

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



PREHĽAD A POROVNANIE CONTENT MANAGEMENT SYSTÉMOV
DIPLOMOVÁ PRÁCA

DIPLOMANT: MIROSLAV POMŠÁR
VEDÚCI DIPLOMOVEJ PRÁCE: RNDR. RICHARD OSTERTÁG

BRATISLAVA 2007

PREHLAD A POROVNANIE CONTENT MANAGEMENT SYSTÉMOV

DIPLOMOVÁ PRÁCA

MIROSLAV POMŠÁR

**UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

Informatika

**VEDÚCI DIPLOMOVEJ PRÁCE:
RNDR. RICHARD OSTERTÁG**

BRATISLAVA 2007

Abstrakt

Pojem *Content Management System* môžeme neformálne definovať ako aplikáciu umožňujúcu efektívnu správu *obsahu*. Pod pojmom *obsah* potom rozumieme akýkoľvek typ alebo časť digitálnej informácie. Správa obsahu zahŕňa funkcie ako definíciu, komunikáciu a uplatňovanie procesov pri vytváraní, respektíve prístupe k informáciám, ako aj definíciu rolí užívateľov v rámci systému. CMS aplikácie logicky oddeľujú rôzne aspekty obsahu napríklad prezentáciu, skutočný obsah, metadáta, prístupové práva, vďaka čomu sprístupňujú obsah v cielených publikáciách.

Nasadenie CMS dnes zvažuje mnoho inštitúcií a organizácií. Problémom je, že pole CMS je relatívne nové a málo prebádané. Kritickým faktorom pre úspešnú integráciu CMS v rámci organizácie je voľba správnej aplikácie, teda definovanie kritérií a vyžadovaných funkcií.

Cieľom práce je podanie prehľadu problematiky CMS a porovnanie vybraných opensource CMS riešení. Súčasťou práce sú tabuľky porovnania CMS podľa konkrétnych funkcií, ako aj analýza niektorých významných technológií.

Kľúčové slová: Content Management System, Weblog, Wiki, Hypertext, Metadáta, Kategorizácia, História modifikácií, Web feed, RDF, RSS, Workflow, Vyhľadávanie.

Čestne prehlasujem, že túto diplomovú prácu som
vypracoval samostatne len s použitím uvedenej literatúry.

Bratislava, Máj 2007

Ďakujem svojmu vedúcemu diplomovej práce, RNDr. Richardovi Ostertágovi za cenné rady a pripomienky pri písaní tejto práce.

Obsah

Úvod.....	8
1 Základné vlastnosti CMS aplikácií.....	10
1.1 Čo sú Content Management Systémy.....	10
1.2 Funkcionalita Content Management Systémov.....	11
1.2.1 Logická reprezentácia.....	11
1.2.2 Vstup obsahu.....	11
1.2.3 Publikácia a prezentácia obsahu.....	12
1.2.4 Správa rolí a procesov.....	12
1.2.5 Fyzická reprezentácia.....	12
1.3 Rozdelenie Content Management Systémov.....	12
1.3.1 Základné rozdelenie WCMS.....	14
1.4 Výber Content Management Systémov.....	16
2 Reprezentácia obsahu v CMS.....	19
2.1 Komponent Obsahu.....	19
2.2 Element.....	21
2.3 Vnútoraná štruktúra obsahu.....	22
2.4 Komplexné elementy.....	25
2.4.1 XHTML.....	25
2.4.2 Wiki.....	27
2.4.3 Štruktúrovaný text.....	29
2.4.4 EzXML.....	30
2.4.5 BBCode.....	31
2.4.6 DocBook.....	31
2.4.7 Zhrnutie.....	33
2.5 Vonkajšia štruktúra obsahu.....	34
2.5.1 Statická štruktúra obsahu.....	35
2.5.2 Dynamická štruktúra obsahu.....	36
3 Vstup Obsahu.....	42
3.1 Aplikácie pre vstup obsahu.....	42
3.1.1 Jednoduché webové formuláre.....	43
3.1.2 Komplexné webové formuláre.....	44
3.1.3 Problémy webových formulárov.....	48

3.1.4 Iné aplikácie.....	50
3.2 Multimediálny obsah.....	51
3.3 Súborovo orientované protokoly.....	52
3.3.1 FTP.....	52
3.3.2 WebDAV	52
3.3.3 CIFS.....	53
3.3.4 Zhrnutie.....	55
3.4 E-mail.....	57
3.5 Web-feed import.....	58
3.5.1 Zjednodušená schéma web-feed.....	59
3.5.2 Problémy web-feed.....	61
3.5.3 Jednorázový import obsahu	61
3.5.4 Priebežná integrácia obsahu.....	61
3.6 Prepojenia rôznych CMS.....	63
4 Formy prezentácie obsahu.....	65
4.1 Výber obsahu.....	65
4.2 Prevod reprezentácie obsahu.....	67
4.2.1 Šablónové systémy.....	67
4.3 Výstupný formát.....	69
4.3.1 Webová stránka.....	69
4.3.2 Ďalšie výstupné formáty.....	73
4.4 Doručenie publikácie.....	73
4.5 Internacionalizácia a lokalizácia.....	74
4.5.1 Internacionalizácia.....	74
4.5.2 Lokalizácia webového rozhrania CMS.....	76
4.5.3 Viacjazyčný obsah.....	77
4.6 Web-feed export.....	80
5 Porovnanie správy rolí a procesov.....	83
5.1 Bezpečnosť.....	83
5.1.1 Kontrola prístupu.....	83
5.1.2 Autentifikácia a vytváranie nových užívateľov.....	91
5.1.3 Audit.....	92
5.2 Správa procesov.....	94
5.2.1 Jednoduché workflow.....	96

5.2.2 Komplexné workflow.....	96
5.2.3 Alternatívne možnosti správy obsahu.....	99
5.3 Spolupráca pri vytváraní obsahu.....	101
5.3.1 Súčasná modifikácia obsahu	101
5.3.2 História modifikácií.....	103
6 Riešenie ukladania, porovnanie fyzickej reprezentácie.....	105
6.1 Vyhľadávanie.....	105
6.2 Fyzická reprezentácia.....	107
7 Záverečný prehľad.....	109
7.1 Rozdelenie CMS.....	109
7.1.1 CMS ako nástroj pre vytvorenie webového sídla.....	109
7.1.2 CMS ako online periodiká.....	110
7.1.3 CMS ako kolaboračná platforma.....	110
7.2 Subjektívne hodnotenie.....	111
8 Záver.....	113
9 Literatúra.....	114

Úvod

Pojem *Content Management System* môžeme neformálne definovať ako aplikáciu umožňujúcu efektívnu správu *obsahu*. Pod pojmom *obsah* potom rozumieme akýkoľvek typ alebo časť digitálnej informácie. Správa obsahu zahŕňa funkcie ako definíciu, komunikáciu a uplatňovanie procesov pri vytváraní, respektíve prístupe k informáciám, ako aj definíciu rolí užívateľov v rámci systému. CMS aplikácie logicky oddeľujú rôzne aspekty obsahu napríklad prezentáciu, skutočný obsah, metadáta, prístupové práva, vďaka čomu sprístupňujú obsah v cieľných publikáciách. Ďalej implementujú stratégie ukladania, indexovania a vyhľadávania.

Pole CMS je relatívne nové. Tento pojem zastrešuje niekoľko technológií, ktoré vznikali spočiatku úplne oddelene. Na jednej strane to bola snaha inštitúcií, firiem či jednotlivcov o prezentáciu na internete, na druhej strane potreba centrálného spravovania digitálnych dokumentov, dát a iných informácií. Spočiatku boli všetky úlohy, ktoré zahŕňa správa informácií, riešené samostatnými aplikáciami. Častými problémami boli nekonzistentný dizajn, stratené alebo neaktuálne údaje, nedostatok či neexistencia možnosti kontroly ako aj malá flexibilita v prípade zmien. Keďže prezentácia na internete je dnes nevyhnutnosťou a súčasne webové rozhranie poskytuje veľmi jednoduchý a dostupný prístup k informáciám, je pochopiteľné, že vznikajú nové CMS a existujúce sa neustále zlepšujú. Podobne, kým web dvadsiateho storočia bol charakteristický vysoko manuálnym prístupom a prezentácia na internete pozostávala z niekoľkých jednoduchých stránok, súčasnosť ovládajú aplikácie, automatizujúce mnohé úlohy, pričom prezentácia obsahu na internete je len jedna z nich.

Presadzuje sa trend využitia webového rozhrania nielen ako prezentačného ale aj aktívneho ovládacieho prvku. Zrejme pre tvorbu multimediálneho obsahu ako zvuk a video je webové rozhranie nedostatočné a nepraktické, na druhej strane je postačujúce pre tvorbu textového obsahu. Podobne možno použiť webové rozhranie i pre správu iných aspektov CMS ako koordináciu užívateľov, vytváranie rolí. Má aj ďalšie výhody ako prenositeľnosť.

Keďže efektívne vyhľadávanie je nevyhnutnosťou, CMS implementujú indexovanie a správu metadát. Pokročilejšie aplikácie umožňujú vytvoriť logický systém prepojení a jednoduché až automatické začlenenie a kategorizáciu obsahu.

Dôraz sa kladie na automatizáciu, preto je možné v CMS aplikáciách navrhnuť časový plán udalostí. V najjednoduchšej forme ide o uverejnenie obsahu v určený čas. Zložitejšie implementácie umožňujú formalizovať procesy ako napríklad kedy ma byť publikovaný nový článok, kto pracuje na texte, kto je zodpovedný za grafiku, kto a kedy skontroluje článok.

Cieľom tejto práce je porovnanie opensource CMS riešení. Tento výber ovplyvnila na jednej strane vysoká kvalita opensource CMS ako aj ich popularita. V neposlednom rade bol tento výber ovplyvnený aj možnosťou nahliadnutia do zdrojového kódu.

V práci predpokladáme, že čitateľ má základné znalosti z oblasti webových technológií ako HTTP, HTML, XML, ECMAScript ako aj znalosti z oblasti relačných databáz. Určená je tak pre ľudí, ktorí uvažujú o nasadení CMS, ako aj pre autorov CMS systémov.

V prvej kapitole sa venujeme popisu vlastností CMS a základným definíciám. Jej cieľom je aj predstavenie CMS, ktoré porovnávame.

Druhá kapitola je zameraná na porovnanie CMS podľa reprezentácie obsahu v rámci CMS. Uvádzame prehľad používaných prístupov a metód pre definíciu štruktúry.

Cieľom tretej kapitoly je porovnanie CMS s ohľadom na vytváranie obsahu.

Dôležitým kritériom v tomto ohľade budú nástroje a techniky pre vstup a vytváranie obsahu.

Štvrtá kapitola je zameraná na prehľad metód prezentácie obsahu, výstupných formátov a médií. Okrem toho sa venujeme aj problematike internacionalizácie.

Zameranie piatej kapitoly je porovnanie vzhľadom kritéria kontroly prístupu, definovanie rolí a procesov.

Vo šiestej kapitole sa zaoberáme najnižšou úrovňou CMS, uložením, fyzickou reprezentáciou obsahu a vyhľadávaním.

V záverečnej siedmej kapitole uvádzame rozdelenie CMS podľa zamerania. Súčasťou je rozdelenie porovnávaných CMS do zadaných skupín.

1 Základné vlastnosti CMS aplikácií

1.1 Čo sú Content Management Systémy

CMS je aplikácia určená na správu obsahu. Tato definícia nieje veľmi užitočná, skúsme teda detailnejší pohľad.

Definícia: Pod *obsahom* chápeme základný stavebný prvok informácie, štruktúrovaný text, informácia o nadchádzajúcej udalosti, obrázok, video, dokument, jednoducho akákoľvek informácia, ktorá má pre organizáciu využívajúcu CMS význam. Všetky tieto prvky chápeme ako obsah, pretože zdieľajú podobné atribúty.

Dôležité aspekty obsahu sú

- *štruktúra* – definuje vzťahy medzi logickými časťami konkrétneho obsahu (*vnútorná štruktúra*), respektíve medzi rôznymi obsahmi (*vonkajšia štruktúra*). Príkladom vnútornej štruktúry je napríklad nadpis, odsek, vonkajšej napríklad kategória.
- *formátovanie* – definuje prezentáciu obsahu, napríklad veľkosť, druh a farba písma.

Definícia: *Metadáta* sú informácie, ktoré sú určitého typu a sú v určitom vzťahu k obsahu. Predstavujú mechanicky spracovateľný popis vlastností obsahu, teda trojice atribút, typ, hodnota, napríklad (dátum poslednej zmeny, dátum, 12.12.2006).

Definícia: Základnou jednotkou správy obsahu je *komponent*. Komponent sa skladá z obsahových a metadáta *elementov*. Elementom môže byť odsek textu, alebo celý dokument, v závislosti od konkrétneho CMS, respektíve užívateľských potrieb.

Komponent predstavuje najmenší objekt, ktorý CMS spravuje. Obsah je možné vytvárať len ako celý komponent, podobne pri archivácii a vyhľadávaní uvažujeme v rámci komponentov.

Element je najmenším *adresovateľným* objektom CMS. Často pri publikácii využívame len niekoľko elementov. Súčasne je možné využiť elementy ako kľúč k nájdeniu konkrétnych komponentov.

Definícia: Druh, počet, poradie a iné atribúty elementov definuje *typ komponentu*. V zmysle objektovo orientovaného programovania chápeme *komponent* ako inštanciu *typu*.

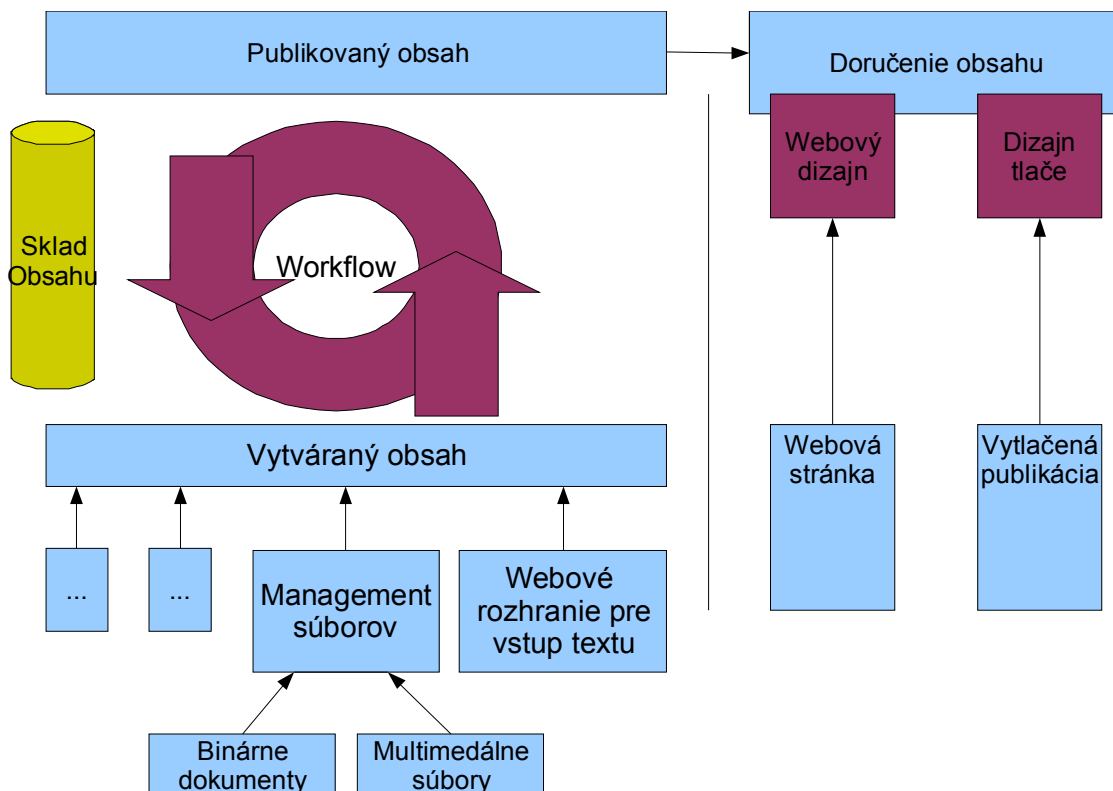
Definícia: K obsahu neprístupujú užívatelia priamo, ale CMS poskytuje oprávneným užívateľom logický pohľad na obsah, ktorý označujeme ako *prezentáciu* obsahu. Jednotkou prezentácie je publikácia, nemusí obsahovať všetky elementy konkrétneho komponentu, respektíve sa môže skladať z niekoľkých elementov rôznych komponentov obsahu.

Neformálne môžeme systém *workflow* definovať ako časť CMS, ktorá kontroluje atribúty obsahu ako vytváranie a upravovanie užívateľmi a byť publikované určitým spôsobom. Systém workflow teda predstavuje množinu pravidiel, určujúcu spôsob interakcie medzi obsahom a užívateľmi. Formálnu definíciu uvádzame v tretej kapitole.

Definícia: *CMS* je nástroj, umožňujúci skupine (technickým aj netechnickým) užívateľov vytvárať, upravovať, spravovať a následne publikovať rôzne druhy obsahu. Pritom postupujú

podľa centralizovanej množiny formálnych pravidiel a procesov, ktoré zaručujú koherentnú a správnu prezentáciu.

1.2 Funkcionalita Content Management Systémov



Obrázok 1: CMS Diagram

1.2.1 Logická reprezentácia

Systém definuje logické typy obsahu teda základné stavebné bloky. Vnútna štruktúra obsahu hovorí o tom z akých častí sa obsah skladá, kým vonkajšia štruktúra umožňuje prepojiť obsahy do vyšších celkov.

1.2.2 Vstup obsahu

Systém zahŕňa užívateľov, nástroje a procedúry na zhromažďovanie obsahu. Poskytuje redakčný ako aj metadáta rámec. Proces vstupu informácií do CMS môžeme rozdeliť do nasledujúcich kategórií.

Vytváranie nového obsahu. Súčasťou CMS je nástroj pre vstup obsahu. Táto metóda je vhodná pre textový obsah, multimedálny obsah sa vytvára špecializovanými aplikáciami.

Agregácia obsahu. Ide o vytváranie nového obsahu z existujúcich obsahov, s využitím vyhľadávacích alebo workflow funkcií CMS. Podobne do tejto kategórie patrí možnosť vkladania existujúceho obsahu, a to tak textového ako aj multimedálneho obsahu.

Redakčné a metadáta funkcie, zahŕňajú nástroje na komunikáciu a kolaboráciu užívateľov ako aj obohatenie obsahu v systéme o metadáta. Obsah je začlenený do určeného rámca po stránke formátu, štýlu a iných definovaných atribútov.

