego poziomu.

- Wyraźnie widoczna jest także hierarchia elementów: elementy niższego poziomu są całkowicie „zanurzone” w elementach wyższego poziomu. I tak np. słowa *pełnia czasu* w tytule pierwszej części tekstu stanowią zawartość elementu `<quote>` (pamiętamy o skrótach) i jednocześnie część zawartości elementu `<title>`, który stanowi część zawartości elementu `<sect1>`, który wreszcie stanowi część zawartości elementu najwyższego poziomu `<article>`.

Ta hierarchia jest często bardzo istotna, bo definiuje kontekst znaczeniowy danego elementu. Na przykład tytuł może być tytułem rozdziału, podrozdziału, tabeli, rysunku, itd. Jest to zawsze tytuł i wygodnie jest go oznaczać wszędzie w ten sam sposób. Już jednak sposób jego formatowania, czy wykorzystania, zależą od tego, czego tytułem jest dany tytuł. I tak np. tytuł rozdziału, poza tym, że ma być sformatowany w odpowiedni sposób musi trafić do spisu treści, a tytuły tabel, do spisu tabel. I znowu, tak jak w przypadku skrótów, sygnalizujemy tylko pewien fakt, który jest w rzeczywistości istotnym zagadnieniem projektowym związanym z zadaniem tworzenia definicji typów dokumentów oraz definiowania sposobów przetwarzania tekstów na potrzeby prezentacji. Wyczerpujące omówienie tego zagadnienia znacznie przekracza ramy tego tekstu i naszego spotkania.

Przy okazji analizy struktury przykładowego tekstu warto wspomnieć o jeszcze jednej ważnej sprawie. Stosując techniki, o których jest mowa w tym tekście wiele zyskujemy. Przede wszystkim praca przebiega niezwykle sprawnie. Raz wybrane standardy kodowania strukturalnego będą nam służyły w przyszłości. Nawet jeśli zajdzie potrzeba wykorzystywania tekstów do celów, których nie jesteśmy sobie w stanie jeszcze nawet wyobrazić, to nie będzie potrzeby ponownego ręcznego opracowywania każdego tekstu. Wystarczy przygotować odpowiednie narzędzia (filtry), które dokonają niezbędnych przekształceń dokumentów automatycznie.

Właściwie opracowane teksty pozwalają także zaoszczędzić mnóstwo pracy na etapie przygotowywania tekstów do późniejszego wykorzystania i prezentacji. Opracowując teksty na potrzeby tego seminarium najmniej problemów mieliśmy z tekstami pisanymi od razu w sposób strukturalny z wykorzystaniem SGML-a. Inne teksty np. nadesłane w  $\text{\LaTeX}$ -u, także nie sprawiały wielkiego problemu i wszystkie elementy strukturalne (np. wyróżnienia kursywą czy pismem pogrubionym) „przeszły” bez wielkiego problemu — wystarczył mały skrypt w perlu i kilka niewielkich ręcznych poprawek. Podobnie teksty w zapisane w HTML-u nie stanowiły problemu. Tutaj (zwłaszcza, że autor, od którego dostaliśmy tekst w HTML-u stosował pewne zdroworozsądkowe konwencje) można było zastosować od razu narzędzia SGML-owe. Najwięcej pracy było z tekstami nadesłanymi w formatach typowych edytorów tekstów (TAG czy MS-Word). Ponieważ nie są nam dostępne takie programy, więc mimo iż sam tekst można przy odrobinie wprawy odzyskać

bez większego problemu, to wszelkiego typu wyróżnienia w tekście już nie bardzo. Poza tym dokument trzeba później ręcznie „poznakować”, czyli dokonać identyfikacji jego struktury logicznej. W przypadku tekstów o prostej strukturze jakie znajdują się w materiałach, nie stanowiło to wielkiego problemu, ale jednak zajęło niepotrzebnie dużo czasu.

## 10. Podsumowanie

Doświadczenia jakie zebraliśmy zajmując przez kilka lat różną działalnością związaną z przetwarzaniem informacji tekstowych na potrzeby różnych metod prezentacji, od tradycyjnej działalności wydawniczej do publikacji dokumentów elektronicznych w sieci WWW czy na płytach CD-ROM wyraźnie wskazują na znaczenie odpowiedniego opracowywania tekstów. Istnieją odpowiednie standardy i narzędzia. Dostępne są w sieci obszerne informacje na temat doświadczeń innych środowisk i ośrodków. Możliwe jest także tworzenie profesjonalnych i niezwykle wydajnych systemów przetwarzania i udostępniania dokumentów elektronicznych w oparciu o oprogramowanie dostępne nieodpłatnie. Przykładem takiego rozwiązania jest właśnie opracowanie niniejszych materiałów. Inny przykład z tej dziedziny przedstawiony jest w tekście opisującym serwis WWW Ruchu Światło-Życie. To co jest konieczne do realizacji tego typu przedsięwzięć można podsumować w kilku punktach:

