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:

dra hab. Jan Kowalski

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.

Spis treści

W]	prowadzenie	4
1.	Wprowadzenie do standardów SGML/XML	5
	1.1. Elementy składowe systemu SGML/XML	5
	1.2. Opracowanie typu dokumentu	6
	1.3. DocBook	7
	1.4. Edytory sterowane składnią	8
	1.5. Emacs i psgml	8
2.	Narzędzia i standardy pokrewne	9
	2.1. Przetwarzanie dokumentów SGML – standard DSSSL	9
	2.2. Przetwarzanie dokumentów XML – standard XSL	10
3.	Przegląd dostępnych narzędzi	11
	3.1. Narzędzia do przeglądania dokumentów SGML/XML	11
	3.2. Parsery SGML/XML	11
	3.3. Wykorzystanie języków skryptowych	12
	3.4. Wykorzystanie szablonów XSL	12
4.	Model Akceptacji Technologii	14
	4.1. Podstawowe informacje	14
Za	kończenie	17
A.	Tytuł załącznika jeden	18
В.	Tytuł załącznika dwa	19
Bil	bliografia	20
Sp	is tabel	23
Sp	is rysunków	24
Sk	orowidz	25

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 Goldfarb (1993) 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.

1.1. Elementy składowe systemu SGML/XML

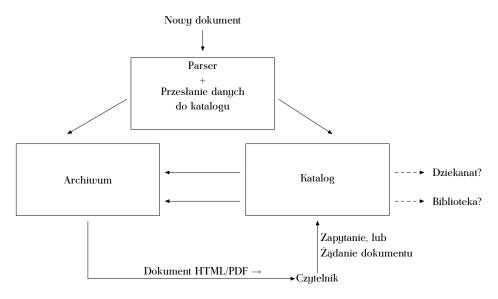
Standard SGML/XML to specyfikacja techniczna metajęzyka North (2000); Burnard i Sperberg-McQueen (2002). Zaś systemem SGML/XML nazywamy zestaw narzędzi i środków niezbędnych do tworzenia, składowania i obróbki dokumentów z wykorzystaniem tego standarduBurnard i Sperberg-McQueen (2002). Typowy proces produkcji dokumentów w standardzie SGML/XML podzielony jest na kilka części. Najważniejszymi elementami tego procesu są (Maler i Andaloussi, 1996, s. 45–47):

• Klasyfikacja tworzonych dokumentów w grupy i wynikający z tego dobór definicji typu dokumentu¹ (por. punkt 1.2).

¹Definicja 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).

- 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

1.2. Opracowanie typu dokumentu

Tworzenie DTD Maler i Andaloussi (1996); Maler (1998); Walsh i Muellner (1999) powinno zacząć się od analizy posiadanych danych oraz potrzeb dotyczących dokumentów wynikowych i informacji. Rezultatem analizy dokumentów jest wyspecyfikowanie (Maler, 1998,

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

s. 212–218) słownika (Burnard i Sperberg-McQueen, 2002, s. 111A–118C) 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* Burnard i Sperberg-McQueen (2002) oraz DocBook opracowany przez konsorcjum firm, głównie sektora IT, używających technologii SGML/XML do tworzenia dokumentacji do swoich produktów Walsh i Muellner (1999).

1.3. DocBook

DocBook DTD to typ dokumentu definiujący strukturę i zawartość zbioru znaczników SGML, służących do dokumentacji oprogramowania i różnego rodzaju dokumentacje techniczne (Walsh i Muellner, 1999, 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 niekomercyjnych projektach tworzenia dokumentacji komputerowej³Burnard i Sperberg-McQueen

³Konstatując brak tanich i funkcjonalnych edytorów do pracy z dokumentami SGML/XML należy wspomnieć

(2002); North (2000); Clark (1999); Bryan (1997); Beebe (1995); Lamport (1994); Bray i inni (1998) (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 WYSI-WYG 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

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...

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 problemów¹.

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-

¹Z 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 XSLT (autor N. Walsh). W chwili obecnej są dostępne narzędzia umożliwiające przetworzenie dokumentów XML do postaci drukowanej (Adobe PDF, por. Adobe Systems Inc. (1996)) oraz hipertekstowej (HTML, por. Bryan (1997); Bray i inni (1998); Van Herwijnen (1990)).

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

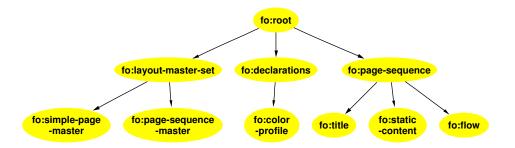
Tabela 3.1. Publicznie dostępne procesory XLST

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.