1.2.3 Publikácia a prezentácia obsahu

Je proces, pri ktorom je obsah vyňatý zo skladu a prezentovaný podľa požiadaviek cieľovej publikácie. Zahŕňa doručenie publikácie ako aj konverziu obsahu do správneho výstupného formátu. CMS poskytujú základnú množinu prezentácií (webová stránka), často dovoľujú vytvárať nové prezentácie zmenou parametrov základných. Z tohto pohľadu je prezentácia „funkcia“ z množín obsahu a doménových parametrov do množiny publikácií. Doménové parametre sú napríklad HTML alebo XML konštrukcie a pod prvkom množiny publikácií chápeme, postupnosť „častí“ obsahu a doménových parametrov.

1.2.4 Správa rolí a procesov

Zameriava sa na dosahovanie bezpečnosti a efektívnosti procesu vytvárania, modifikácie a publikácie obsahu. K dosahovaniu bezpečnosti slúžia nástroje k definícií kontroly prístupu a sledovania akcií. O efektívnosť a priradenie zodpovednosti sa stará systém workflow.

1.2.5 Fyzická reprezentácia

Predstavuje tak sklad obsahu, metadát ako aj nástroje na ich efektívnu správu. Často sklad spravuje aj ostatné informácie v systéme ako užívateľov, roly pod ktorými vystupujú. V sofistikovanejších CMS aplikáciach aj definície procesov. Všetky ostatné funkčné časti CMS by nemali spravovať vlastné informácie, ale spoliehať sa na centrálny sklad. Táto funkčná jednotka sprostredkuje dátový tok medzi zvyšnými časťami CMS aplikácie.

1.3 Rozdelenie Content Management Systémov

Pojem CMS zahŕňa množinu technológií, ktoré si teraz bližšie predstavíme. Pozrime sa ako môžeme rozdeliť CMS aplikácie podľa druhu spravovaného obsahu.

DMS document management system je CMS, v ktorom sa kladie obmedzenie na druh obsahu. Dokumentom rozumieme ucelený obsah s vlastnou vnútornou štruktúrou. Dokumenty sú vytvárané špecializovanými aplikáciami, ktoré bývajú do rôznej miery integrované s DMS, alebo sú digitalizované z pôvodnej papierovej formy s využitím OCR nástrojov. Dôležité sú textovo orientované funkcie ako vyhľadávanie a podpora viacerých jazykov.

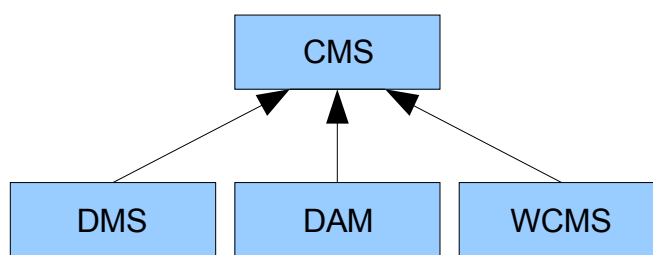
DAM digital asset management, sú nástroje, ktoré podobne ako v prípade DMS spravujú existujúci obsah s previazaním na špeciálne aplikácie na ich vytváranie. V tomto prípade ide hlavne o multimedialny obsah. Medzi základné funkcie patrí automatické transformovanie do požadovaného výstupného formátu, napríklad zníženie kvality videa pre pomalší kanál, alebo farebné konverzie. Keďže multimedialny obsah vyžaduje veľa miesta na

uloženie, k základným vlastnostiam patrí aj schopnosť efektívnej manipulácie s rozsiahlym obsahom. Medzi funkcie DAM patrí i ochrana autorských práv, použitím DRM alebo elektronickej formy vodotlače.

WCMS web content management system sú aplikácie, ktoré spravujú obsah, vytváraný a publikovaný cez webové rozhranie. Poskytujú nástroje pre vytváranie obsahu ako aj nástroje pre definíciu prezentácie. Umožňujú vytváranie štruktúr a prepojení medzi obsahmi. Implementujú striktné oddelenie obsahu od prezentácie.

Tabuľka 1: Druhy CMS

	DMS	DAM	WCMS
Integrované vytváranie obsahu	Nie	Nie	Áno
Integrácia so špecializovanými aplikáciami	Čiastočne	Áno	Nie
Oddelenie prezentácie od obsahu	Nie	Nie	Áno
Veľkosť obsahu	Veľká	Veľká	Malá
Štruktúry a prepojenia medzi obsahmi	Nie	Nie	Áno
Transformácia obsahu	Nie	Áno	Áno
Workflow	Áno	Áno	V obmedzenej forme
Integrovaná kontrola použitia obsahu	Áno	Áno	Nie
Druh obsahu	Vytlačený text. Súbory kancelárskych aplikácií (MS Word, OpenOffice Writer).	Obrázky, CAD, 3-D, Video, Audio, Macromedia Flash	Text, HTML, Obrázky
Hlavné úlohy	Centrálny sklad, Vyhľadávanie, Workflow	Kontrola prístupu, Publikácia	Vytváranie, Agregácia, Publikácia



Obrázok 2: Druhy CMS

Slovo enterprise v **Enterprise CMS**, poukazuje na množinu funkcií a nástrojov v rámci CMS s ohľadom na efektívne modelovanie business procesov v rámci organizácie. Dôraz je kladený na integráciu. EMS obsahujú DMS, ako aj WCMS pre vnútorné a vonkajšie použitie. EMS sa zvykne označovať ako *disciplina* a nie konkrétna aplikácia, ide teda o spôsob správy obsahu v rámci celej organizácie.

1.3.1 Základné rozdelenie WCMS

Podľa **interakcie** užívateľov s obsahom rozdeľujeme WCMS na:

- **Producent – Konzument.**
 - Relatívne obmedzená skupina užívateľov vytvára obsah, kým skupina konzumentov je rozsiahla.
 - Užívatelia sú rozdelení do skupín s rôznymi právomocami.
 - Obsah je vytváraný podľa určeného plánu.
 - Spravidla existuje špeciálne rozhranie pre konzumentov a producentov, pričom tieto rozhrania sú
 - jednou funkčnou jednotkou (jedna aplikácia, resp. webový server)
 - rozhrania sú fyzicky rozdelené (najčastejšie jedno rozhranie pre producentov, a konfigurovateľný počet rozhraní pre konzumentov)
 - CMS predstavuje iba rozhranie pre producentov. Obsah je publikovaný vo forme súborov a jeho prezentácia je úlohou samostatnej aplikácie.
- **Kolaboračné.**
 - V tomto prípade môžu vytvárať obsah všetci (registrovaní) užívatelia.
 - Obsah môžu vytvárať nezávisle na sebe a bez spoločného plánovania.
 - Existuje jediné rozhranie, a CMS tvorí jediná aplikácia.

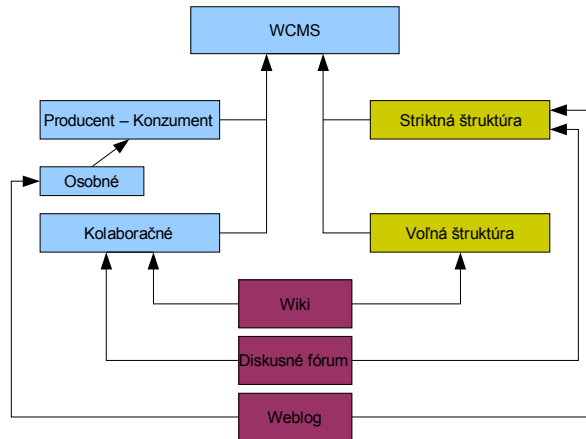
Podľa miery kontroly nad **vnútornou štruktúrou** obsahu

- Bez vnútornej štruktúry. Komponent sa skladá z jediného obsahového elementu, ktorý je z pohľadu CMS jednoliatou informáciou.
- So štruktúrou.
 - Štruktúra obsahu je definovaná pomocou špeciálneho jazyka.
 - Obsahu je priradený typ, ktorý určuje z akých častí sa skladá.
 - Pokročilejšie CMS umožňujú vytvárať nové typy obsahu, a to pomocou špeciálneho jazyka (napríklad *XML schema*), alebo nástroja.

Tieto možnosti, nie sú exkluzívne. Napríklad špeciálny jazyk môže určovať časti obsahu.

Vzhľadom na **vonkajšiu štruktúru** obsahu.

- CMS striktno určuje vonkajšiu štruktúru obsahu. Obsah je možné vytvárať iba na určitom mieste v štruktúre alebo je implicitne zaradený do štruktúry.
- CMS dovoľuje do určitej miery voľne vytvárať vonkajšiu štruktúru, ako aj zaradiť obsah do niekoľkých nezávislých štruktúr.



Obrázok 3: WCMS

Pre niektoré druhy WCMS sa zaužívali špeciálne označenia.

Definícia: Diskusné fórum je kolaboračný druh WCMS, v ktorom je obsah možné vytvárať len v rámci štruktúry – diskusie. Diskusné fórum je prakticky webovou paralelou emailového zoznamu (*mailing list*).

Definícia: Diskusia chápeme ako druh vonkajšej štruktúry obsahu, kde prvý obsah (*topic – téma*) udáva tému, pričom ďalšie obsahy predstavujú buď reakciu na prvý, predchádzajúci, alebo ktorýkoľvek obsah. Z tohto pohľadu štruktúra vytvára plytký strom, vlákno (*thread*) alebo všeobecný strom (niekedy sa označuje ako vlákno celý strom, alebo cesta v strome).

Definícia: Weblog, blog je WCMS, v ktorom je obsah publikovaný v chronologickom poradí. Obsah je od publikácie nemenný. Najčastejšie existuje jediný producent obsahu.

Definícia: Wiki predstavuje kolaboračný druh WCMS, ktorý je založený na týchto princípoch:

1. Obsah je text, najmenšou jednotkou je wiki-stránka.
2. Existuje množina syntaktických konštrukcií označovaná ako wiki jazyk. Syntaktické konštrukcie sa používajú priamo v textovom obsahu.
3. Jednoduché vytváranie prepojení medzi wiki-stránkami.
4. Správa verzií. Konkrétnu wiki-stránku je možné vrátiť do podoby ktorejkoľvek predchádzajúcej verzie.

Skutočné WCMS aplikácie implementujú podmnožinu z hore uvedených technológií. Často WCMS obsahuje súčasne wiki ako aj diskusné fórum, alebo poskytujú nástroje pre vstup alebo výstup dokumentov (v zmysle definície DMS), či multimediálneho obsahu. Z tohto dôvodu je rozšírené použitie termínu CMS ako synonyma WCMS.

1.4 Výber Content Management Systémov

V tejto práci predstavíme techniky a nástroje, ktoré CMS používajú k dosahovaniu určených cieľov. Súčasne budeme porovnávať implementáciu nástrojov vo vybraných CMS. V práci sa obmedzíme na tieto opensource riešenia:

Bricolage 2.10	
Populárne stránky	<ul style="list-style-type: none"> • http://www.bricolage.cc/about/sites/ • http://www.macworld.com • http://www.who.int/en/
Dokumentácia	Online: http://www.bricolage.cc/docs , www.perl.com Komunita: http://www.bricolage.cc/support/lists/ Knihy: <ul style="list-style-type: none"> • Rolsky D., Williams K.: <i>The Mason Book</i>
Licencia	Modifikovaná BSD

Drupal 5.0	
Populárne stránky	<ul style="list-style-type: none"> • http://www.mtv.co.uk/ • www.ubuntu.com • www.linuxjournal.com • http://drupalsites.net/ • demo: http://opensourcecms.com/index.php?option=content&task=view&id=132
Dokumentácia	Online: drupal.org/handbooks Komunita: drupal.org/forum Knihy: <ul style="list-style-type: none"> • Mansfield N.: <i>Practical Drupal: Evaluating and Using a Web Content Management System</i> • VanDyk J., Westgate M.: <i>Pro Drupal Development</i>
Licencia	GPL[36]

eZ publish 3.9	
Populárne stránky	<ul style="list-style-type: none"> • http://ez.no/customers/references • http://jhmri.jhsph.edu/ • demo: http://www.opensourcecms.com/index.php?option=content&task=view&id=181&Itemid=159
Dokumentácia	Online: http://ez.no/doc/ Komunita: http://ez.no/community/forum Knihy: <i>Learning eZ publish 3: Building Content Management Solutions</i>
Licencia	GPL

Magnolia 3.0	
Populárne stránky	<ul style="list-style-type: none"> • www.map.es • http://www.magnolia.info/en/about-magnolia/reference-sites.html • demo: http://www.magnolia.info/en/about-magnolia/demo.html
Dokumentácia	Online: http://www.magnolia.info/wiki/ Komunita: http://www.magnolia.info/en/services/developer-support.html
Licencia	GPL

Mambo 4.5.2 / Joomla 1.0.12	
Populárne stránky	<ul style="list-style-type: none"> • http://www.mamboserver.com/index.php?option=com_content&task=view&id=94&Itemid=123 • www.desman.com • demo: http://www.mamboserver.com/index.php?option=com_content&task=view&id=35&Itemid=116
Dokumentácia	Online: help.mamboserver.com Komunita: forums.mamboserver.com Knihy: <ul style="list-style-type: none"> • Pirtle M.: <i>The Definitive Guide to Mambo</i> • Graf H.: <i>Building Websites with Mambo</i>
Licencia	GPL

MediaWiki 1.9	
Populárne stránky	<ul style="list-style-type: none"> • http://www.wikipedia.org
Dokumentácia	Online: http://www.mediawiki.org/wiki/Help:Contents Komunita: www.mwusers.com Knihy: <ul style="list-style-type: none"> • Rahman M.: <i>MediaWiki Administrators' Tutorial Guide</i>
Licencia	GPL

MoinMoin 1.5	
Populárne stránky	<ul style="list-style-type: none"> • https://help.ubuntu.com/community/ • http://wiki.debian.org/ • http://wiki.apache.org/general/
Dokumentácia	Online: http://moinmoin.wikiwikiweb.de/HelpContents Komunita: http://moinmoin.wikiwikiweb.de/MoinMoinMailingLists
Licencia	GPL

Plone 2.9	
Populárne stránky	<ul style="list-style-type: none"> • http://plone.org/about/sites • www.desman.com • demo: http://plonehostingdemo.nidelven-it.no/
Dokumentácia	Online: http://plone.org/documentation , http://api.plone.org/ Komunita: http://plone.org/contact Knihy: <ul style="list-style-type: none"> • McKay A.: <i>The Definitive Guide to Plone</i> [http://docs.neuroinf.de/PloneBook] • Pelletier M., Shariff M. : <i>Plone Live</i>
Licencia	GPL

TikiWiki 1.9	
Populárne stránky	<ul style="list-style-type: none"> • http://wiki.kde.org/ • http://www.aspdeveloper.net/tiki-index.php
Dokumentácia	Online: http://doc.tikiwiki.org/tiki-index.php Komunita: http://tikiwiki.org
Licencia	LGPL [37]

Typo3 4.0	
Populárne stránky	<ul style="list-style-type: none"> • http://www.typo3.com/References.1249.0.html • http://www.dfhec.harvard.edu • demo:http://www.opensourcecms.com/index.php?option=content&task=view&id=181&Itemid=159
Dokumentácia	Online: http://typo3.org/documentation/ Komunita: http://typo3.org/documentation/mailling-lists Knihy: Fritz R. A kol.: <i>TYPO3: Enterprise Content Management</i>
Licencia	GPL

Wordpress 2.0	
Populárne stránky	<ul style="list-style-type: none"> • blogs.zdnet.com • http://www.markshuttleworth.com/ • http://wordpress.com/ • demo:http://opensourcecms.com/index.php?option=com_content&task=view&id=2143
Dokumentácia	Online: http://codex.wordpress.org/Main_Page Komunita: http://wordpress.org/support/ Knihy: Langer M., Jordan M.: <i>WordPress 2 (Visual QuickStart Guide)</i>
Licencia	GPL

V základnom prevedení sú tieto CMS poskytované bezplatne, doplňujúce služby či rozšírenie funkcionality môže byť spoplatnené. Každá aplikácia je podporovaná niekedy hneď niekoľkými poradenskými a hostingovými spoločnosťami.

Hlavným kritériom výberu bola populárnosť riešenia, zdravá programátorská a užívateľská komunita ako aj dostupnosť dokumentácie a možnosť nahliadnutia do zdrojového kódu. Súčasne sme sa snažili výberom pokryť široké spektrum CMS.

2 Reprezentácia obsahu v CMS

Ako vyzerá obsah, ktorý CMS spravujú. Aké druhy obsahu existujú a aké sú ich vzájomné vzťahy. V tejto kapitole sa pokúsime odpovedať na predchádzajúce otázky.

2.1 Komponent Obsahu

Z definície komponentu obsahu vieme, že sa skladá z obsahových a metadáta elementov. Definícia však neuvádza o aké druhy elementov ide, ani v akom poradí, respektíve v akom počte je možné daný element v danom komponente použiť. Dôvod je jednoduchý, neexistuje nijaký štandard, pre definíciu komponentov, ktorý by bol univerzálne podporovaný. Existuje však niekoľko spoločných vlastností, podľa ktorých môžeme CMS porovnávať.

Na najvyššej úrovni rozdeľujeme CMS na dva druhy podľa toho ako je komponent definovaný.

1. Komponent obsahu má typ, ktorý určuje výber a poradie elementov komponentu. Každý typ elementu má priradenú kardinalitu, a to buď **jednoduchú**, 0-1 teda nepovinný respektíve povinný element alebo **komplexnú** 0-X (kde X je premenná). Pod kardinalitou chápeme povolený počet výskytov konkrétneho elementu v konkrétnom komponente.

Existujú dve základné podskupiny:

- **Lineárny zoznam**, typ je určený zoznamom povolených typov elementov.
- **Hierarchia**, elementy sa sami skladajú z elementov.

Ide vlastne o analógiu objektovo orientovaného programovania. Typ komponentu je trieda (*class* v jazyku JAVA) pričom sa skladá z vnorených tried elementov. Komponent je potom inštanciou svojho typu, pričom jednotlivé elementy sú rekurzívne inštanciou svojich tried.

2. Komponent nemá typ, je síce vytváraný z elementov, avšak CMS nedefinuje nijaké obmedzenia pre poradie alebo výber elementov. Obsahový komponent pozostáva z ľubovoľného počtu elementov, v ľubovoľnom poradí. Z pravidla CMS dovoľujú obmedziť druhy elementov avšak na odlišnej úrovni: v Typo3 je jedným z atribútov užívateľských rolí práve množina povolených elementov.