1. W sposób przemyślany musi być zorganizowany przepływ informacji między ludźmi zaangażowanymi w przedsięwzięcie.
2. Należy opracować jednoznaczne i możliwie najprostsze schematy przepływu informacji i ustalić standardy kodowania. Najlepiej jest jeśli schematy te są oparte na przepływie informacji całkowicie tekstowej, którą da się modyfikować w zwyczajnych edytorach tekstu. Do przetwarzania można używać dowolnie skomplikowanych narzędzi, ale standardy komunikacyjne *muszą być najprostsze z możliwych*.
3. Konieczne należy rozdzielić proces kodowania struktury logicznej tekstu od procesu formatowania. Tekst powinien być tworzony w sposób strukturalny już przez autora. Nie zawsze musi być konieczne korzystanie ze wszystkich możliwości jakie dają odpowiednie narzędzia i standardy.
4. Konieczne jest aby w tego rodzaju przedsięwzięciach (zwłaszcza w fazie planowania i rozruchu) brał udział odpowiednio przygotowany informatyk, znający nowoczesne techniki związane z przetwarzaniem i prezentacją dokumentów tekstowych, umiejący opracować schematy komunikacji i standardy wymiany danych dostosowane do potrzeb, przyzwyczajień i możliwości percepcyjnych oraz ograniczeń członków zespołu.

Do zadań informatyka należeć powinna także ocena przydatności narzędzi używanych przez członków zespołu w codziennej pracy i ew. zaproponowanie konwencji jakich powinni oni przestrzegać aby jak najbardziej usprawnić przepływ informacji.

Ważne jest, że dodatkowy wysiłek jaki trzeba włożyć w opanowanie technik i pojęć, o których piszemy oraz zdobycie i oswojenie narzędzi szybko daje efekty. Już zastosowanie pewnych najprostszych technik, nawet bez używania specjalizowanych narzędzi (np. edytorów strukturalnych) daje wymierne efekty. Nawet tak prymitywne i w sumie niedopasowane do potrzeb narzędzia jak mechanizm „stylów” w programach DTP i powszechnie dostępnych edytorach, jeśli będą stosowane w sposób konsekwentny i przemyślany mogą znacznie ułatwić przetwarzanie tekstu w przyszłości i oszczędzić wiele czasu.

Na odpowiednie narzędzia przyjdzie czas. Najważniejsza jest świadomość istnienia problemu i jego znaczenia. I to u wszystkich. Naszym zdaniem najwięcej do zrobienia zostaje na etapach najwcześniejszych czyli na biurkach autorów. Jednocześnie na tym pierwszym etapie pracy nad tekstem można osiągnąć najwięcej. I to nie kosztem autorów, którym pozornie chcemy dorzucić nowe obowiązki i zadania. Czy nie wystarczy, że autor będzie świetnie znał się na tematyce, o której pisze? Czy, zwłaszcza jeśli nie ma on technicznego wykształcenia, należy mu zwracać głowę takimi sprawami jak struktura tekstu i stosowanie odpowiednich technik i narzędzi do jej nadania powstającemu tekstowi? Otóż właśnie należy i mimo pozornie zwiększonego nakładu pracy już autor (a dzięki niemu i przyszli jego czytelnicy) odniesie wymierne korzyści. Dbanie o „porządek strukturalny” ułatwia zadbanie o „porządek logiczny” całej wypowiedzi. Techniki — jak to techniki szybko wchodzą w nawyk i usprawniają pracę. A strukturalizacja tekstu sprowadza się tak naprawdę do pewnych technik, które szybko dają zyski, tym bardziej, że na dalszych etapach wykorzystania tekstu rośnie rola automatycznego przetwarzania. Główny zysk można więc osiągnąć na etapach początkowych, gdzie rola człowieka jest bardzo duża.

## 11. Źródła informacji

Na temat SGML-a i elektronicznych publikacji dostępne są bardzo bogate źródła informacji. Wydano wiele niezłych książek (niestety nie po polsku), z których najbardziej wyczerpującą pozycją jest „The SGML Handbook” autorstwa Charlesa F. Goldfarba. Książka ta zawiera kompletny tekst standardu ISO-8879 oraz bardzo dobre tutoriale. Nie polecamy jej jednak początkującym.

Dla początkujących świetnie natomiast nadaje się tekst „A Gentle Introduction to SGML” (drugi rozdział dokumentu „TEI Guidelines” dostępnego pod adresem <http://www.uic.edu/orgs/tei/p3/doc/p3.html>).

Najważniejszy serwis WWW, w którym można znaleźć najwięcej informacji na temat SGML-a dostępny jest pod adresem <http://www.sil.org/sgml/sgml.html>. Jest to najobszerniejszy dostępny w internecie katalog obejmujący wszystkie zagadnienia związane z SGML-em i sprawami pokrewnymi. Dostępna jest tam obszerna bibliografia, spis dostępnego oprogramowania (nieodpłatnego i komercyjnego), opis istniejących zastosowań (przemysłowe, rządowe, akademickie) oraz odnośniki do serwerów zawierających informacje na tematy pokrewne: HTML, XML, DSSSL, VRML itp.

W Polsce istnieje w zasadzie tylko jedna strona poświęcona standardowi SGML, zawiera niestety bardzo mało informacji i nie jest zbyt ciekawa (<http://www.fuw.edu.pl/ksiezyk/sgml.html>)