ROZDZIAŁ 4

Model Akceptacji Technologii

4.1. Podstawowe informacje

Model Akceptacji Technologii (Technology Acceptance Model, TAM) jest najczęściej wykorzystywaną teorią objaśniającą wykorzystanie i akceptację systemów informacyjnych w praktyce badawczej Informatyki Ekonomicznej Venkatesh i inni (2003); Venkatesh i Davis (2000). Lee i inni Lee i inni (2003) określają model TAM jak "najczęściej stosowaną teorię opisującą indywidualną akceptację systemów informacyjnych". Model TAM został zaadaptowany z rozwiniętej na gruncie psychologii społecznej teorii uzasadnionego działania Ajzena i Fishbeina (Theory of Reasoned Action, TRA) i teorii planowanego działania (Theory of Planned Behaviour, TPB) będącej jej rozwinięciem Fishbein i Ajzen (1975); Ajzen (1991). Według teorii uzasadnionego działania działanie poprzedza intencja, która z kolei kształtuje się pod wpływem dwóch czynników: subiektywnej normy¹ oraz postawy² względem tego zachowania. Wreszcie postawy wynikają z przekonań (beliefs) względem użytkowania (systemów informacyjnych w tym przypadku). Teoria planowanego działania jest rozszerzeniem TRA i zakłada, że działania kształtuje oprócz norm subiektywnych i postaw także trzeci czynnik określany jako postrzegana kontrola behawioralna³. PBC jest wykorzystywany w rozszerzonych wersjach modelu TAM, por. przykładowo Venkatesh i inni (2003) lub Mathieson i inni (2001).

TRA/TBP to ogólny model zachowania a jego konkretyzacja polega m.in. na określeniu przekonań, które są istotne dla zachowania będącego przedmiotem zainteresowania badacza. Inne przekonania determinują palanie bądź nie palenie tytoniu a inne używanie bądź nie używanie aplikacji komputerowych. Davis postuluje, że kluczową rolę odgrywają dwa przekonania⁴ nazwane przez niego postrzegana użyteczność *preceived usefulness*) oraz postrzegana łatwość użytkowania. (PEOU, *perceived ease of use*).

Pierwsze przekonanie jest określone jako "stopień przekonania użytkownika, że korzy-

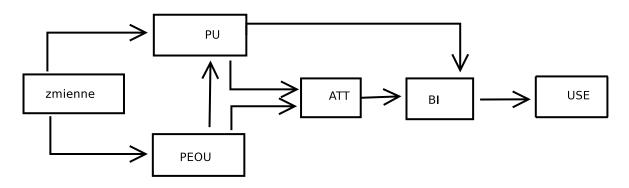
¹Subiektywne normy (SN, *subjective norms*) to "przekonania jednostki na ile jej działania będą akceptowane na tak lub nie przez ważne dla tej jednostki osoby" (Ajzen, 1991, s. 302) za (Venkatesh i Davis, 2000, s. 187).

²Postawa względem zachowania (ATT, attitude toward behaviour), to przekonania względem konsekwencji zachowania oraz ocena tych konsekwencji dla jednostki.

³Postrzegana kontrola behawioralna (PBC, *perceived behavioural control*) to przekonanie co do możliwości wykonania działania Ajzen (bdw). Ajzen wskazuje na podobieństwo PBC z czynnikiem samoskuteczności z teorii społecznego uczenia S. Bandury (Ajzen, 1991; Bandura, 1994, s. 184).

⁴Ajzen i Fishbein rekomeđują więcej – od pięciu do dziewięciu przekonań (Fishbein i Ajzen, 1975, s. 218).

stanie z określonego systemu zwiększy jego efektywność pracy" Davis i inni (1989); Davis (1989). Z kolei postrzegana łatwość użytkowania jest definiowana jako "stopień przekonania użytkownika, że korzystanie z określonego systemu będzie łatwe" Davis i inni (1989); Davis (1989). Skale pomiarowe dla obu czynników zostały opracowane i zweryfikowne w Davis (1989) oraz w Adams i inni (1992). Reasumując: "klasyczny" model TAM zawiera pięć czynników (por. rys. ??): postrzegana użyteczność (PU), postrzegana łatwość użytkowania (PEOU), postawa wobec używania (ATT), intencja używania (BI) oraz wykorzystanie (USE)⁵.