V jednoduchších CMS 1. typu sú typy preddefinované, nieje možné pridávať nové typy, ani modifikovať existujúce.

Ďalším kritériom je ako sú komponenty v CMS spravované.

1. **Homogénne** – všetky komponenty sú z hľadiska správy rovnocenné, poskytujú rovnakú základnú množinu akcií, môžeme ich zaradiť do rovnakých typov vonkajších štruktúr, pri vytváraní publikácií používame rovnaké nástroje.
2. **Nehomogénne** – existujú rôzne druhy komponentov, pričom každý je spravovaný odlišným spôsobom.

Tabuľka 2: Komponenty obsahu

CMS	Štruktúra komponentov	Vytváranie typov	Kardinalita	
Bricolage	Hierarchia	Áno	Jednoduchá, 0-neobmedzený počet	Elementy sú rozdelené na dva druhy field a element. Field sú v hierarchii listy, elementy sú uzly. Kardinalitu je možné definovať len pre field.
Drupal	Zoznam	Áno	Jednoduchá, Kontrolovaná konkrétnym typom elementu	Modul CCK Komplexná kardinalita závisí od typu elementu, vo všeobecnosti je to však 0 až neobmedzený počet.
ezPublish	Zoznam	Áno	Jednoduchá	
Plone	Zoznam	Áno	Jednoduchá	Modul Archetypes, Komplexná kardinalita elementov ArrayField ako aj štruktúra elementov CompoundField je vo vývoji [39].
Mambo	Preddefinovaná	Nie		
MediaWiki	Preddefinovaná	Nie		
MoinMoin	Preddefinovaná	Nie		
Tikiwiki	Preddefinovaná	Nie		
Wordpress	Preddefinovaná	Nie		
Magnolia	Voľná	Nie		
Typo3	Voľná	Nie		

Pre definíciu typu komponentu poskytujú CMS tieto možnosti:

1. Interaktívnu aplikáciu, najčastejšie vo forme webových formulárov.
2. Špeciálny jazyk pre definíciu typu.
 1. Vlastný jazyk, špecifický pre CMS.
 2. Otvorený štandard, napríklad XML Schema alebo RDF/XML.

Tabuľka 3: Definícia typu komponentu

CMS	Webový formulár	Jazyk	Poznámky
Bricolage	Áno	Vlastný XML	Export a import definícií pomocou dodávanej aplikácie bric_soap. 1)
ezPublish	Áno	Vlastný	
Drupal	Áno	Vlastný	Umožňuje export a import definícií.
Plone	Nie	Vlastný, UML	Je možné vytvoriť typ komponentu (respektíve produkt s niekoľkými typmi) v UML a pomocou ArchGenUML ho pridať do Plone. ArchGenUML. 2)

bric_soap je aplikácia určená na správu CMS z príkazového riadku. Import, export a modifikácia typov komponentov predstavuje iba časť funkcionality.

Výpis 1: Export všetkých typov obsahu v Bricolage

```
1 ./bric_soap element_type list_ids | ./bric_soap element_type export -
```

ArchGenUML, alebo *ArchGenXML* je aplikácia, ktorá spracuje UML popis vo formáte *xmi* a vygeneruje definíciu typu komponentu. Na vytvorenie UML diagramu vo formát *xmi* je možné použiť aplikácie *ArgoUML*, *Poseidon*, *ObjectDomain*. Viac sa dozvie čitateľ v [39], [40].

2.2 Element

V predchádzajúcej kapitole sme hovorili o elementoch komponentov vo všeobecnosti, v tejto kapitole sa pozrieme na elementy bližšie. Podobne ako v prípade typu komponentu, existuje typ elementu. Navyše má element i ďalšie atribúty, napríklad už spomenutú kardinalitu. Elementy môžeme rozdeliť do dvoch základných skupín, elementy obsahu a metadata elementy. Obsahové elementy reprezentujú samotný obsah komponentu, kým metadata elementy slúžia k popisu komponentu, teda napríklad dátum vytvorenia, meno autora.

CMS dodávajú štandardne množinu **základných typov** elementov vid' tabuľka 4. Pri definícii typu komponentu, však nepoužívame tieto typy priamo, reálne typy elementov od nich odvodíme. Poznamenajme, že nové základné typy je nutné dodať ako špecifický modul pre konkrétne CMS.

Typ elementu sa skladá z nasledujúcich častí:

- Meno alebo identifikátor elementu.
- Dátový typ, prevzatý od základného typu (tj. zdedený od rodičovského typu).
- Atribúty: kardinalita, použitie vo vyhľadávaní, zoznam povolených hodnôt, ...
- Metódy: validácia, vstup, výstup, ...
- Aplikácia: najčastejšie druh webového formulára, napríklad textový blok, zoznam hodnôt, alebo, kalendárová javascript aplikácia na výber dátumu pomocou myši.

K definícii nového typ elementu využívame jednu z dvoch nasledujúcich možností:

- **Reštrikcia:** zdedený typ je obmedzený, napríklad na niekoľko možných hodnôt, uvedením hornej a dolnej hranice (pre dátum), uvedením dĺžky (pre text).
- **Kompozícia:** nový typ je definovaný ako zoznam existujúcich typov (v závislosti od CMS môže byť typ definovaný priamo alebo nepriamo rekurzívne). Pre úplnosť dodajme, že kompozícia je predpokladom pre hierarchickú definíciu typu komponentu.

Metódy sú najčastejšie prevzaté zo základného typu, avšak všetky CMS v našom prehľade ich dovoľujú rozšíriť prípadne nahradiť (v zmysle vytvoriť funkciu v programovacím jazyku CMS).

Ďalšou vlastnosťou, ktorá ma korene v OOP je možnosť využiť existujúci definovaný typ v ktoromkoľvek type komponentu. Plone ide v tomto smere ešte ďalej, komponent môže byť poskladaný z existujúcich typov komponentov (čo môžeme chápať ako viacnásobne dedenie v rámci typov komponentov).

Tabuľka 4: Základné typy elementov

Základný typ elementu	Bricolage	ezPublish	Drupal	Magnolia	Plone	Typo3
Boolean	X	X	X		X	
Integer	X	X	X		X	
Float	X	X	X		X	
Text	X	X	X	X	X	X
HTML	X		X	X	X	X
XML		X				
Dátum	X	X	Modul Date		X	
Obrázok			Modul ImageField	X	X	X
Tabuľka				X		X
Odkaz na komponent	X	X	X	X	X	X
Citácia	X			X		
ISBN		X				
Cena		X				
Autor		X				
Email		X				
URL		X		X		X
Produkt		X				

2.3 Vnútoraná štruktúra obsahu

Pomocou typu komponentu je možné formalizovať štruktúru obsahu. Jedným s problémov je však granularita. Ako vidíme v tabuľke 4 poskytujú CMS elementy pre HTML resp. XML obsah. Na druhej strane umožňuje napríklad Bricolage kontrolu až na úrovni paragrafov, v Typo3 máme k dispozícii typ tabuľka. Paragrafy a tabuľky však môžeme vytvárať aj priamo v rámci HTML elementu.

Akú granularitu CMS poskytujú pri definícii typu komponentu si ukážeme na príklade. Pokúsime sa definovať typ komponentu diplomová práca.

Diplomová práca sa skladá z elementov:

1. Názov
2. Autor
3. Konzultant
4. Kľúčové slová
5. Abstrakt
6. Úvod
7. Jadro
 - Kapitoly - Nadpis
 - Podkapitoly - Nadpis

- Text v rámci kapitol je rozdelený na paragrafy
8. Záver

Elementy 1-4 sú metadáta, 5-8 sú obsahové. Z tabuľky 4 je zrejme, že všetky elementy okrem siedmeho je možné vytvoriť v každom CMS v našom prehľade. Problém siedmeho elementu je, že sa sám skladá z elementov a navyše má komplexnú kardinalitu. Túto mieru detailu dokáže na úrovni elementov zachytiť iba Bricolage.

Tabuľka 5: Príklad 1

Bricolage	Drupal	Plone, ezPublish
Kapitola [0-X] <ul style="list-style-type: none"> • Podkapitola [0-X] • Paragraf [0-X] [Text] 	Kapitola [0-X] [HTML]	Text [1] [HTML/XML]

Z príkladu vyplývajú dôležité závery.

1. Štruktúru obsahu v rámci CMS môžeme definovať na dvoch úrovniach – typom komponentu, a typom elementu – HTML/XML. Niektoré CMS poskytujú dostatočne silný model pre typy elementov, že štruktúru v rámci elementov nepotrebujeme. Na druhej strane mnohé jednoduchšie CMS štruktúru obsahu na úrovni elementov definovať nedovoľujú.
2. Granularita nieje pevná a je možné ju (do istej miery) ovplyvniť podľa našich požiadaviek. Napríklad všetky obsahové elementy v príklade sme v prípade Drupal mohli nahradiť jediným elementom (HTML) a štruktúru kapitol definovať v rámci neho.

Typy elementov s vnútornou štruktúrou nazývame komplexné. Sú predmetom nasledujúcej kapitoly.

Spoločným vodítkom pri definovaní typov elementov je definícia elementu ako najmenšieho adresovateľného objektu v rámci CMS. Na našom príklade diplomových prác: atraktívnou funkciou je možnosť nahliadnuť na diplomovú prácu, teda poskytnúť jednoduchý dvojestupňový model, abstrakt, záver – celá práca. Ak sa rozhodneme, že prístup po kapitolách nepotrebujeme je výsledkom typ komponentu s tromi obsahovými elementami, abstrakt, práca (úvod a jadro), záver. Na druhej strane prístup po kapitolách má mnohé výhody, napríklad v rámci webového rozhrania by bol prístup po kapitolách príjemný, keďže diplomové práce bývajú rozsiahle. V tomto prípade dostávame typ komponentu: abstrakt, úvod, N kapitol, záver.

V definícií štruktúry komponentov v CMS sme identifikovali tieto problémy:

1. Prelínanie štruktúry elementov a štruktúry komponentov.
2. Neštandardný popis štruktúry. Domnievame sa, že autori CMS by mali využiť existujúce štandardy pre popis dátových štruktúr. Kandidátmi sú štandardy XML Schema, RELAX NG, prípadne OWL.
 - Prvým krokom je export popisu štruktúry komponentu, ktorý môže byť následne použitý aplikáciami tretích strán (napríklad XML editory s podporou XML schema).
 - Ďalším krokom je umožniť definíciu štruktúry priamo pomocou štandardného jazyka. Na základe schémy potom (automaticky) vytvoriť webové formuláre pre

vstup obsahu ako aj fyzickú reprezentáciu (tj. model tabuliek v relačnej databáze, hierarchiu súborov, ap.)

3. Vnútna štruktúra komponentov zachytáva informácie o sémantike obsahu. Tieto informácie sú viditeľné len v rámci CMS, ich sprístupnenie aj v rámci publikácií je kľúčovou požiadavkou pre *Sémantický Web*. Z praktického hľadiska je najvhodnejším kandidátom štandard RDF.

Čiastočné riešenie druhého problému poskytuje Plone. Keďže Plone dovoľuje vytvoriť nový komponent z UML diagramu, môžeme použiť UML aplikáciu, ktorá vie na základe XML Schema/Relax NG/OWL vytvoriť UML diagram. Príkladom je aplikácia Protege¹, ktorá (pomocou prídavných modulov) zabezpečí preklad medzi OWL a UML(xmi) formátom.

Definícia: RDF je všeobecný model pre popis (webových) zdrojov, použitím mechanicky spracovateľných metadát. Technicky ide o jazyk pre popis orientovaných grafov. Vrcholy grafu predstavujú zdroje, orientované hrany definujú vzťah medzi zdrojmi. Graf sa popisuje pomocou *tripleto*v, vektorov vo forme subject-predicate-object. Často sa RDF označuje ako jazyk *Sémantického Webu* (Web 2.0). Sémantikou je práve mechanicky čitateľný význam prepojení [14].

1 <http://protege.stanford.edu>

2.4 Komplexné elementy

Komplexné elementy sú elementy s vnútornou štruktúrou. Objekty v rámci tejto štruktúry už nie sú v CMS viditeľné. To znamená že ich nezávisle od elementu nie je možné použiť v publikáciách, nieje možné kontrolovať prístup k nim a podobne. Samozrejme existuje úroveň detailu kde funkcie CMS už nepotrebujeme, pre väčšinu použití je vytváranie elementov pre každý znak alebo slovo textu zbytočné.

CMS sa často spoliehajú na externé aplikácie pre vytváranie obsahu, a tak namiesto implementácie komplikovaných konverzií, využívajú aplikačný formát priamo v komplexných elementoch.

K definícií štruktúry textového obsahu sa najčastejšie používajú takzvané *markup* jazyky. Tieto sú založené na využívaní syntaktických konštrukcií vsadených priamo do textu. Rovnako ako existuje široké spektrum CMS, existuje aj široké spektrum používaných jazykov.

Podľa pôvodu jazyka:

- otvorený štandard (HTML, DocBook, OpenDocument).
- vlastný jazyk, špecifický pre CMS.

Podľa typu jazyka:

- XML jazyk (XHTML, DocBook).
- jazyk nezaložený na XML (Wiki, BBCode).

Podľa použitia:

- Jazyk orientovaný striktne na štruktúru obsahu (DocBook).
- Jazyk zahŕňa tak konštrukcie pre definíciu štruktúry ako aj formátovania obsahu.

2.4.1 XHTML

Využitie formátu HTML v CMS je rozšírené. Prakticky sa však nepoužívajú všetky konštrukcie z HTML, prípadne CMS dovoľujú množinu povolených konštrukcií definovať. XHTML je XML verzia HTML, preberá všetky elementy, pričom definuje striktnejšie pravidlá pre ich používanie. Predpokladáme, že užívateľovi je problematika HTML – XHTML dostatočne známa.

Výhody:

- Známy jazyk.
- Vyzreté nástroje pre vytváranie a konverziu do iných formátov
- XHTML je XML jazyk, je teda možné použiť rozšírené nástroje pre spracovanie a kontrolu (štruktúry) obsahu.
- Nieje potrebná konverzia pre použitie na webe.

Nevýhody:

- Relatívne málo štruktúrovaný jazyk. Napríklad je možné použiť nadpis tretej úrovne bez predchádzajúcich nadpisov prvej a druhej. Obsah nieje možné rozdeliť do (vnorených) logických častí (tj. kapitol, podkapitol).
- Nakoľko nieje potrebná konverzia, často sa zanedbáva kontrola obsahu. Jedným z

dôsledkov sú napríklad XSS (*Cross site scripting* tiež CSS) útoky. Dnes však existujú nástroje, ktoré problém filtrovania HTML riešia, populárna je napríklad aplikácia HTML Tidy.

- Problém bieloho miesta. Medzera a nový riadok patria medzi základné formátovacie nástroje, HTML však biele miesto nerešpektuje, všetky znaky sú nahradené jedinou medzerou.

Definícia: ECMAScript je programovací jazyk, najčastejšie používaný v rámci HTML dokumentov. Je vykonávaný na strane klienta, tj. vo webovom prehliadači na počítači užívateľa.

Poskytuje prístup k obsahu celej stránky a cookies.

Definícia: *Cookies* sú dáta asociované s webovým serverom uložené na strane klienta. Keď webový prehliadač žiada server o konkrétnu webovú stránku, pošle súčasne aj cookies. Naopak pri odpovedi môže server poslať okrem stránky aj nové cookies.

Výpis 2: XSS 1.

```
1 <a href="javascript:document.write('hello');document.write(' world');">hello</a>
```

V prípade, že užívateľ klikne na tento odkaz, obsah stránky sa prepíše. Na prvý pohľad, sa javí tento problém ako neškodný.

Výpis 3: XSS 2.

```
1 <a href="javascript:document.location='http://www.test.sk/'+document.cookie;">Go!</a>
```

Druhý príklad demonštruje riziko: *cookie*, ktoré CMS uložila na strane klienta sú poslané tretej strane, v našom príklade *www.test.sk/*. Poznamenajme, že častým použitím cookies je záznam o autentifikácii užívateľa ako aj automatické prihlásenie.

2.4.2 Wiki

Neexistencia štandardu pre wiki jazyk spôsobila, že CMS, ktoré podporujú vytváranie obsahu vo forme wiki, implementujú vlastný dialekt. Vytvorenie štandardu je snahou napríklad týchto projektov [5], [6].

Napriek rozdielom majú wiki jazyky mnoho spoločných vlastností.

- Jednoduchosť vytvárania odkazov. Prakticky všetky Wiki, umožňujú vytvárať odkaz spojením slov, tak že prvé písmená slov sú veľké, takzvaný *camel case*.
- Riadkovo orientované, väčšina konštrukcií musí začínať na novom riadku, výnimkou sú napríklad konštrukcie pre ležaté resp. tučné písmo. Paragrafy sú vytvárané pomocou dvoch bezprostredne nasledujúcich znakov nového riadku.
- Používanie špeciálnych znakov pre syntaktické konštrukcie.

Tabuľka 6: Syntaktické konštrukcie wiki dialektov

Syntax			Sémantika	Prezentácia
TikiWiki	MoinMoin	MediaWiki		
MojaStranka NoveSpravy			Odkaz na internú wiki-stránku	MojaStranka NoveSpravy
[[http://tikiwiki.org/ The TikiWiki Site]]			Odkaz na externú webovú stránku. Prvá časť špecifikuje URL, nepovinná časť za je použitá ako názov odkazu.	The TikiWiki Site
"slovo"			Ležaté písmo	<i>slovo</i>
__slovo__		NA	Podčiarknutie	<u>slovo</u>
slovo	" slovo " ²		Tučné písmo	slovo
#prvý element #druhý element	1. prvý element 1. prvý element	#prvý element #druhý element	Číslovaný zoznam	1. prvý element 2. druhý element
*prvý element *druhý element **sub element	*prvý element *druhý element *sub element	*prvý element *druhý element **sub element	Zoznam	• prvý element • druhý element • element
= Nadpis =	!Nadpis	= Nadpis =	Nadpis 1. úrovne	Nadpis
== Nadpis 2 ==	!!Nadpis 2	== Nadpis 2 ==	Nadpis 2. úrovne	Nadpis 2

Okrem konštrukcií, ktoré ovplyvňujú vzhľad obsahu, poskytujú wiki jazyky aj možnosti pre definíciu štruktúry a to napríklad vo forme nadpisov. Jednotlivé nadpisy sú následne spojené do tabuľky obsahu, čo je atraktívna funkcia hlavne v prípade rozsiahlejšieho obsahu.

Zaujímavú funkciu poskytuje MediaWiki. Dovoľuje totiž modifikovať nielen celý obsah, ale aj samostatne jednotlivé časti – odseky. Odsek je text medzi dvoma nadpismi, respektíve medzi nadpisom a koncom obsahu. Domnievame sa, že toto riešenie je pre wiki veľkým prínosom. Hlavne v prípade malých zmien, je často problematické nájsť v rámci rozsiahleho textu konkrétne miesto kde zmenu vykonať. Mierne zlepšenie z pohľadu

² Ide o tri jednoduché úvodzovky.

užívateľa navrhuje autor [7]. Keďže pozícia ovládacieho prvku nekorešponduje presne s daným odsekom (presunúť ovládací prvok z miesta nad odsekom, pod daný odsek).

Výhody:

- Jednoduchosť vytvárania odkazov (*CamelCase*) a paragrafov v texte (dva nové riadky).

Nevýhody:

- Neexistencia štandardu, ktorá na jednej strane vyžaduje opakované zaškolenie pre nový CMS a súčasne zväzuje obsah s konkrétnym CMS.
- Využívanie špeciálnych znakov, napríklad [{ * |, nakoľko rozloženie kláves sa pre tieto znaky líši medzi národnými klávesnicami a súčasne sú relatívne zložito dostupné.
- Špeciálne znaky môžu byť využívané v texte, čo vedie k pomerne komplikovaným programom pre spracovanie wiki jazykov.

Výpis 4: Príklad nejednoznačnosti syntaxe wiki jazyka [http://en.wikipedia.org/wiki/Lost_\(TV_series\)](http://en.wikipedia.org/wiki/Lost_(TV_series))

```
''Lost''s two-part [[pilot episode]] was the most expensive in the network's history, reportedly costing between [[United States dollar|US $]]10 and US $14 million,<ref>{{cite news|title=New series gives Hawaii 3 TV shows in production| last=Ryan|first=Tim|url=http://starbulletin.com/2004/05/17/news/story7.html|publisher=Honolulu Star-Bulletin|date=May 17, 2004}}</ref> compared to the average cost of an hour-long pilot in 2005 of US $4 million.
```

Vidíme, že slovo **lost** je v príklade obklopené dvoma úvodzovkami na ľavej a tromi na pravej strane. Je jedna z pravých úvodzoviek súčasťou textu, alebo má byť text za nimi tučným písmom? Výsledok závisí od všetkých nasledujúcich konštrukcií v rámci paragrafu.

Za seba hovorí aj použitie programu *Tidy* ako posledného kroku pri konverzií do HTML. Algoritmus prekladu medzi wiki a HTML je značne komplikovaný, zvedavého čitateľa odkazujeme na zdrojový kód Mediawiki, nakoľko prakticky žiadna iná technická dokumentácia alebo schéma neexistuje.

Syntaktické pravidlá pre wiki jazyky sú často komplikované, nedostatočne striktné a nedostatočne zdokumentované. Migrácia z CMS s obsahom vo wiki jazyku je preto náročnou úlohou a vyžaduje užívateľskú asistenciu pri riešení konfliktných situácií.

2.4.3 Štruktúrovaný text

Podobne ako wiki jazyk používa aj štruktúrovaný text špeciálne znaky pre definíciu štruktúry a formátovania. Rozdielom je používanie bieleného miesta ako štruktúrálnej informácie.

Tabuľka 7: Príklad syntaxe štruktúrovaného textu

Syntax	Sémantika	Prezentácia
slovo	Ležaté písmo	<i>slovo</i>
slovo	Podčiarknutie	<u>slovo</u>
slovo	Tučné písmo	slovo
Nadpis paragraf paragraf paragraf paragraf paragraf paragraf paragraf paragraf Nadpis paragraf paragraf paragraf paragraf	Nadpis a paragraf textu, vnorený nadpis a vnorený paragraf textu. Podstatné sú odsadenie paragrafov a nový riadok medzi nadpisom a textom.	Nadpis 1 paragraf paragraf paragraf paragraf Nadpis paragraf paragraf paragraf paragraf
* prvok zoznamu * vnorený prvok * ďalší prvok	Zoznam. Znovu, vnorenie je definované odsadením, prvky sú oddelené novými riadkami.	<ul style="list-style-type: none"> • prvok zoznamu <ul style="list-style-type: none"> • vnorený prvok • ďalší prvok
"Odkaz":http://www.examble.com	Odkaz na webovú stránku	Odkaz

Ďalším vývojovým štádiom štruktúrovaného textu je „reštruktúrovaný“ text. Odstraňuje niektoré komplikované konštrukcie, napríklad vnorené paragrafy textu. Namiesto odsadenia využíva podčiarknutie nadpisu špeciálnym znakom. Tak ako sú rôzne podčiarknutia nachádzané sú im priradené úrovne.

Nadpis 1 ----- paragraf paragraf paragraf paragraf Nadpis 2 +++++++ paragraf paragraf paragraf paragraf Nadpis 3 ----- paragraf paragraf paragraf paragraf	Nadpis a paragraf textu, vnorený nadpis a vnorený paragraf textu. Podstatné je podčiarknutie špeciálnym znakom, pričom počet špeciálnych znakov musí byť rovnaký ako počet znakov nadpisu	Nadpis 1 paragraf paragraf paragraf paragraf Nadpis 2 paragraf paragraf paragraf paragraf Nadpis 3 paragraf paragraf paragraf paragraf
--	---	--

Výhody:

- Jednoduchý jazyk, inšpirovaný syntaxom jazyka Python a wiki.
- Dovoľuje definovať hierarchické logické časti.

Nevýhody:

- Málo rozšírený.
- Pomerne komplexné pravidlá.

2.4.4 EzXML

Komplexné elementy v ezPublish využívajú jazyk ezXML. Ako je zrejme z názvu ide o XML jazyk. Niektoré xml-elementy sú prevzaté z HTML čo privítajú užívatelia, ktorí jazyk HTML už poznajú. Rozšírením oproti HTML je využitie vnorených elementov a striktnější štruktúra xml-elementov.

Výpis 5: Zjednodušený náhľad na štruktúru ezXML

<pre>section • header (podobný <H1>, <header level='1'>) • strong (ekvivalent) • emphasize (ekvivalent <i>) • link • paragraph • table • tr • td • header • paragraph • section • table • literal (ekvivalent <pre>) • ul, ol • li • paragraph • strong, emphasize, link • section (logická podkapitola/pod-odsek)</pre>

Výhody:

- Jednoduchý jazyk, inšpirovaný HTML a DocBook-om.
- Dovoľuje definovať hierarchické logické časti (section).
- Syntaktické pravidlá sú pomerne presné, ezPublish dodáva schému a obsah je pred uložením kontrolovaný.

Nevýhody:

- Málo rozšírený, v súčasnosti používaný iba CMS ezPublish.
- Chýbajú niektoré základné HTML elementy ako podčiarknutie, či prečiarknutie.
- Chýba schéma v otvorenom formáte ako XML schéma, alebo RELAX NG.
- Existujúca schéma je nedostatočná, nedovoľuje ovplyvniť kardinalitu xml elementov.

2.4.5 BBCode

Tento jazyk je podobný HTML, preberá názvy niektorých elementov ako aj myšlienku otváracích a uzatváracích konštrukcií. Na rozdiel od HTML (XML) namiesto znakov < > využíva znaky [].

Pôvodne vznikol ako preferovaný jazyk CMS typu diskusné fórum (napríklad CMS *phpBB* alebo *vBulletin*, skratka BB znamená *bulletin board*). Dnes podobne ako v prípade wiki existuje niekoľko čiastočne odlišných a vzájomne nekompatibilných verzií.

Tabuľka 8: Porovnanie syntaxe BBCode a HTML

BBCode	HTML	Prezentácia
[url=http://www.phpbb.com/]odkaz[/url]	odkaz	odkaz
[b]text[/b]	text	text
[color=green]farebný text[/color]	farebný text	Farebný text

Výhody:

- Jednoduchý jazyk, inšpirovaný HTML.
- Jednoduchší ako HTML, menej priestoru pre potenciálne zneužitie.
- Podstatne jednoduchší na spracovanie ako wiki.
- Syntaktické pravidlá sú presne, podobne ako XML využíva tagy.
- Poskytuje špeciálne elementy pre diskusné fóra, napríklad [q] [/q] pre citovanie.

Nevýhody:

- Málo rozšírený.
- Vhodný hlavne na krátke texty.

2.4.6 DocBook

DocBook je veľmi populárna množina tagov pre popis kníh, článkou a iných textových dokumentov, najmä technickej dokumentácie [49]. DocBook je podobne ako XHTML a ezXML príkladom XML jazyka.

Najdôležitejším rozdielom oproti HTML je, že DocBook popisuje výhradne štruktúru – sémantiku dokumentu, neposkytuje nijaké konštrukcie pre definíciu formátovania, tj. vzhľadu.

V nasledujúcich príkladoch si predstavíme niektoré základné tagy a konštrukcie v DocBook. Použité príklady pochádzajú z [49].

Výpis 6: Příklad Knihy

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<book>
<bookinfo>
  <title>My First Book</title>
  <author><firstname>Jane</firstname><surname>Doe</surname></author>
  <copyright><year>1998</year><holder>Jane Doe</holder></copyright>
</bookinfo>
<preface><title>Foreword</title> ... </preface>
<chapter> ... </chapter>
<chapter> ... </chapter>
<chapter> ... </chapter>
<appendix> ... </appendix>
<appendix> ... </appendix>
<index> ... </index>
</book>
```

Výpis 7: Příklad Kapitoly

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<chapter><title>My Chapter</title>
<para> ... </para>
<sect1><title>First Section</title>
<para> ... </para>
<example> ... </example>
</sect1>
</chapter>
```

Výpis 8: Příklad Článku

```
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<article>
<artheader>
  <title>My Article</title>
  <author><honorific>Dr</honorific><firstname>Emilio</firstname>
    <surname>Lizardo</surname></author>
</artheader>
<para> ... </para>
<sect1><title>On the Possibility of Going Home</title>
<para> ... </para>
</sect1>
<bibliography> ... </bibliography>
</article>
```

Výhody:

- Založený na XML.
- Dôsledne oddeľuje prezentáciu a štruktúru obsahu.
- Striktná schéma v otvorenom formáte: DTD, XML Schema, RELAX, RELAX NG.
- Veľké množstvo špecializovaných tagov, vyzretý a v praxi overený štandard OASIS (Organization for the Advancement of Structured Information Standards).
- Rozšíriteľný, MathML, SVG.
- Mnoho nástrojov a aplikácií pre ďalšie spracovanie.
- Podpora pre viac jazykov (kontrolované atribútom lang) v rámci jedného dokumentu.

Nevýhody:

- Komplexný štandard, čiastočné riešenie poskytuje „Simplified DocBook“, ktorý však stále obsahuje 119 xml elementov.
- Aplikácie, ktoré DocBook spracúvajú sú vzhľadom na komplexnosť DocBook-u relatívne náročné na systémové prostriedky, mnohé sú dostupné len v komerčnej verzii a za pomerne vysokú cenu.

2.4.7 Zhrnutie

Tabuľka 9: Podpora textových elementov

CMS	Text	HTML	ezXML	Wiki	BBCode	DocBook	Štruktúrovaný text
Bricolage	Áno						
Drupal	Áno	Áno, konfigurovateľná podmnožina			Áno		
ezPublish	Áno		Áno				
Magnolia	Áno	Áno					
Mambo	Áno	Áno				Experimentálny modul	
MediaWiki		Áno, v rámci Wiki		Áno			
Plone	Áno	Áno, konfigurovateľná podmnožina				Čiastočne, modul PloneDocbook	Áno
Tikiwiki		Áno		Áno			
Typo3	Áno						
Wordpress		Áno					

MediaWiki, MoinMoin, TikiWiki

HTML je vložený priamo do obsahu vo wiki jazyku. Toto riešenie považuje za nevhodné nakoľko značne zvyšuje komplexnosť spracovania.

Drupal

Okrem jazykov spomenutých v tabuľke 9, podporuje i mnohé ďalšie formou prídavných modulov. Zaujímavé je na tejto architektúre, že prídavné moduly môžu byť použité súčasne, v zadefinovanom poradí. Ich úlohou je zdrojový jazyk preložiť do reprezentácie vhodnej pre HTML.

Za pozornosť stojí modul XML Content, ktorý predstavuje „jazykový generátor“ pre Drupal. Nový jazyk je vytvorený pomocou schémy (XML Schema, alebo RelaxNG) a XLST transformácie (do XHTML).

2.5 Vonkajšia štruktúra obsahu

Ludské chápanie sveta je z veľkej miery založené na schopnosti organizovať informácie. Organizácia je kľúčom k nájdeniu, vysvetleniu a pochopeniu informácie. Nové poznatky, akokoľvek prevratné nevznikajú vo vákuu, vždy vychádzajú z existujúcich. Súčasne sú vždy implicitne spojené z otázkami, na ktoré sa snažia odpovedať. Samotná informácia je bezcenná, zmysel jej dáva až prepojenie z inými informáciami, kontext v ktorom existuje. Preto je jednou z najdôležitejších funkcií CMS práve organizácia obsahu. Možnosti a nástroje pre organizáciu komponentov obsahu sú predmetom tejto kapitoly.

Existujú dva základné modely pre organizačnú štruktúru.

1. **Statická štruktúra** modeluje explicitné prepojenie medzi komponentami.
2. **Dynamická štruktúra** vychádza z vlastností a atribútov obsahu. Medzi komponentami s podobnými alebo rovnakými atribútmi vytvára implicitné prepojenie.

Na základe statickej štruktúry sme identifikovali dva základné druhy CMS:

1. Komponenty je možné vytvárať len v rámci statickej štruktúry.
2. Medzi komponentami nie sú povinné explicitné prepojenia.

Tabuľka 10: Rozdelenie CMS podľa vonkajšej štruktúry obsahu

CMS 1. druhu	CMS 2. druhu
ezPublish	Bricolage
Magnolia	Drupal
MoinMoin	Mambo
Plone	MediaWiki
Typo3	Wordpress
	Tikiwiki

Ako sme uviedli v tabuľke 10 CMS 1. druhu nútia užívateľov vytvárať obsah v explicitnom kontexte. Štandardným postupom je použitie hierarchie. Kým v EzPublish a Plone sú uzlami v strome špeciálne komponenty typu adresár – *folder*, v ostatných prípadoch sú tak uzly ako aj listy stromu ľubovoľné komponenty.

Výhody:

- Obsah v rámci CMS je možné rozdeliť do logických častí, napríklad podľa zamerania, alebo skupiny užívateľov.

Nevýhody:

- Domnievame sa, že ide o relatívne komplikovaný dizajn, ktorý navyše spája prezentačnú logiku so štruktúrou obsahu. Často je voľba kam obsah zaradiť nejednoznačná, čo vedie ku nekonzistencii.

2.5.1 Statická štruktúra obsahu

Diskusia

Štruktúru diskusia sme už predstavili. V prípade, že užívatelia môžu nové diskusie vytvárať hovoríme o *voľnej* diskusii. *Viazaná* diskusia je štruktúra, v ktorej existujúci komponent, chápeme ako koreň diskusie. Právo vytvoriť prvý obsah je zvyčajne vyhradené len pre niektorých užívateľov.

Drupal

Drupal poskytuje hierarchickú štruktúru – *knihu*. Definuje povolené prepojenia medzi elementárnymi obsahmi – *stránkami*. Povolené prepojenia kopírujú stromovú štruktúru predchodca – nasledovník. Navyše sú nasledovníci daného uzla usporiadaný zľava do prava. V tomto prípade hovoríme o sesterskom vzťahu. Stránky sú zaradené do hierarchie pri vytvorení, ale i neskôr je ich možné presúvať. Každá stránka knihy je zobrazená priamo s navigačnými prepojeniami. V prípade, že stránka je zmazaná, sú všetky následnícke stránky zaradené do špeciálnej knihy *sirotské stránky - orphaned pages*.

Tikiwiki

Obdobou knihy sú v Tikiwiki takzvané *structures*, ktoré predstavujú wiki-stránky tvoriace strom. Podobne ako v Drupal existujú sesterské vzťahy medzi wiki-stránkami. Pri vytvorení novej *structure*, Tikiwiki automaticky vygeneruje prázdne stránky zodpovedajúce uzlom stromu. Do *structures* môžeme pridávať nové alebo aj existujúce obsahy. Navyše konkrétny obsah môže byť, na rozdiel od Drupal, súčasne v niekoľkých *structures*. Problematická je však navigácia, automaticky majú stránky len odkazy na vedľajšie sesterské stránky a predchodcu, nie však na nasledovníkov. Riešenie spočíva v použití rezervovaného slova *{toc}* v texte wiki-stránky, ktoré spôsobí zobrazenie odkazov na celú časť podstromu.

MoinMoin, MediaWiki

Poskytujú možnosť vytvárať pod-stránky a definujú syntax pre vyjadrenie vzťahu predchodca – nasledovník ako aj sesterských vzťahov wiki-stránkami. Myšlienka vychádza z definície cesty v súborovom systéme, výraz *../* označuje predchodcu wiki-stránky, rezervované slovo *subpages* vloží odkazy na nasledovníkov wiki-stránky (prípustné sú i kombinácie, napríklad *../subpages*, predstavuje odkazy na sesterské wiki-stránky). Pripomeňme, že Wikipedia zakazuje vytvárať pod-stránky nakoľko sa pevná štruktúra ukázala ako nevhodná (pod-stránka bola relevantná pre viac ako jednu wiki-stránku).

Tabuľka 11: Podpora statických štruktúr

CMS	Diskusia		Explicitná statická štruktúra
	Voľná	Viazaná	
Bricolage	Nie	Áno	Nie
Drupal	Áno	Áno	Book (Kniha)
EzPublish	Nie	Áno	Adresár
Magnolia	Nie	Nie	Áno
Mambo	Áno, rozširujúci modul	Nie	Nie
MediaWiki	Čiastočne*	Čiastočne*	Pod-stránky
MoinMoin	Nie	Nie	Pod-stránky
Plone	Áno	Áno	Adresár
TikiWiki	Áno	Áno	Structure (štruktúra)
Typo3	Áno	Nie	Áno
Wordpress	Nie	Áno	Nie

* Poskytuje kľúčové slovo, ktoré vloží odkaz na *diskusiu* o konkrétnom článku. Diskusia je však bežná wiki-stránka, ktorú užívatelia modifikujú takým spôsobom, aby bolo zrejmé **kto** a na **čo** reaguje. Udržovanie štruktúry je úlohou autorov.