Rysunek 4.1. Model Akceptacji Technologii, opracowanie własne za Davis i inni (1989)

Model TAM nie zawiera czynnika *norm subiektywnych*, który to czynnik jest według hipotez TRA/TBP istotnym determinantem intencji Fishbein i Ajzen (1975), (Taylor i Todd, 1995, s. 150) oraz (Burton-Jones i Hubona, 2005, s. 59–60). Davis uprościł swój model usuwając z niego czynnik *norm subiektywnych* ponieważ, po pierwsze ich wpływ w przeprowadzonych przez niego eksperymentach nie był znaczący, a po drugie stosowane do pomiaru *norm subiektywnych* skale psychometryczne były w jego opinii słabe (Davis i inni, 1989, s. 986)⁶. Teoria TRA/TBP zakłada ponadto, że przekonania wyjaśniają intencje *wyłącznie*, za pośrednictwem norm subiektwych i postawy (oraz kontroli behawioralnej w przypadku TPB)⁷ podczas gdy w oryginalnym modelu TAM (por. rys. 4.1) postulowana jest statystycznie istotna bezpośrednia zależność także pomiędzy PU-BI. Późniejsze badania potwierdziły empirycznie, że istotnie w modelu TAM istnieje związek pomiędzy PU-BI. W rezultacie

⁵Czynniki ATT oraz BI są wprost zaadaptowane z modeli TRA/TPB.

⁶Ta decyzja była słuszna w sytuacji nasycenie organizacji w technologię IT było niskie, gdy użytkownicy posługiwali się prostymi systemami, a ich wykorzystanie w organizacji było dobrowolne – typowa sytuacji w końcu lat osiemdziesiątych. Powstaje pytanie na ile sytuacja powyższa jest w dalszym ciagu aktualna–współcześnie typowe jest raczej posługiwanie się aplikacjami wymagajacymi uprzedniego intensywnego szkolenia a do korzystania z której pracownik jest zobowiązany. Przykładami niech będą systemy zarządzania organizacją typu ERP czy systemy EDI/IOS pozwalające na wymianę danych pomiędzy organizacjami.

⁷Innymi słowy normy subiektywne i postawy w całości mediują wpływ zmiennych zewnętrznych.

w późniejszych wersjach modelu TAM postawy (ATT) zostały usunięte a czynniki PU/PEU determinują bezpośrednio intencje⁸.

TAM zakłada, że *zmienne zewnętrzne* (charakterystyka systemu IT, charakterystyka użytkownika, czynniki organizacyjne) wpływają *pośrednio* na intencje determinując wielkość PU/PEOU (Venkatesh i Davis, 2000, s. 187), przy czym wpływ czynnika PU jest znacząco większy niż PEOU. Ponadto wykorzystanie SI/TI może być przewidywane na podstawie intencji. Zamiarem Davisa było opracowanie "prostego, teoretycznie uzasadnionego modelu, zdolnego do objaśnienia czynników akceptacji systemów komputerowych w sposób ogólny, to jest dla różnych grup użytkowników końcowych i różnych rodzajów systemów [...] TAM ma stanowić teoretyczną podstawę objaśniającą w jaki sposób zewnętrzne czynniki wpływają na przekonania, postawy i intencje" (Davis, 1989, s. 985). Z uwagi na trudność z pomiarem użytkowania systemu wiele badań posługuje się modelem czteroskładnikowym zakładającym, że wysoka wartość BI determinuje wysoką wartość użytkowania, por. Taylor i Todd (1995)⁹.

Model TAM¹⁰ porównywano ze "zwykłym" modelem TPB nie zawierającym czynników pierwszego stopnia w którym postawa, normy subiektywne oraz kontrola behavioralna były mierzone bezpośrednio¹¹. "Zwykły" model TPB oraz model TAM zawierały po trzy czynniki objaśniające intencje oraz wykorzystanie systemu. Wykorzystanie TPB nie poprawiło prognostycznych właściwości modelu ponieważ oba wyjaśniały po 34% zmienności wykorzystania systemu. W modelu zdekomponowanym wszystkie czynniki pierwszego stopnia za wyjątkiem PEOU oraz kompatybilności okazały się istotne, zaś model ma nieznacznie lepsze właściwości prognostyczne ponieważ udział wyjaśnianej zmienności wykorzystania wzrósł do 36%. Taylor i Todd przyznają, że prognostyczne właściwości modelu poprawiły się w niewielkim stopniu, zwłaszcza uwzględniając wzrost złożoności modelu (13 zamiast 5 czynników) ale uważają, że model zdekomponowany lepiej objaśnia przyczyny wykorzystania SI/TI. Przykłady zastosowania modelu wykorzystującego teorię planowanego działania do określenia akceptacji SI/TI można także znaleźć w pracach Lin i inni (2006); Hung i Chang (2005); Hsu i Chiu (2004) oraz Mathieson (1991).