2.5.2 Dynamická štruktúra obsahu

Dynamickou štruktúrou popisujeme také usporiadanie obsahu, ktoré nie je dané explicitne, ale je odvodené automaticky z metadát. Prakticky všetky CMS implementujú istý druh dynamickej štruktúry. Vzhľadom na vytváranie delíme dynamické štruktúry na dva druhy:

1. **Top-down:** oprávnený užívateľ vytvorí preddefinovanú množinu pojmov a ich vzťahov, ktoré následne užívatelia – autori používajú k popisu obsahu.
2. **Bottom-up:** autori môžu použiť k popisu ľubovoľné pojmy. Vytváranie vzťahov medzi pojmami je až ďalším krokom.

Kategórie

Taxonómiu definujeme ako množinu pojmov - kategórií a množinu povolených vzťahov medzi pojmami. Pôvodný význam tohto slova je vedecká disciplína, ktorá sa zaoberá klasifikáciou organizmov. V prenesenom zmysle ide o (hierarchickú) klasifikáciu predmetov. Taxonómia je založená na princípe špecializácie smerom zhora nadol. Nasledovníci daného uzla sa navzájom líšia v nejakom konkrétnom znaku, alebo vlastnosti, pričom zdieľajú znaky predchodcu. Iným príkladom hierarchie je vzťah celok – časť.

Použitie taxonómie má zmysel v prípade, že poznáme znaky podľa, ktorých hierarchiu vybudujeme. Reprezentujú teda top-down prístup. Pozrime sa bližšie aké druhy existujú.

Z hľadiska vzťahov medzi pojmami:

- **Hierarchické** – každá kategória (okrem koreňa) má práve jedného predchodcu (otca).
- **Poly-hierarchické** – kategória môže mať niekoľko predchodcov.
- Rozšíreným druhom je **thesaurus**. Prakticky ide o hierarchické kategórie, pričom definuje ďalšie vzťahy medzi kategóriami. Základným vzťahom je všeobecnejší – špecifickejší pojem, tento vzťah modeluje hierarchiu. Vo vzťahu ekvivalencia sú synonymá, pričom jeden z pojmov je preferovaný a ostatné označujeme ako variant. Posledným vzťahom je asociácia, ktorá nemá presne definovanú sémantiku.

Z hľadiska použitia taxonómií v CMS existujú tieto základné kritéria:

1. Podporovaný druh taxonómie. Teda povolené druhy vzťahov kategóriami.
2. CMS dovoľujú definovať niekoľko nezávislých taxonómií.
3. Komponent môže byť popísaný súčasne niekoľkými pojmami, teda patriť do niekoľkých kategórií.
4. Zaradenie komponentu do kategórie je povinné.

EzPublish, Plone

Tieto CMS nepodporujú taxonómie priamo, avšak vďaka flexibilnému systému komponentov je možné túto funkciu pridať. V ezPublish môžeme pri definícii typu komponentu pridať nový typ elementu, ktorý dovolí užívateľovi vybrať z niekoľkých pojmov. Priamo v štandardnej inštalácii môžeme použiť typ elementu *enum*. K definícii hierarchie pojmov však musíme vytvoriť vlastný typ elementu (tj. rozširujúci modul).

Problém takéhoto riešenia je, že nieje integrovaný s CMS, mnohé funkcie (napríklad vyhľadávanie podľa kategórie) je nutné vytvoriť (či už pomocou dostupných nástrojov CMS alebo ako rozširujúci modul).

TikiWiki

Nakoľko sú si *structures* a kategórie v TikiWiki do istej miery podobné, pozrime sa aký je medzi nimi rozdiel. Kým *structures* definujú vzťahy čiastkových obsahov v rámci komplexného obsahu, kategórie agregujú obsah iba približne, pričom obsahy v rovnakej kategórii sú nezávislé.

Mambo

Obmedzenú formu dynamickej štruktúry poskytuje Mambo. Na najvyššej úrovni sú sekcie, v rámci sekcií sú kategórie. Okrem tohto vlastného vzťahu medzi sekciami a kategóriami, sú sekcie a kategórie nezávislé. Každý obsah je nutné zaradiť do práve jednej sekcie a do práve jednej jej kategórie. Výnimkou je špeciálny druh obsahu takzvaný *statický obsah*, ktorý nieje možné kategorizovať.

MoinMoin

MoinMoin umožňuje vytvoriť dynamickú štruktúru pomocou *backlink-ou*. *Backlinky* predstavujú množinu wiki-stránok v rámci MoinMoin, ktoré sa odkazujú na konkrétnu wiki-stránku. Funkcie wiki jazyka MoinMoin umožňujú vložiť zoznam backlinkov priamo do obsahu stránky. Konvencia určuje použitie názvov v tvare KategóriaNázov pre mená stránok kategórií. Kategorizované wiki-stránky, by mali uvádzať kategórie na konci textu, pod čiarou. Opakujeme, že ide iba o konvenciu, nakoľko je celá kategorizácia založená na backlinkoch, môžu sa odkazy na kategórie vyskytovať kdekoľvek na stránke, rovnako meno pre stránku kategórie môže byť ľubovoľné. Systém dovoľuje vytvorenie neobmedzeného množstva kategórií. Vzhľadom na vytváranie kategórií je definícia štruktúry v rukách autora.

Tabuľka 12: Taxonómie v CMS

CMS	Druh	Nezávislé Taxonómie	Viac kategórií	Povinné
Bricolage	Hierarchické	Áno	Áno	Áno
Drupal	- Hierarchické a polyhierarchické - Thesaurus (synonymá, príbuzné kategórie)	Áno	Áno(konfigurovateľné)	Áno(konfigurovateľné)
EzPublish	Nie	Nie	Nie	Nie
Mambo	- Obmedzená dvoj-úrovňová hierarchická štruktúra.	Áno	Nie	Áno
MediaWiki	- Voľná štruktúra prepojení.	Áno	Áno	Nie
MoinMoin	- Voľná štruktúra kategórií.	Áno	Áno	Nie
Plone	Nie	Nie	Nie	Nie
TikiWiki	Hierarchické	Nie	Áno	Nie
Typo3	Nie	Nie	Nie	Nie
Wordpress	Hierarchické	Áno	Áno	Áno, inak je použitá predvolená kategória

Tag

Populárnym druhom dynamickej bottom-up štruktúry je **Tag**. Množinu tagov vytvárajú a používajú súčasne všetci autori. Vzťahy medzi tagmi sú definované postupne, tak ako sú objavované vzory použitia. Príkladom je *cluster*, využívaný napríklad webovým portálom *flickr*. *Cluster* je podmnožina tagov, ktorá popisuje signifikantný počet obsahov s podobným predmetom. Využívanie tagov sa často označuje ako *social classification*. Stretnúť sa môžeme i s pojmom *folksonomy*.

Explore / Tags / holiday / clusters



[christmas](#), [tree](#), [lights](#), [xmas](#), [family](#), [christmastree](#), [decorations](#), [red](#), [ornament](#), [winter](#)

→ [See more in this cluster...](#)



[beach](#), [vacation](#), [travel](#), [summer](#), [sun](#), [sky](#), [blue](#), [sunset](#), [nature](#), [trip](#)

→ [See more in this cluster...](#)

Obrázok 4: Cluster (www.flickr.com)

Kľúčové slová

Často sú tagy prekladané ako kľúčové slová. Avšak kým kľúčové slová hovoria o predmete daného obsahu, použitie tagov je všeobecnejšie, napríklad miesto alebo štát pôvodu obrázku a podobne. Kľúčové slová teda predstavujú špecializovaný druh tagov.

Tabuľka 13: Porovnanie kategórií a tagov

	Kategórie	Tag
Štruktúra	Definovaná, najčastejšie strom	Bez štruktúry. Tagy sú nezávislé
Počet vzhľadom ku konkrétnemu obsahu	Z pravidla jediná kategória	Neobmedzené množstvo
Časová premenlivosť	Statická, časovo nepremenná	Tagy je možné kedykoľvek dopĺňať

Kategórie umožňujú hľadať obsah, podľa všeobecných črt – od všeobecnejšieho popisu k detailnejšiemu, kým tagy zjednodušujú hľadanie obsahu, podľa konkrétnych atribútov.

Keďže na rozdiel od kategórií nemajú tagy (preddefinované) vnútornú štruktúru vzniká problém ako zobrazit' tagy prislúchajúce obsahu.

Definícia: *Tag cloud*, alebo vážený zoznam kľúčových slov, je zobrazenie množiny kľúčových slov, tak že slová s väčšou váhou – častejšie využívané, sú graficky zvýraznené. Slová sú zoradené v abecednom poradí, zvýraznenie sa dosahuje farbou alebo veľkosťou.

[ajax](#) [apple](#) [art](#) [article](#) [articles](#) [audio](#) [bittorrent](#) [blog](#) [blogs](#) [books](#) [business](#) [comics](#) [community](#) [computer](#) [cool](#) [css](#)
[culture](#) [daily](#) [database](#) [design](#) [development](#) [diy](#) [education](#) [entertainment](#) [finance](#) [firefox](#) [flash](#) [food](#) [free](#) [freeware](#) [fun](#)
[funny](#) [games](#) [gaming](#) [google](#) [graphics](#) [gtd](#) [hardware](#) [health](#) [history](#) [home](#) [howto](#) [html](#) [humor](#) [internet](#) [java](#) [javascript](#)
[jobs](#) [korean](#) [library](#) [linux](#) [mac](#) [magazine](#) [maps](#) [media](#) [misc](#) [music](#) [news](#) [opensource](#) [osx](#) [p2p](#) [personal](#) [photo](#)
[photography](#) [photos](#) [php](#) [politics](#) [productivity](#) [programming](#) [radio](#) [reference](#) [research](#) [rss](#) [ruby](#) [sagefeeds](#)
[science](#) [search](#) [security](#) [shopping](#) [software](#) [sports](#) [tech](#) [technology](#) [tips](#) [tools](#) [toread](#) [travel](#) [tutorial](#)
[tutorials](#) [tv](#) [updatedcollection.info](#) [updateddatabase.info](#) [video](#) [web](#) [web2.0](#) [webdesign](#) [webdev](#) [wiki](#) [windows](#)
[work](#)

Obrázok 5: Tag cloud

Váha tagu je vlastne počet obsahov, ktorým bol priradený. V prípade konkrétneho obsahu, môžeme takto zobrazit' tagy, ktoré sme použili a tým definovať príbuzné obsahy.

Inou možnosťou je použiť tagy všetkých obsahov, čo reprezentuje istý druh vyhľadávania. V tomto prípade sú tagy v *tag cloud* dvoj-stavové ovládacie elementy. Pri kliknutí na niektorý tag je tento zvýraznený, pri opakovanom kliknutí je zvýraznenie odstránené. Množina zvýraznených tagov definuje množinu obsahov. Pre všetky tagy, ktoré nie sú zvýraznené, ale sú relevantné k danej množine obsahu, sa spočíta váha a sú zobrazené rovnako ako v prvom prípade.

Využívanie tagov získavajú na popularite, v neposlednej rade aj vďaka špecializovaným vyhľadávacím službám napríklad Technorati alebo službe Digg.

Tabuľka 14: Podpora tagov v CMS

CMS	Podpora
Bricolage	- Kľúčové slová
Drupal	Áno, ľubovoľný počet nezávislých množín tagov.
EzPublish	- Kľúčové slová
Mambo	- Kľúčové slová, nedovoľuje výber z existujúcej množiny.
Plone	- Kľúčové slová, pri vytváraní je možné vybrať z lineárneho zoznamu existujúcich tagov.
Wordpress	Áno - Poskytuje i rozhranie pre služby automatickej kategorizácie. - Tag cloud modul.

Automatická klasifikácia obsahu

Problém klasifikácie je prinútiť autorov správne a dostatočne kategórie a tagy využívať. Nakoľko sémantická analýza textu je zložitý problém, automatizované klasifikovanie sa musí spoliehať na jednoduchšiu analýzu textu obsahu. Priradenie kategórie obsahu je úlohou klasifikačnej funkcie. Uvedme jednoduchý príklad klasifikačnej funkcie. Kategórie sú reprezentované množinou slov a fráz. Klasifikačná funkcia vyhodnocuje relevantnosť obsahu vzhľadom na konkrétnu kategóriu. Relevantnosť definujeme ako frekvenciu výskytov slov a fráz (priradených kategórii) v konkrétnom obsahu. Priamočiaro by sme mohli do klasifikačnej funkcie doplniť výrazy, napríklad pomocou boolovských spojok AND a NOT. Výsledkom by bola automatická alebo semi-automatická klasifikácia. V druhom prípade by autor vybral vhodnú kategóriu zo zredukovanej, prípadne podľa relevancie zoradenej množiny kategórií.

Zaujímavý prístup zvolili autori [17]. Kategóriám sú priradené množiny matic. Prvý stĺpec matice je zoznam slov, druhý stĺpec predstavuje frekvencie ich výskytov v konkrétnom obsahu (v tomto prípade webovej stránke). Matice v kategóriách sú pred-vytvorené z obsahov v existujúcej kategorizácii (pre každý obsah jedna matica). Klasifikačná funkcia následne porovná rovnako vytvorenú maticu nového obsahu s maticami jednotlivých kategórií.

V súčasnosti poskytujú službu automatickej klasifikácie napríklad spoločnosť Tagyu a Yahoo (Wordpress). Samozrejme problém klasifikácie senzitivných dát nieje možné riešiť takýmto spôsobom.

V štádiu príprav sa nachádza rozširujúci modul pre Drupal, ktorý dodáva funkciu definovania vzťahov. Založený je na štandardoch RDF a OWL. RDF sme už definovali spomeňme, že hlavnou myšlienkou je definícia relácií medzi zdrojmi. Web Ontology Language – OWL je štandard, ktorý stavia na RDF a umožňuje okrem iného definovať vzťahy medzi reláciami, napríklad *inverziu* alebo *tranzitivnosť*. Implementácia OWL by umožnila dedukovať nové vzťahy, a teda automaticky vytvárať prepojenia medzi obsahmi.

Časová os

Dôležitou štruktúrou obsahu je umiestnenie v čase, hlavne pre obsah, ktorý ma charakter správ. Hlavným predstaviteľom sú blogy. V najjednoduchšej forme ide o usporiadanie čiastkových obsahov na stránke v chronologickom poradí. Pokročilejšou funkciou je vytváranie interaktívnych kalendárov, ktoré sú prepojené na archívy obsahu, podľa dňa, týždňa, či mesiaca.

Keďže tento druh obsahu je uverejnený iba určitú dobu, nemá presné miesto v statickej štruktúre. V prípade, že sa chceme odkázať na konkrétny obsah, nieje možné použiť aktuálnu adresu – URL. Riešením sú trvalé odkazy - *permalinky*. *Permalink* predstavuje webovú adresu, ktorá odkazuje na konkrétny obsah a to tak v prípade, že je správa aktuálne zverejnená, alebo sa už nachádza v archíve. Systémy Wordpress, Serendipity umožňujú formát *permalinkov* konfigurovať, napríklad tak, že url obsahuje dátum zverejnenia. *Permalink* však môžeme chápať aj v inom zmysle. Napríklad v prípade vznikajúcej špecifikácie, majú jednotlivé verzie priradený vlastný, jedinečný odkaz. *Permalink* špecifikácie potom predstavuje odkaz na najaktuálnejšiu verziu.

Implementácia *Permalink* predstavuje vo wiki značný problém nakoľko daná verzia môže byť zmazaná. Príkladom je problematické citovanie článkov Wikipédie, vzhľadom na premenlivý charakter wiki.

Zaujímavým využitím histórie wiki vzhľadom na časovú os, je *návrat v čase*. Užívateľ definuje dátum, následne CMS prezentuje verzie wiki-stránky, ktoré boli aktuálne k danému okamihu.

Tabuľka 15: Podpora dynamických štruktúr

CMS	Časová os
Drupal	- Áno - Poskytuje modul kalendár. - Permalink.
Mambo	Áno
MediaWiki	- Nové a staré wiki-stránky. - Permalink podľa jedinečného identifikátora wiki-stránky.
MoinMoin	Nie - Permalink pre wiki-stránku využíva verziu.
Plone	- Áno - Kalendár
TikiWiki	- Pre komponenty <i>blog</i> a <i>article</i> . - Permalink pre wiki-stránku využíva verziu. - Permalink pre <i>blog</i> .
Typo3	- Pre modul <i>news</i> .
Wordpress	Áno - Modul kalendár je dodávaný ako súčasť inštalácie. - Permalink

3 Vstup Obsahu

Interakcia medzi užívateľom, obsahom a CMS sa skladá z týchto častí:

1. Komunikačný kanál, HTTP, WebDAV, FTP.
2. Formát (v zmysle MIME) obsahu, text, HTML, OpenDocument.
3. Aplikácia pre vytváranie/modifikáciu obsahu v danom formáte.

Definícia: Konkrétny výber jednotlivých častí nazývame *interakčná schéma*.

Úlohou CMS je:

1. Implementácia konca komunikačného kanálu.
 - Z pravidiel implementuje CMS jeden koniec a klientská aplikácia druhý koniec kanálu. Pod klientskou aplikáciou myslíme aplikáciu na užívateľskom počítači, ktorá môže byť súčasťou aplikácie na vytváranie obsahu, dodávaná nezávisle, alebo súčasťou operačného systému.
 - Niektoré CMS (Plone) dodávajú aj klientskú aplikáciu, prakticky teda implementujú oba konce komunikačného kanálu.
2. Konverzia formátov. Konverzia formátov sa skladá z výberu elementov komponentu obsahu ako aj samotnej konverzie formátovania a štruktúry obsahu z vnútorného formátu CMS do cieľového formátu (a naopak). Z hľadiska výberu implementujú CMS tieto možnosti:
 - ktorýkoľvek element
 - konkrétny element, (v závislosti od typu obsahu)
 - celý komponent obsahuV prípade jednoduchých CMS nieje konverzia potrebná, keďže výber elementu je implicitný (existuje iba jeden) a vnútorný a cieľový formát je rovnaký (napríklad HTML, v prípade DMS napríklad MS Office, alebo OpenDocument).
3. V prípade textového obsahu je aspoň jednoduchá aplikácia súčasťou CMS.

V tejto časti sa pozrieme na interakčné schémy, ktoré CMS poskytujú na vstup obsahu. Pod pojmom vstup, myslíme tak vytváranie nového, ako aj modifikovanie či agregáciu existujúceho obsahu.

3.1 Aplikácie pre vstup obsahu

V tejto kapitole sa pozrieme na možnosti pre vstup obsahu cez webové rozhranie. Pod pojmom webové rozhrania, chápeme rozhranie webového prehliadača. Niektoré metódy využívajú štandardne funkcie dostupné prakticky na všetkých typoch webových prehliadačoch, iné využívajú špeciálne moduly.