Model TAM był wielokrotnie wykorzystany w badaniu akceptacji różnych systemów komputerowych: aplikacje biurowe takie jak arkusze kalkulacyjne i edytory (Burton-Jones

⁸Por. przykładowo Igbaria i inni (1997); Burton-Jones i Hubona (2005); Venkatesh i Davis (2000).

⁹W przeglądzie 145 artykułów wykorzystujących TAM wskazuje się, że 43% badań nie zawierało relacji pomiędzy intencją a faktycznym wykorzystaniem Shumaila Y. Yousafzai (2007a).

¹⁰W związku z tym skala formatywna jest kontekstowa i powinna być adaptowana do potrzeb konkretnego badania. Przykładowo pytanie o dostęp do danych ma sens w środowisku bazodanowym, w którym użytkownicy mają – podyktowany względami bezpieczeństwa –ograniczony dostęp do danych. W przypadku aplikacji biurowej czy programu komunikacyjnego takie pytanie może nie mieć sensu.

¹¹Ponieważ skala pomiarowa obejmowała zaledwie 9 pytań więc można mieć poważne zastrzeżenia co do precyzji pomiaru (Taylor i Todd, 1995, s. 173–174).

i Hubona (2005); ?,?); Davis (1989); Davis i inni (1989); Deng i inni (2005); Burton-Jones i Hubona (2005)), technologie komunikacyjne, poczta elektroniczna Gefen i Straub (1997); Burton-Jones i Hubona (2005); Mathieson i inni (2001) systemy bazodanowe, aplikacje specjalistyczne Venkatesh i Davis (2000) systemy WWW Chau i Lai (2003); Słomka i inni (2007). Przegląd wykorzystania modelu TAM do badania akceptacji można znaleźć w Venkatesh i inni (2003); Shumaila Y. Yousafzai (2007a,b) oraz Han (2003). Kilka badań wykorzystujących omawiany model jest na tyle interesująca, że poniżej przedstawiamy je bardziej szczegółowo.

Zakończenie

Możliwości, jakie stoją przed archiwum prac magisterskich opartych na XML-u, są ograniczone jedynie czasem, jaki należy poświęcić na pełną implementacje 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.

Bibliografia

Adams, D. A., Nelson, R. R., i Todd, P. A. (1992). Perceived usefulness, ease of use, and usage of information technology: a replication. *MIS Q.*, 16(2):227–247.

Adobe Systems Inc. (1996). Portable Document Format Reference Manual Version 1.2. Addison-Wesley.

Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, 50(2):179–211.

Ajzen, I. (bdw). Theory of planned behavior. http://www.people.umass.edu/aizen/tpb. html.

Bandura, A. (1994). *Self-efficacy*, volume 4, pages 71–81. Academic Press, New York. http://www.des.emory.edu/mfp/BanEncy.html.

Beebe, N. H. F. (1995). Bibliography prettyprinting and syntax checking. *TUGBoat*, 14:395–419. http://www.gust.org.pl/PDF/BIUL/10/12html.pdf.

Bray, T., Paoli, J., i Sperberg-McQueen, C. M. (1998). Extensible markup language (XML) 1.0. http://www.w3.org/TR/1998/REC-xml-19980210.

Bryan, M. (1997). SGML and HTML explained. Addison-Wesley.

Burnard, L. i Sperberg-McQueen, C. M. (2002). TEI lite: An introduction to text encoding for intechange. http://www.tei-c.org/Lite/.

Burton-Jones, A. i Hubona, G. S. (2005). Individual differences and usage behaviour: Revisiting a technology acceptance model assumption. *Data Base*, 36(2):58–77.

Chau, P. Y. i Lai, V. S. (2003). An empirical investigation of the determinants of user acceptance of internet banking. *Journal of Organizational Computing and Electronic Commerce*, 13(2):123–145.

Clark, J. (1999). Xsl transformations (xslt) version 1.0. http://www.w3.org/TR/xslt.

Davis, F. D. (1989). Perceived usefulness, perceived easy of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340.

Davis, F. D., Bagozzi, R. P., i Warshaw, P. R. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Decision Science*, 35(8):982–1003.

Deng, X., Doll, W. J., Hendrickson, A. R., i Scazzero, J. A. (2005). A multi-group analysis of structural invariance: an illustration using the technology acceptance model. *Journal of Marketing*, 42(5):745–759.

Fishbein, M. i Ajzen, I. (1975). *Belief, Attitude, Intention, and Behaviour: An Introduction to Theory and Research*. Addison-Wesley.