- Webové formuláre
 - Pokročilé funkcie dodávané ako *ECMAScript* aplikácia
- Java Applet, Adobe Flash aplikácie, vyžadujú rozširujúci modul, dodávaný pre všetky populárne prehliadače.
- Riešenia závislé na type webového prehliadača.
 - Microsoft Internet Explorer: ActiveX modul
 - Mozilla Firefox: XPI modul

3.1.1 Jednoduché webové formuláre

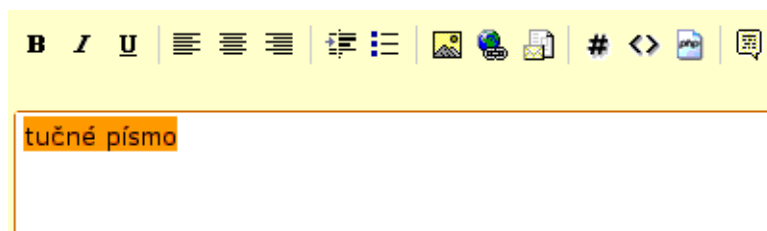
V podstate ide o jednoduché rozšírenia štandardného HTML elementu *TextArea*. Keďže *TextArea* element je určený pre vstup čistého (*plain*) textu, užívateľ musí použiť špeciálny jazyk pre definovanie štruktúry (prípadne formátovania). Najčastejšie ide o rovnaký jazyk, aký používa CMS pre komplexné elementy, teda XHTML, wiki, BBCode.

Definícia: DOM Document Object Model je platformovo a jazykovo neutrálne rozhranie, ktoré dovoľuje programom a skriptom dynamicky pristupovať a meniť obsah, štruktúru a formátovanie dokumentov [8]. V súčasnosti existujú tri na seba nadväzujúce štandardy DOM: DOM Level 1, 2 a 3.

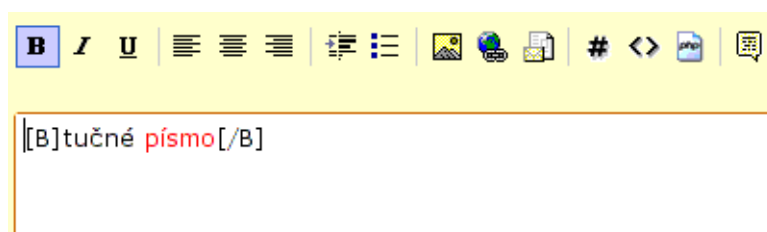
DOM vznikol ako snaha štandardizovať existujúce, navzájom nekompatibilné riešenia webových prehliadačov. Tieto sú označované ako DOM Level 0, príkladom je napríklad DHTML spoločnosti Microsoft. Nutno dodať, že i podpora štandardov ako *DOM* sa líši tak medzi rôznymi prehliadačmi ako aj medzi rôznymi verziami konkrétneho prehliadača.

Definícia: DHTML, Dynamic HTML je termín využívaný niektorými výrobcami webových prehliadačov. Popisuje kombináciu HTML, CSS a (určitého) skriptovacieho jazyka, ktorá dovoľuje dynamickú modifikáciu HTML dokumentu.

Panel nástrojov je množina ovládacích prvkov, spravidla tlačidiel. Funkcia nástroja – respektíve zodpovedajúceho ovládacieho prvku, je napríklad vloženie formátovacích znakov pre tučné písmo. Dynamické vkladanie znakov využíva *ECMAScript* a *DOM*. Jednoduchým rozšírením panela nástrojov je vloženie začiatočného a ukončovacieho znaku alebo znakov pred a za aktuálne zvýraznený text.



Obrázok 6: Panel nástrojov



Obrázok 7: Panel nástrojov, tučné písmo

Napriek tomu, že ide na prvý pohľad o relatívne jednoduchú funkciu, na samotné získanie zvýrazneného textu je nutné použiť jednu z troch metód, podľa použitého prehliadača:

Výpis 9: Získanie zvýrazneného textu

1.1	<code>window.getSelection()</code>	2.	Mozilla/Firefox, Safari
3.2	<code>document.getSelection()</code>	4.	Mozilla/Firefox, Opera, IE pre Apple OSX
5.3	<code>document.selection.createRange.text()</code>	6.	IE 5, 6 pre MS Windows

Okrem formátovania textu, poskytujú panely nástrojov i špeciálne dialógy pre vkladanie webových odkazov, obrázkov, tabuliek, alebo mapu znakov (nástroj na vkladanie národných, respektíve špeciálnych znakov: Á, © Copyright, € Euro). Vo všeobecnosti panel nástrojov rieši problém objavenia konštrukcií jazyka vložení kostry. Častá je i načrtnutá podpora zvýrazneného textu. (V prípade TikiWiki funguje napríklad pre tučné písmo, ovládací prvok pre tabuľky však zvýraznenie ignoruje a vždy vloží kostru novej tabuľky.)

Výhody:

- Jednoduchá implementácia, vstup možný z prakticky každého webového prehliadača.

Nevýhody:

- Potreba zaškoliť užívateľov v používaní jazyka. Riešenie: online dokumentácia, panel nástrojov.
- Nesprávny vstup. Nekorektné použitie syntaxe jazyka. Výsledok konštrukcií jazyka nieje zrejмый. Riešenie: (online) náhľad.

Tabuľka 16: Jednoduché webové formuláre

CMS	Panel nástrojov	Online dokumentácia	Náhľad	Chybový oznam
Bricolage	Nie	Áno	Nie	Áno
Drupal	Áno, ale len pre BBCode	Áno	Áno	Nie, nekorektná syntax je odstránená bez oznamu.
ezPublish	Nie	Nie	Áno	Áno
Magnolia	Nie	Nie	Nie	Nie
Mambo	Nie	Nie	Áno	Nie
MediaWiki	Áno	Áno	Áno	Nie
Plone	Nie	Nie	Nie	Nie
Tikiwiki	Áno	Áno	Nie	Nie
Typo3	Nie	Nie	Nie	Nie
Wordpress	Áno	Nie	Nie	Nie, akceptuje nevalidný vstup.

3.1.2 Komplexné webové formuláre

Nástroje z tejto množiny sú založené na WYSIWYG prístupe. Štruktúra a formátovanie sú viditeľné priamo v texte, ktorý užívateľ vytvára. Syntax konkrétneho jazyka je skrytá, namiesto toho používa užívateľ ovládacie prvky nástroja, napríklad tlačidlá. Spoločnou črtou webových formulárov je využitie *ECMAScript* a *DOM*. Spomeňme ešte, že nástroje tejto skupiny sa označujú ako RichText Editor (RTE).

Existujú dva druhy nástrojov:

1. Založené na podpore *execCommand*.
2. Nástroje využívajúce *DOM*.

Nástroje založené na *execCommand*

V prvom prípade obsluhuje vstup znakov z klávesnice ako aj vytváranie HTML elementov webový prehliadač, pričom nástroje dodávajú iba rozširujúce funkcie (napríklad filtrácia, komunikácia so CMS).

Definícia: *execCommand* je funkcia poskytovaná webovým prehliadačom, ktorú je možné využiť z vykonávaného skriptu v prehliadači. Podľa parametrov, s ktorými je vykonaná, mení vzhľad alebo štruktúru aktuálne zvýrazneného obsahu, prípadne tieto zmeny vykoná pre následne vkladajúci obsah.

Využitie tejto funkcie je podmienené nastavením premennej *contentEditable*, respektíve *designMode* na *true*. Jednoduchý WYSIWYG nástroj nájde užívateľ v [10].

Výpis 10: Použitie funkcie *execCommand*

1 <html> 2 <head> 3 <script> 4 doc = document.getElementById('edit').contentWindow.document; 5 doc.designMode = "on";	HTML element môže byť modifikovaný.
6 doc.execCommand('createLink', false, 'www.uniba.sk');	Vytvorí element odkaz a vloží ho do edit elementu.
7 doc.execCommand('bold', false, null) ;	Nastaví tučné písmo pre aktuálne zvýraznený text.
8 doc.execCommand('formatBlock', false, '<h1>') ;	Zvýraznený text je formátovaný ako nadpis prvej úrovne.
9 </script> 10 </head> 11 <body> 12 <iframe id="edit" width="200px" height="200px"></iframe> 13 /body> 14 </html>	HTML element, do ktorého webový prehliadač vkladá text a HTML elementy podľa predchádzajúcich funkcií.

Tabuľka 17: Podpora WYSIWYG XHTML nástrojov vo webových prehliadačoch

Nástroj	Internet Explorer	Mozilla / Netscape	Firefox	Safari (OS X)	Opera
TinyMCE 2.1.0 tinymce.moxiecode.com	5.5+	1.8 (Seamonkey 1.0)	1.5+	2.X častočne	9.0 častočne
Xinha xinha.python-hosting.com/	5.5+	1.3	1.0+	-	
FCKEditor 2.4.2 www.fckeditor.net	5.5+	1.3+ / 7.0+	1.0+	-	-
Kupu 1.3.9 http://kupu.oscom.org	5.5+	1.3.1+	1.0+	-	-
Ez Online Editor ³ súčasť ezPublish CMS	5.0+	1.3+/7.0+	1.0+	-	-

3 Licencia k nástroju nie je jasná.

Poznámka: Podporu akcií Cut/Copy/Paste je nutné vo Firefox-e explicitne povoliť (pričom je nutné ručne editovať konfiguračný súbor *user.js*).

FCKEditor: V konfiguračnom súbore (*fckconfig.js*) je možné zmeniť CSS súbor, ktorý nástroj používa pre dizajn textu. Súčasne je možné definovať povolené druhy písma (*font*), farby a veľkosti písma.

TinyMCE: Centrálnou myšlienkou konfigurácie sú *theme*. Na úrovni jednotlivých *theme* je možné nezávisle definovať množinu ovládacích prvkov.

Výhody:

- Nástroje sú vzhľadom a funkciami podobné kancelárskym aplikáciám, takže využívajú existujúce schopnosti užívateľov.
- Užívateľ nemôže vytvoriť nekorektný obsah (vzhľadom na syntax jazyka).
- Nakoľko užívateľ vidí výsledok operácie (tj. vloženie konkrétnej syntaktickej konštrukcie na pozadí) okamžite, nie sú potrebné opakované cykly: vstup->náhľad->vstup ako v prípade jednoduchých formulárov.
- Často je možné nástroje integrovať s CMS a poskytnúť ďalšie funkcie ako zjednodušené vkladanie existujúcich obrázkov a odkazov na obsah v rámci CMS.

Nevýhody:

- Výstup týchto nástrojov je XHTML. Je teda nutné zabezpečiť konverziu formátov v prípade, že CMS nevyužíva komplexné elementy s XHTML jazykom.
- Nakoľko nástroje používajú vnútorne XHTML je obsah často obohatený nielen o štruktúrne ale aj formátovacie informácie. Riešenie: Konfigurácia nástroja pre konkrétne použitie, tj. vynechanie ovládacích prvkov pre formátovanie.
- Keďže formátovanie publikácie ovplyvňujú i ďalšie faktory, výsledok nie je zhodný s tým čo užívateľ vidí (na rozdiel od kancelárskych aplikácií). Na tento problém je nutné užívateľov upozorniť.

Nástroje založené na DOM

Nástroje tohto typu využívajú objekty, funkcie a udalosti *DOM*. Tak vstup znakov ako aj vytváranie elementov sú implementované v *ECMAScript*. Veľkou výhodou týchto nástrojov je, že dovoľuje v rámci webového prehliadača používať ľubovoľný jazyk XML, kým prvé riešenie je previazané s HTML. Nato aby sme mohli reálne použiť XML jazyk v nástroji je potrebné:

- Definovať slovník XML elementov a ich vzťahov (napríklad aké elementy môže element obsahovať a v akom poradí). Keďže túto úlohu plnia XML schéma jazyky, najčastejšie nástroje implementujú ich podporu.
- Definovať ovládacie prvky a iné funkcie nástroja. V tomto prípade existuje viac možností, často sa používa rozšírenie XML schémy o vlastné elementy.
- Definovať zobrazenie XML elementov. Znovu sa využíva niektorý so štandardných jazykov určených na prezentáciu XML: CSS alebo XSLT.

V kapitole Komponent Obsahu sme hovorili o potrebe definovať komponent pomocou otvoreného štandardu. Ako je vidieť v tvorcovia CMS by mohli všetky tri hore uvedené body automatizovať alebo aspoň značne zjednodušiť, tým že by definícia komponentu bola v Relax NG schéme.

Výpis 11: Zobrazenie XML vo webovom prehliadači, pre Internet Explorer 6.0 a Firefox 1.5

```

<html XMLNS:xx>
<head>
<style>
    xx\:test { display:block; position:absolute; left:500;top:100;width:100;height:100;
border:solid; }
</style>
</head>
<body>
    <xx:test>
        some test
    </xx:test>
</body>
</html>

```

Tabuľka 18: Podpora WYSIWYG XML nástrojov vo webových prehliadačoch

Nástroj	Internet Explorer	Mozilla / Firefox	XHTML	XML
BXE 1.0 http://bitfluxeditor.org/ nástroj je založený na Mozile a Kupu.	-	1.5+	Áno	Áno, je potrebné dodať CSS súbor a Relax NG schému
Mozile 0.7	-	1.5+	Áno	-
Mozile 0.8a http://mozile.mozdev.org/	6.0+	1.5+	Áno	Áno, je potrebné dodať CSS súbor a Relax NG schému

Výhody:

- Obsah má pevne definovanú štruktúru, užívateľ nemôže vytvoriť nekorektný obsah a to tak vzhľadom na syntax ako aj štruktúru jazyka.
- Využitie ľubovoľného XML jazyka.
- Založené na štandardných a overených technológiách.
- Nakoľko užívateľ vidí výsledok operácie (tj. vloženie konkrétnej syntaktickej konštrukcie na pozadí) okamžite, nie sú potrebné opakované cykly: vstup->náhľad->vstup ako v prípade jednoduchých formulárov.
- Často je možné nástroje integrovať s CMS a poskytnúť ďalšie funkcie ako zjednodušené vkladanie existujúcich obrázkov a odkazov na obsah v rámci CMS.

Nevýhody:

- Pomerne komplexné nástroje, nástroj je nutné doplniť o definície schémy a prezentácie jazyka.
- Problémy validácie:
 - definícia „prázdneho“ obsahu, teda základnej kostry (môže byť niekoľko rôznych).
 - definícia „prázdneho“ elementu a problém komplexnej štruktúry elementu, napríklad: Relax NG aj XML Schema podporuje definíciu dátového typu elementu pomocou regulárnych výrazov.
- Keďže formátovanie publikácie ovplyvňujú i ďalšie faktory, výsledok nie je zhodný s tým čo užívateľ vidí (na rozdiel od kancelárskych aplikácií). Na tento problém je nutné užívateľov upozorniť.

Tabuľka 19: Porovnanie podporovaných WYSIWYG nástrojov a ich integrácie

CMS	WYSIWYG nástroj	Integrácia	Poznámky
Bricolage	Xinha		Minimálna integrácia.
Drupal	TinyMCE	Konfigurácia priamo v CMS, integrácia do systému profilov a užívateľských rolí. Rolám môžu byť priradené <i>theme</i> .	Integrácia s modulom <code>img_assist</code> pre vkladanie obrázkov.
EzPublish	Ez Online Editor	Súčasťou štandardnej inštalácie.	Integrácia s vkladáním obrázkov, súborov a odkazov. Transparentný preklad medzi ezXML a XHTML.
TikiWiki	V štádiu príprav		
MoinMoin	FCKEditor	Súčasťou štandardnej inštalácie.	Transparentný preklad medzi XHTML a wiki jazykom.
Magnolia	Kupu		Integrácia s vkladáním obrázkov, súborov a odkazov.
Mambo	TinyMCE	Súčasťou štandardnej inštalácie.	Integrácia s vkladáním obrázkov a súborov.
	BXE	Experimentálny modul <code><docbook: collab></code>	Dodáva sa s podporou pre DocBook
Plone	Kupu	Súčasťou štandardnej inštalácie	Integrácia s vkladáním obrázkov, súborov a odkazov.
	Epoz (predchodca Kupu)		
	FCKeditor	Modul FCKEditor.Plone, konfigurácia v CMS.	
Typo3	Dodáva vlastný nástroj ⁴ . Modul pre TinyMCE aj FCKEditor (beta verzie).	Súčasťou štandardnej inštalácie.	Integrácia s vkladáním externých súborov a obrázkov.
Wordpress	TinyMCE	Súčasťou štandardnej inštalácie.	Integrácia s vkladáním obrázkov a súborov.

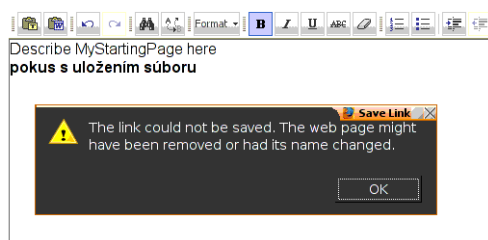
3.1.3 Problémy webových formulárov

Komunikácia medzi CMS a počítačom užívateľa môže byť prerušená, je teda nevyhnutné aby užívateľ mohol vytvorený obsah uložiť na lokálnom počítači a neskôr pokračovať. V prípade použitia špeciálneho jazyka môže užívateľ štandardným spôsobom vložený text skopírovať do lokálneho súboru. V prípade WYSIWYG je kopírovanie problematické, podpora závisí od konkrétneho webového prehliadača.

Podpora pre ukladanie obsahu je zabudovaná v prehliadači Internet Explorer. Využíva ju FCKEditor avšak napríklad konfigurácia v MoinMoin túto funkciu neposkytuje. Alternatívne môže užívateľ využiť funkciu *zobrazenie HTML*, a následne použiť bežnú funkciu webového prehliadača pre uloženie stránky. S ohľadom na bežného užívateľa, túto možnosť nepovažujeme za dostatočnú. V prípade, že užívateľ využije možnosť uloženia celej webovej stránky, narazí na problém, že webový prehliadač uloží východziu stránku, alebo stránku, ktorú si na pozadí znovu vypýta od CMS. Táto stránka však neobsahuje vytvorený

4 Kompatibilný len s Internet Explorer

respektíve zmenený obsah, dokonca pri vytváraní nového obsahu v MoinMoin ani nemusí existovať (Obrázok 8).