Gefen, D. i Straub, D. W. (1997). Gender differences in the perception and use of e-mail: An extension to the technology acceptance. *MIS Quarterly*, 21(4):389–400.

Goldfarb, C. (1993). The SGML Handbook. Oxford University Press.

Han, S. (2003). Individual adoption of information systems in organizations: A literature review of technology acceptance model. Technical Report 540, Turku Center for Computer Science. http://www.tucs.fi/research/publications/insight.php?id=tHan03b&table=techreport.

Hsu, M.-H. i Chiu, C.-M. (2004). Internet self-efficacy and electronic service acceptance. *Decis. Support Syst.*, 38(3):369–381.

Hung, S.-Y. i Chang, C.-M. (2005). Wireless application protocol; technology acceptance model; theory of planned behavior; decomposed theory of planned behavior; information technology acceptance. *Computer Standards and Interfaces*, 27(4):359–370.

Igbaria, M., Zinatelli, N., Cragg, P., i Cavaye, A. L. M. (1997). Personal computing acceptance factors in small firms: A structural equation model. *MIS Quarterly*, 21(3):279–305.

Lamport, L. (1994). Lamport, L. (1994). Lamport, L. (1994). Lamport preparation system.

Lee, Y., Kozar, K. A., i Larsen, K. R. T. (2003). The technology acceptance model: Past, present and future. *Communications of AIS*, 12:752–780.

Lin, J., Chan, H. C., i Wei, K. K. (2006). Understanding competing application usage with the theory of planned behavior: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 57(10):1338–1349.

Maler, E. (1998). Sgml exceptions and xml. http://www.arbortext.com/Think_Tank/XML_Resources/SGML_Exceptions_and_XML.

Maler, E. i Andaloussi, J. E. (1996). *Developing SGML DTDs. From Text to Model to Markup*. Prentice Hall.

Mathieson, K. (1991). Predicting user intentions: Comparing the technology acceptance model with the theory of planned behavior. *Information Systems Research*, 2:173–191.

Mathieson, K., Peacock, E., i Chin, W. W. (2001). Extending the technology acceptance model: The influence of perceived user resources. *SIGMIS Database*, 32(3):86–112.

North, S. (2000). XML dla każdego. Helion.

Shumaila Y. Yousafzai, Gordon R. Foxall, J. G. P. (2007a). Technology acceptance: a meta-analysis of the TAM: Part 1. *Journal of Modelling in Management*, 2.

Shumaila Y. Yousafzai, Gordon R. Foxall, J. G. P. (2007b). Technology acceptance: a meta-analysis of the TAM: Part 2. *Journal of Modelling in Management*, 2.

Słomka, A., Przechlewski, T., i Wrycza, S. (2007). Examining oss sucess: Information technology acceptance by firefox users. In G., K., G.W., W., J., Z., i S., W., editors, *Advances in Information Systems Development: New Methods and Practice for the Networked Society*, pages 447–456.

Taylor, S. i Todd, P. A. (1995). Understanding information technology usage: A test of competing models. *Information Systems Research*, 6(2):144–176.

Van Herwijnen, E. (1990). Practical SGML. Kluwer Academic Publishers.

Venkatesh, V. i Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, 46(2):186–204.

Venkatesh, V., Morris, M. G., Davis, G. B., , i Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3):425–478.

Wall, L., Christiansen, T., i Schwartz, R. (1996). Programming Perl. O'Reilly & Associates.

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

Spis tabel

1.1.	Porównanie wielkości popularnych definicji typu dokumentu	7
3.1.	Publicznie dostępne procesory XLST	12

Spis rysunków

1.1.	Schemat archiwum	6
3.1.	Ogólna struktura dokumenu XSL-FO	13
4.1.	Model Akceptacji Technologii, opracowanie własne za Davis i inni (1989)	15

Skorowidz

DocBook, 7–8
edytor — biurowy, 8 — Emacs, 8 — strukturalny, 8 ESIS, 11
intencja, 14
kontrola behawioralna, 14, 15 koszty, 5
model — TAM, 15
normy subiektywne, 14, 15
parser, 8, 9, 11 postrzegana — łatwość użytkowania, 14 — użyteczność, 14
system XML, 5, 19
TEI, zob. Text Encoding Iniciative teoria — planowanego działania, 14 — uzasadnionego działania, 14 Text Encoding Iniciative, 7
użytkownik końcowy, 16
walidacja, 6 WYSIWYG, 8
Xalan, 12 XSLT, 7, 10, 12

Xt, 12

zebra, 18 zmienne — zewnętrzne, 15 źdźbło, 18 żółw, 18