Obrázok 8: MoinMoin webová stránka nemôže byť uložená

Žiadny nástroj neumožňuje obnoviť uložený obsah a pokračovať práci s obsahom vo WYSIWYG nástroji. Z bezpečnostných dôvodov nemá skript, ktorý je vykonávaný vo webovom prehliadači prístup k lokálnym súborom, preto nieje možné túto funkciu implementovať priamočiaro.

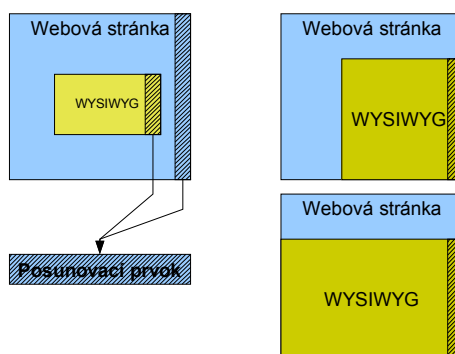
Možné riešenia:

- V CMS by existovala webová stránka *Otvoriť Súbor*, ktorá by obsahovala štandardný HTML formulár pre posielanie súboru. Užívateľ by vložil uložený súbor a odoslal formulár. Po prijatí formuláru, by CMS použil prijatý obsah súboru a vygeneroval stránku z WYSIWYG nástrojom, ktorý by bol pred-vyplnený obsahom súboru.
- Zložitejšia, ale užívateľsky príjemnejšou možnosť by bola pri ukladaní do výsledného súboru uložiť okrem konkrétneho obsahu aj odkaz priamo na špeciálnu stránku. Užívateľ by následne iba otvoril uložený html súbor vo webovom prehliadači a bol by automaticky presmerovaný na stránku s WYSIWYG nástrojom. Táto možnosť však naráža na problém s autentifikáciou užívateľa.
- Podpora zo strany webového prehliadača. V súčasnosti existuje štandard *Load And Save*, ktorý je súčasťou štandardu *DOM Level 3* [11]. Tento adresuje práve potrebu ukladaní súborov na strane klienta, ide však o riešenie všeobecnejšie, neobmedzuje sa len na potreby WYSIWYG nástrojov. V čase písania tejto práce však žiaden z populárnych webových prehliadačov, tento štandard nepodporuje. Pokiaľ je CMS nasadený len v intranete, teda v kontrolovanom prostredí a súčasne sa využíva webový prehliadač MS Internet Explorer, existuje ešte iná možnosť. Internet Explorer dovoľuje rozdeliť webové stránky do zón (*zone*) (Podľa URL respektíve fragmentov URL). Ku každej zóne je možné priradiť stupeň bezpečnosti. V prípade zóny *Intranet* je explicitne povolený prístup k súborovému systému hositeľského operačného systému z vykonávaného skriptu vo webovom prehliadači. Týmto sa otvára možnosť zapisovať a čítať uložené súbory. Opakujeme, že vzhľadom na bezpečnostné riziká spojené s touto technikou odporúčame použitie len v kontrolovanom prostredí.

Ďalším problémom, ktorý nieje obmedzený len na WYSIWYG nástroje je problém, ktorý môžeme definovať ako *dokument v dokumente*.

- WYSIWYG nástroj preberá veľkosť HTML elementu, do ktorého bol vložený. Často však voľné miesto nieje postačujúce pre vytváranie alebo modifikáciu dlhšieho textu. V systéme Wordpress má užívateľ možnosť zväčšiť plochu WYSIWYG nástroja. (Podobne i v TikiWiki aj keď v tomto prípade nejde o plnohodnotný WYSIWYG nástroj.) Z užívateľského hľadiska však ide o relatívne ťažko objaviteľnú funkciu.
- V niektorých prípadoch môže nastať situácia, kedy sú súčasne viditeľné dva posunovacie elementy (*scrollbar*). Tento problém je často ešte zhoršený neintuitívnym

spracovaním posunovacích príkazov. Napríklad vo webovom prehliadači je posúvaný buď obsah WYSIWYG nástroja, alebo celá webová stránka, v závislosti od polohy myši.



Obrázok 9: Vloženie WYSIWYG nástroja do webovej stránky.

Príkladom riešenia je napríklad projekt *Writely*. WYSIWYG nástroj zaberá okrem panela nástrojov celú plochu webového prehliadača (viď Obrázok 9 v pravo dole). Typo3 obsahuje špeciálny ovládací prvok, ktorý otvorí novú inštanciu webového prehliadača, ktorý rovnako ako *Writely*, obsahuje iba panel nástrojov a WYSIWYG nástroj. Užívateľ môže následne použiť relatívne malú inštanciu ako náhľad, respektíve pre vykonanie drobných úprav, kým v prípade väčšieho zásahu alebo vytvárania nového obsahu použije prvú možnosť.

V článku [12] poukazujú autori, že väčšina užívateľov pri presúvaní textu v rámci uceleného obsahu, postupujú metódou vymazania a následného vloženia. Autori prišli k záveru, že tento postup volia užívatelia práve z dôvodu malej plochy nástroja pre vkladanie textu.

3.1.4 Iné aplikácie

Tabuľka 20: Iné WYSIWYG webové aplikácie

Adobe Flash	Link	Licencia
Obedit	http://www.oblius.com/?projects.obedit	LGPL
eWebWP	http://www.ektron.com/ewebwp.aspx	Komerčný produkt
Java		
Ekit	http://www.hexidec.com/ekit.php	LGPL
Editize	http://www.editize.com	Komerčný produkt

Širší prehľad WYSIWYG nástrojov nájde čitateľ v [9].

Plone dodáva aplikáciu *external editor*. Funguje v role sprostredkovateľa medzi CMS a aplikáciami na počítači užívateľa. Užívateľ teda môže kliknúť na tlačítko modifikácie v CMS a následne sa spustí natívna aplikácia na jeho počítači. Na pozadí však kliknutie ako aj spustenie obsluhuje *external editor*. Počas modifikácie je komponent na strane CMS chránený, po ukončení aplikácie je automaticky poslaný späť do CMS.

3.2 Multimediálny obsah

Do tejto skupiny zaraďujeme obrazový, video a audio obsah. Pozrieme sa aké funkcie poskytujú CMS vzhľadom na vstup obsahu, spracovanie metadát ako aj agregáciu tohoto druhu obsahu v rámci textových komponentov.

Existujú v zásade dve možnosti.

1. Prvou je takzvaná **príloha**, k existujúcemu (respektíve vytváranému) obsahu sú priložené multimediálne súbory. Do textu sú následne vložené odkazy na priložené súbory, ktoré sú pri prezentácii nahradené ich multimediálnou podobou.
2. Použiť **centrálne úložisko súborov**.

Tabuľka 21: Vstup multimediálneho obsahu

CMS	Metóda	
Bricolage	Centrálne úložisko	- Poskytuje bohaté možnosti pre správu (napríklad workflow).
Drupal	Hybridný prístup, modul img_assist	- Obrázky sú pripojené ku konkrétnemu obsahu, je však možné použiť aj existujúce obrázky (teda také, ktoré boli v minulosti pripojené k nejakému obsahu).
EzPublish	Príloha	- Predstavuje komponent obsahu.
Mambo	Centrálne úložisko	- Umožňuje vytvárať hierarchiu súborových zložiek. - Spoločné pre všetkých užívateľov - Pri vytváraní textového komponentu, užívateľ vyberá len z existujúcich súborov.
Magnolia	Centrálne úložisko	
MediaWiki	Centrálne úložisko	- Najčastejšie implementované ako samostatný namespace.
MoinMoin	Príloha	
Plone	Centrálne úložisko	
TikiWiki	Centrálne úložisko Príloha	- Nevýhodou centrálneho úložiska je, že užívateľ musí vložiť URL do textového obsahu. - Ak použijeme metódu prílohy, multimediálne súbory sú spravované samostatne, ako súčasť komponentu.
Typo3	Centrálne úložisko	- Umožňuje vytvárať hierarchiu súborových zložiek. - Je možné definovať vstupný bod do hierarchie pre konkrétnych užívateľov a tým poskytnúť nezávislé množiny multimediálnych súborov.
Wordpress	Hybridný prístup	- Vstup multimediálnych súborov je možný len cez nástroj pre vytváranie textového obsahu, súčasne umožňuje použiť i existujúce súbory.

3.3 Súborovo orientované protokoly

Jedným z prvých konceptov, s ktorým sa užívatelia pri používaní počítača stretnú, je koncept súborov a súborových adresárov. Informácie sú uzavreté v súboroch, ktoré sú hierarchicky usporiadané v adresároch. Súčasne sú na tomto princípe založené i aplikácie na vytváranie obsahu. Je teda pochopiteľné, že sa autori CMS snažia sprístupniť obsah takýmto spôsobom. Riešením je sprístupnenie obsahu vo forme virtuálnej hierarchie súborov pomocou súborovo orientovaných protokolov. Dôraz je na slovo virtuálnej, tak súbory ako aj zložky môžu ale nemusia v rámci CMS existovať. Vďaka tomu môžu CMS prezentovať konkrétny fyzický obsah ako niekoľko virtuálnych súborov (možný je aj opačný prístup, spojenie niekoľkých obsahov do jedného virtuálneho súboru). Najčastejšie a najpraktickejšie riešenie je mapovanie:

- Komponent – **súbor** (alebo niekoľko súborov reprezentujúcich rovnaký obsah v rôznych formátoch).
- Vonkajšia štruktúra obsahu (napríklad kategórie) – **adresár**.

Medzi najznámejšie a najrozšírenejšie patria protokoly:

- FTP
- WebDAV
- CIFS (SMB)

3.3.1 FTP

Predstavuje najstarší protokol pre vzdialený prístup k súborovému systému.

Výhody:

- Rozšírený protokol.
- Relatívne jednoduchý protokol.

Nevýhody:

- Nepodporuje šifrovanie, tak prihlasovanie ako aj dátový prenos prebieha cez nezabezpečený kanál (existuje rozšírenie protokolu, ktoré používa SSL – FTPS)
- Pôvodne bol navrhnutý ako interaktívny protokol, s čoho vyplýva pomerne zložitá implementácia niektorých funkcií ako zoznam súborov v adresáre, podpora symbolických odkazov atď.
- Minimálna podpora atribútov súborov.
- Nepodporuje uzamknutie súboru.
- Nepodporuje *Unicode* (existuje rozšírenie).

3.3.2 WebDAV

Je množina rozšírení HTTP protokolu, s cieľom umožniť skupine užívateľov spoluprácu pri vytváraní a spravovaní súborov na vzdialenom web-serveri.

Bol publikovaný vo februári 1999, ako RFC 2518 *HTTP Extensions for distributed authoring* [13]. Medzi rozšírenia WebDAV patrí DeltaV protokol, ktorý pridáva vzdialenú správu verzií. Nakoľko správa verzií patrí medzi kľúčové funkcie wiki, bolo by zaujímavé sprístupniť ju pomocou DeltaV protokolu.

K odstráneniu potenciálnych konfliktov pri súčasnej modifikácii obsahu, pomocou WebDAV a webového rozhrania, je nevyhnutné zabezpečiť transparentné uzamykanie. Teda uzamknutie súboru cez WebDAV protokol musí byť viditeľné vo webovom rozhraní.

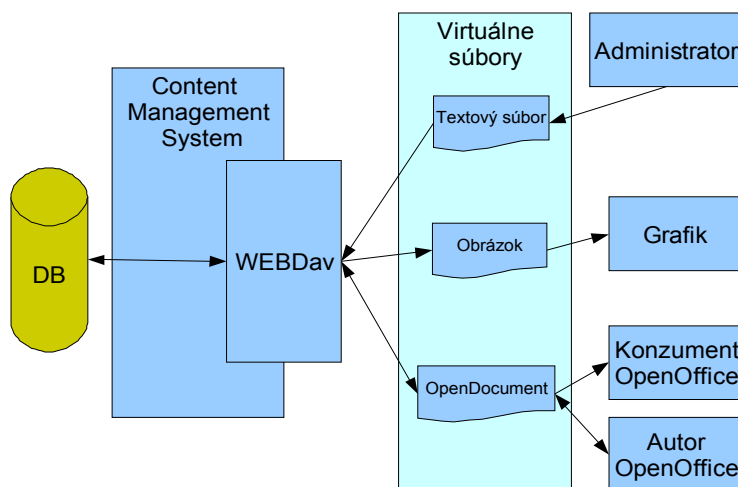
Na okraj spomeňme, že existuje istá miera podobnosti medzi XML-RPC, SOAP a WebDAV. Všetky protokoly využívajú HTTP a XML k prenosu dát. Rozdiel je v spôsobe použitia, kým XML-RPC a SOAP sú metódy pre všeobecnú komunikáciu aplikácií, WebDAV je špecializovaný protokol.

Výhody:

- Keďže WebDAV je rozšírením HTTP, má tento protokol, automaticky všetky jeho výhody:
- Bezpečný šifrovaný kanál - *SSL*,
- Podpora proxy serverov, prechod cez *firewall* (nakoľko využíva štandardný HTTP port 80).
- Jednoduchý protokol, vďaka čomu je možné ho implementovať aj v jazykoch ako PHP.
- Dovoľuje asociovať súbor s množinou atribútov.
- Kolaboráciu pri vytváraní obsahu, zabezpečujú zámky, uzamknutý súbor nie je možné modifikovať.

Nevýhody:

- Pomerne málo rozšírený. Podpora pokročilejších funkcií WebDAV je stále málo rozšírená, problémom je hlavne správa metadát a zámok. Rozšírenie WebDAV protokolu o služby vyhľadávania DASL existuje len vo forme konceptu.



Obrázok 10: WebDAV

3.3.3 CIFS

Common Internet File System predstavuje najkomplexnejší súbor orientovaný protokol. Pochádza od spoločnosti Microsoft, v súčasnosti je však podporovaný aj na ďalších platformách. V systéme Microsoft Windows sa CIFS využíva plne transparentne na sprístupnenie súborov v sieti (v doméne alebo pracovnej skupine), aplikácie prakticky nevedia, že prístupujú k súboru na vzdialenom počítači.

V našom prehľade CIFS nepodporuje žiadne CMS, populárnym CMS (respektíve skôr DMS) s podporou CIFS je *Alfresco*⁵.

Výhody:

- Rozšírený, transparentná podpora na mnohých OS.
- Poskytuje tak šifrovanie ako aj podpisovanie správ.
- Uzamknutie súboru alebo časti (podporujú aplikácie MS Office).
- Transakcie.
- Náhodný prístup v rámci súboru, FTP a WebDAV túto funkciu nepodporujú.
- *Unicode* pre mená súborov.

Nevýhody:

- Extrémne komplexný protokol, s mnohými verziami a rozšíreniami.
- Staršie verzie boli málo bezpečné (heslá prenášané ako text, slabé hašovacie algoritmy).
- Nerozlišuje veľké a malé písmená v menách súborov (SAMBA poskytuje rozšírenie, ktoré rozlišovanie umožní).

Tabuľka 22: Podpora protokolov v OS

Protokol	MS Windows	Apple OS X	Linux (Unix)	Aplikácie
FTP	Web adresár	Finder	KDE: zabudovaná	OpenOffice
WebDAV	NetDrive ⁶ , WebDrive		podpora GNOME: nautilus	
CIFS	Zabudovaná	Samba	Zabudovaná/Samba	

- *Web adresár* je funkcia správcu súborov *Windows Explorer*.
- *NetDrive*, *WebDrive* sú programy spoločnosti *Novell*, ktoré umožňujú sprístupniť WebDAV a FTP server ako disk (teda transparentne pre všetky aplikácie).
- *Finder* je správca súborov *Apple OS X*.
- *KDE*, *GNOME* sú grafické užívateľské prostredia pre Linux a unixové OS (Sun Solaris, FreeBSD, OpenBSD).
- *Samba* je primárne CIFS server avšak dodáva sa spolu s klientom *smbclient*.
- *OpenOffice* je balík kancelárskych aplikácií. Podporuje tak vstup ako aj výstup obsahu na vzdialené WebDAV serveri.

5 <http://www.alfresco.com/>

6 Program NetDrive má nejasnú licenciu.

3.3.4 Zhrnutie

Tabuľka 23: Podpora súborovo orientovaných protokolov v CMS

CMS	Podpora	Adresár	Súbor	Konverzia
Bricolage	FTP	-	Iba pre šablóny.	Bez konverzie
Plone	Zope FTP/WebDAV - podporuje uzamykanie WebDAV	Folder	Komponent	- Definovaná komponentom, <i>Marshaller</i>
Typo3	Meta FTPd FTP/WebDAV	Komponent (page)	Element	- Definovaná v konfiguračnom súbore.
ezPublish	-	Folder	Len binárne súbory	Bez konverzie
TikiWiki	Experimentálny modul	Structure	Komponent	Automatická pre: Wiki Text OpenDocument DocBook

EzPublish

Konverzia formátov prebieha iba jedným smerom a to pri importe obsahu. Štandardne sú podporované formáty OpenDocument, MSWord, RTF. Pre importovaný súbor je vytvorený komponent a obsah je vložený do elementu. Tak typ komponentu ako aj typ elementu sú globálne nastavenia. Opačným smerom je možné pristupovať len k elementom s binárnym obsahom (obrázky, audio, video ap.). Tieto sú uložené ako komponent typu súbor.

Mapovanie adresárov prebieha na úrovni typov komponentov. Elementy, ktoré predstavujú odkazy na binárne dáta sú následne reprezentované ako súbory.

Príklad: nech komponent A typu článok obsahuje dva obrázky. Ak A nieje nastavený ako adresár je reprezentovaný prázdny súborom, inak je reprezentovaný adresárom s dvoma obrázkovými súbormi. WebDAV rozhranie EzPublish je vhodné pre pridávanie binárnych súborov, alebo import textového obsahu. Export textového obsahu nieje možný, čo považujeme za veľkú nevýhodu.

Plone

Konverziu obsahu komponentu je možné ovplyvniť týmito spôsobmi.

1. V rámci definície komponentu preťažiť metódu `manage_FTPget`.
 - Komponent s HTML obsahom vloží metadáta elementu do `<meta>` HTML tagov v hlavičke posielaného súboru.
2. Súčasťou definície typu komponentu je *marshaller*. Tento objekt je zodpovedný za konverziu do formátu vhodného pre externé aplikácie. Existujú dva základné typy:
 - `PrimaryFieldMarshaller` – element ktorého typ je označený ako *primary* je použitý ako zástupca celého komponentu.
 - `RFC822Marshaller` – výsledkom konverzie je množina dvojíc meno typu elementu a obsah elementu, vo forme kľúč:hodnota. Nakoľko sú dvojice vložené priamo do textu, hrozí ich poškodenie zo strany externej aplikácie.
 - Otvorili sme komponent s obsahom typu štruktúrovaný text v aplikácií

OpenOffice. Pri uložení boli dvojice zahrnuté do textu, keďže aplikácia odstránila nové riadky medzi dvojicami a na začiatok súboru vložila formátovacie elementy.

Nové typy Marshaller objektov je možné dodať ako rozširujúci modul.

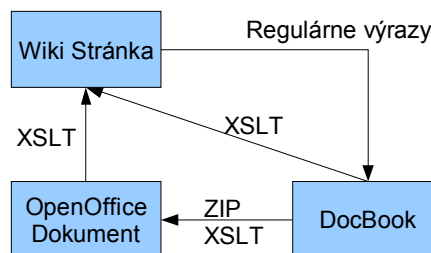
TikiWiki

Implementácia je založená na module *HTTP_WebDAV_Server* [34], použiteľný pod PHP licenciou.

Každý komponent je reprezentovaný niekoľkými virtuálnymi súbormi:

- Wiki text: obsah je prezentovaný v originálnej forme, súbory je teda možné modifikovať či vytvárať, použitím wiki jazyka.
- DocBook. Čiastočná podpora.
- OpenDocument: formát používaný OpenOffice a StarOffice, je podobne ako DocBook založený na XML. Nakoľko však obsahuje mnohé pokročilé funkcie ako vkladanie obrázkov, či tabuliek, je jeho podpora len čiastočná.

Zjednodušenú schému konverzií medzi DocBook, OpenOffice dokumentami a wiki-stránkami prezentuje Obrázok 11.



Obrázok 11: Transformácia obsahu

Typo3

FTP a WebDAV služby sú poskytované modulom Meta FTPd. Sprístupňuje tak binárne súbory (napríklad obrázky, video) ako aj obsah (len z tabuľky TT_content). Konverzia je iba jednosmerná a to pre import obsahu. Pre každý typ (podľa prípony) importovaného súboru, môžeme definovať:

- databázovú tabuľku
- stĺpec v tabuľke pre meno (použije sa meno súboru)
- stĺpec pre dáta (dáta z importovaného súboru po transformácii)
- voliteľne cestu k programu, ktorý sa použije pre transformáciu.

Nový riadok v databáze predstavuje nový element komponentu – adresára, do ktorého sme súbor importovali.

Export je jednoduchší, riadky tabuľky zodpovedajúce elementom komponentu sú prezentované ako súbory s menom <identifikátor>[<názov>].<typ> napríklad: 3[Vitajte].text.

Veľkou výhodou je možnosť použiť externé aplikácie ako aj pomerne jednoduchá konfigurácia. Podobne ako pre TikiWiki aj v tomto prípade ide o experimentálny modul založený na *HTTP_WebDAV_Server*.

3.4 E-mail

E-mail v súčasnosti predstavuje najrozšírenejší druh elektronickej komunikácie. Nie je preto prekvapením, že sa autori CMS snažia využiť toto médium. V tejto časti sa pozrieme na možnosti využitia e-mail ako nástroja pre vstup obsahu.

Princíp použitia je jednoduchý, autor obsahu pošle e-mail na preddefinovanú adresu. CMS následne tento e-mail prijme (či už prostredníctvom POP3, IMAP z e-mail serveru, alebo priamo z lokálneho súborového systému) a spracuje.

Tento druh vstupu obsahu je jednorázový, preto sa najčastejšie využíva pre *blogy*, respektíve pre *diskusné fóra*. (V prípade využitia v kombinácii s diskusným fórom, dochádza k zaujímavému efektu, totiž takéto diskusné fórum je veľmi blízke archívu e-mailového zoznamu tzv. *mailinglist*.)

Tabuľka 24: E-mail ako mechanizmus vstupu obsahu

CMS	Modul	Poznámky
Drupal	mailhandler	Vstup: IMAP, POP3, lokálny adresár. Bezpečnosť: Možnosť overenia hesla v texte e-mailu. Možnosť definovať kategóriu obsahu, okamžité zverejnenie. Využitie pre nový obsah ako aj komentáre (<i>diskusné fórum</i>).
TikiWiki	-	Vstup: POP3 Podporuje pridávanie nových, prepisovanie existujúcich wiki-stránok alebo pripojenie k existujúcim wiki-stránok. Súčasne podporuje posielanie wiki-stránok. Využitie i pre <i>Články</i> a <i>diskusné fóra</i> (konfiguruje sa samostatne). Bezpečnosť: Možnosť porovnania e-mailovej adresy odosielateľa s adresami existujúcich užívateľov.
Wordpress	-	Vstup: POP3 Možnosť definovať kategóriu pre obsah prichádzajúci týmto kanálom. Bezpečnosť: žiadna, každý e-mail
	Postie	Vstup: POP3, IMAP, podporuje zabezpečené spojenie cez SSL Možnosť uviesť kategórie pod ktorými má byť obsahu publikovaný v <i>predmete</i> e-mailu. Filtrácia a podpora pripojených súborov (<i>attachment</i>), automatické premenovanie kvôli zabráneniu kolízií, generovanie zmenšených náhľadov obrázkov. Bezpečnosť: Možnosť porovnania e-mailovej adresy odosielateľa s adresami existujúcich užívateľov.

Bezpečnosť tejto vstupnej metódy je relatívne nízka. Útočník totiž môže adresu odosielateľa e-mailu zvoliť ľubovoľnú.

V prípade Drupal je možné vyžadovať heslo v tele e-mailu, v tomto prípade je však nutné použiť zabezpečený kanál. Použitie SSL smerom od odosielateľa k e-mail serveru je dnes relatívne rozšírené, avšak Drupal nepodporuje bezpečný kanál smerom od e-mail serveru k CMS. Preto odporúčame použiť e-mail server, ktorý je pod našou kontrolou, kde komunikácia medzi CMS a e-mail serverom prebieha v kontrolovanom prostredí lokálnej siete. Súčasne použitie hesla a HTML e-mailu nieje v implementácii programu Drupal možné. Nakoľko e-mailový klienti, do HTML vkladajú špeciálne formátovacie konštrukcie, robustná detekcia hesla by bola zložitá. Možným riešením je vytvorenie HTML šablóny (podpora je relatívne rozšírená medzi e-mailovými klientmi), ktorá by obsahovala element pre vstup hesla (*TextArea*).

3.5 Web-feed import

Web-feed je technológia s komplikovanou históriou, popisuje totiž hneď niekoľko štandardov. Termín *web-feed* sa používa ako spoločné pomenovanie tak RSS (všetkých verzií) ako aj ATOM štandardov.

Počiatok sa spája so skratkou RSS. Technológia *Really simple syndication* vznikla ako metóda pre distribúciu a zdieľanie obsahu medzi webovými stránkami. RSS bol v tomto zmysle názov formátu pre konkrétny druh xml jazyka. Tento formát definoval jednoduchý slovník xml elementov, ktoré slúžili k popisu metadát webových stránok. Ide o RSS 0.9x.

Počiatočná definícia sa čoskoro ukázala ako nedostatočná, neustále sa pridávali nové xml elementy, rôzne inštitúcie pridávali vlastné neštandardné a nekompatibilné. Potreba revízie štandardu bola čoraz zreteľnejšia. Na tomto mieste sa štandard rozdelil.

Časť autorov sa rozhodla použiť existujúci model pre popis zdrojov – RDF a aplikovať ho na *web-feed*. Tak vznikol nový štandard s rovnakou skratkou ale iným menom *RDF Site Summary*. Pomocou RDF boli definované typy zdrojov – prvky, kanály, metadáta a ich vzťahy, pričom niektoré z nich boli prebrané z existujúcich štandardov ako Dublin Core. RDF je však iba model, jedným zo spôsobov jeho zápisu je XML, pričom zápis nieje jednoznačný. Napríklad môžeme použiť atribúty namiesto elementov a naopak. Dôležitým aspektom tohto štandardu je, že zápis RDF do XML je normalizovaný, teda predpisuje tak povolené RDF ako aj XML konštrukcie. Tento štandard je označujeme ako RSS 1.0.

Druhá skupina sa rozhodla pre menej radikálne riešenie a to formalizovaním možnosti rozšírenia RSS pomocou XML namespace. Podobne ako RSS 1.0 používa Dublin Core. Existujú i ďalšie rozdiely oproti staršiemu štandardu ale prakticky ide o relatívne jednoduché rozšírenia. Napriek tomu, že ide o evolúciu štandardu 0.9 nesie meno RSS 2.0.

Najmladším štandardom v rodine *web-feed* je ATOM. Zámerom autorov bolo vytvoriť skutočne otvorený a striktný štandard. Rovnako ide o xml jazyk, využívajúci *XML namespace*, nevyužíva však RDF. Ďalšou výhodou je striktná definícia garantovaná XML schémou (konkrétne RELAX NG).

Diskusia o rozdieloch a výhodách existujúcich *web-feed* štandardov je nad rámec tejto práce. Čitateľov odkazujeme na [15]. V tabuľke 25 prezentujeme stručný náhľad na aktuálny stav *web-feed*.

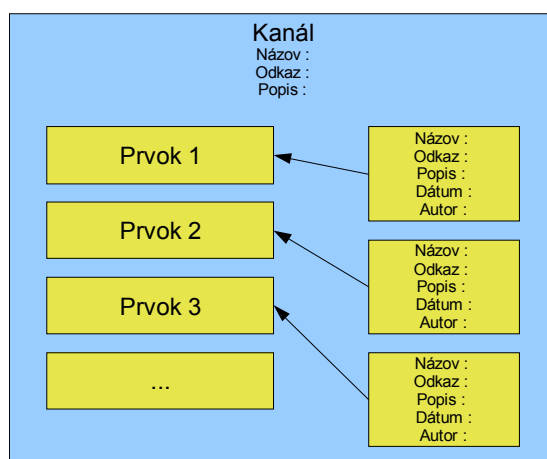
Tabuľka 25: Verzie web-feed štandardov

Verzia web-feed	Autor	Výhody	Odporúčania
0.90	Netscape		Nepoužívať.
0.91	UserLand	Jednoduchosť	V súčasnosti stále rozšírené, dobrá podpora na strane klienta. Len pre jednoduché použitie.
0.92,(0.93)	UserLand	Bohatšie metadáta	Odporúča sa použiť verziu 2.0.
1.0	RSS-DEV Working Group http://web.resource.org/rss/1.0	RDF, rozšíriteľnosť	Pre aplikácie založené na RDF.
2.0	UserLand http://backend.userland.com/rss	Rozšíriteľnosť, kompatibilné s 0.9x	Všeobecne použiteľné.
ATOM 1.0	Otvorený štandard. http://www.ietf.org/rfc/rfc4287	Vyžaduje globálny identifikátor prvkov. Vyžaduje konkrétny formát pre dátum.	Všeobecne použiteľné.

Najčastejšie sa s technológiou web-feed stretávame ako konzumenti, teda ako publikačného mechanizmu. V rámci tejto kapitoly sa budeme zaoberať využitím *web-feed* ako komunikačného média medzi CMS. Poznamenajme, že to bolo práve toto využitie, ktoré stálo pri zrode *web-feed* technológie. Na začiatok si však *web-feed* predstavíme bližšie a uvedieme niekoľko problémov tejto technológie a ich riešení.

3.5.1 Zjednodušená schéma web-feed

Napriek syntaktickým rozdielom všetky štandardy popisujú jednoduchú dvoj-úrovňovú hierarchiu. V tejto kapitole uvádzame zjednodušenú schému, ktorá je základom všetkých druhov *web-feed*.



Obrázok 12: Štruktúra Web-feed súboru

- Koreňom hierarchie je kanál - *feed* (respektíve *channel*).
 - **Názov** – čistý text.
 - **Odkaz** – URL, odkaz na webovú stránku, z ktorej obsah pochádza. Nejde o URL *web-feed*!
 - **Krátky popis** – čistý text alebo HTML.
 - Ďalšie metadáta, závislé od konkrétneho štandardu. Napríklad autor obsahu, administrátor web stránky, kontakt, autorské práva, periodicita obsahu a podobne.
 - Množina prvkov:
 - **Názov** – čistý text.
 - **Odkaz** – URL, na tejto adrese nájde užívateľ obsah, ktorý je reprezentovaný daným prvkom.
 - **Dátum** – existujú dva štandardy: RFC 822 a RFC 3339. V praxi sa však stretáme s veľkým množstvom variácií predchádzajúcich štandardov, pri implementácii odporúčame nahliadnuť do zdrojových kódov existujúcich open-source programov alebo ich priamo využiť. Pre zaujímavosť populárny program pre konzum *web-feed* RSSOwl⁷ podporuje 23 formátov dátumu (a napriek tomu sa autorom podarilo nájsť *web-feed* ktorého dátum nepodporoval). Ďalším problémom je, že často pri implementácii zabúda na časovú zónu – príkladom je *web-feed* stránky populárnych slovenských novín www.sme.sk.
 - **Popis** – čistý text najčastejšie však HTML. Popis môže byť celý obsah komponentu, alebo len prvých niekoľko viet. Znova pri implementácii sa môžeme stretnúť z nevalidným HTML, preto odporúčame podobne ako pri vstupe HTML obsahu filtrovať! Za problém považujeme i použitie iného kódovania textu ako *UTF-8*. Pomerne často sa môžeme stretnúť s entitami z kódovania *Latin*. Štandard ATOM definuje elementy pre krátky úvod ako aj celý obsah. Považujeme to za zbytočné, nakoľko najčastejšie ide iba o duplicitné dáta.
 - **Kategória** – čistý text. Prípadne niekoľko kategórií. Nieje podporovaná všetkými druhmi *web-feed*, ale využitie je populárne.
 - **Príloha** – URL, odkaz na súvisiaci obsah, najčastejšie multimediálny. V prípade, že odkaz je na audio súbor, sa často daný *web-feed* označuje ako **podcast**, (video – **videocast**).
 - Podobne ako pre kanál sú ďalšie metadáta závislé od konkrétneho štandardu.

K *web-feed* najčastejšie pristupujeme pomocou HTTP alebo HTTPS protokolov.

V praxi sa využívajú *web-feed* dvomi spôsobmi:

- Posuvné okno – *web-feed* obsahuje preddefinovaný počet najnovších komponentov.
- Kompletný archív – *web-feed* obsahuje všetky komponenty (napríklad z predvolenej štruktúry).

7 RSSOwl <http://www.rssowl.org>

3.5.2 Problémy *web-feed*

Jedným zo základných problémov je aktualizácia, teda ako často má klient *web-feed* stiahnuť. V súčasnosti tvoria *web-feed* značnú časť všetkých prenesených dát. Existuje niekoľko riešení:

- Na úrovni *web-feed*, použiť elementy daného štandardu.
- Na úrovni http protokolu, využiť hlavičky HTTP. Webové servery často spolu s *web-feed* posielajú metadáta ako dátum modifikácie alebo jedinečný identifikátor (eTag). Pri ďalšej požiadavke pošle klient dátum modifikácie alebo identifikátor serveru, ktorý v prípade, že sa obsah nezmenil, odpovie iba špeciálnou chybovou správou.

Ani jedno riešenie však neumožňuje preniesť iba naozaj potrebné dáta, teda nové komponenty od poslednej aktualizácie. Skutočným riešením je *Delta-Encoding* [50] rozšírenie HTTP. Pracuje na podobnom princípe ako použitie hlavičiek HTTP rozdiel je však v tom, že v prípade, že sa obsah na servery (CMS) zmenil, odpoveď pozostáva z rozdielov. *Delta-Encoding* definuje niekoľko spôsobov ako rozdiely vytvárať, z hľadiska *web-feed* je najzaujímavejším *diff* alebo *feed* riešenie. V tomto prípade je totiž odpoveď syntakticky validný *web-feed*. Problémom tohto riešenia je však zaťaženie serveru, keďže *web-feed* je nutné generovať pri každej požiadavke.

Druhým závažným problémom je ako zistiť, že prvky v dvoch aktualizáciách *web-feed* popisujú rovnaký obsah. Najčastejšie sa ako jedinečný identifikátor používa odkaz prvku, prípadne spolu s hašom popisu. Inou možnosťou je využiť elementy daného *web-feed* štandardu, ktoré boli práve na tento účel určené. Napríklad RSS 2.0 poskytuje element **guid** v ATOM **id**.

3.5.3 Jednorázový import obsahu

V tomto smere je zaujímavý formát OPML, ktorý predstavuje obálku niekoľkých *web-feed* kanálov, zaradených do hierarchickej štruktúry kategórií. Zaujímavou možnosťou by bolo využiť tento formát pre export/import všetkého obsahu CMS. Špecifické potreby CMS by bolo možné zakomponovať pomocou nových *XML namespace*.

3.5.4 Priebežná integrácia obsahu.

1. Jednoduchá integrácia, špeciálny element na webovej stránke prezentuje obsah *web-feed*. CMS kontinuálne v určených intervaloch stiahne *web-feed* súbor. Niektoré CMS dovoľujú užívateľom definovať vlastné kanály.
2. Hlboká integrácia, obsah *web-feed* je integrovaný do CMS. V závislosti od nastavenia, užívateľ nieje schopný rozoznať z akého zdroja obsah pochádza. Takéto využitie dovoľuje synchronizáciu niekoľkých CMS.

Špeciálnym druhom CMS, je takzvaný *Planet*. Vyznačuje sa tým, že vstup obsahu je obmedzený na opakovaný import obsahu z *web-feed* od rôznych autorov. Takýto degenerovaný CMS je použiteľný napríklad v prípade veľkého počtu autorov s rovnakým zameraním, ktorí súčasne využívajú vlastné CMS. Z CMS v našom prehľade je možné takto použiť Drupal.

Tabuľka 26: Existujúce Planet riešenia

Planet	Popis
http://planet.gnome.org	Agregácia nezávislých <i>blogov</i> programátorov Gnome.
http://www.planetkde.org	Agregácia nezávislých <i>blogov</i> programátorov KDE.

Tabuľka 27: Integrácia obsahu z *web-feed*

CMS	Modul	
Drupal	aggregator2	<ul style="list-style-type: none"> - Poskytuje obe možnosti integrácie. - Pre každý kanál vytvorí blokový element*. - Nové správy môžu byť zobrazené na úvodnej stránke. - Prvky môžu byť zaradené do kategorizácie, respektíve je možné využiť informácie o kategórii priamo z prvku <i>web-feed</i> kanálu.
Typo3	xml_ttnews_import	Využíva modul ttnews. Prvky <i>web-feed</i> súboru sú zobrazené v rámci modulu ttnews ako nové správy.
	tw_rssfeeds	Jednoduchá integrácia. Poskytuje vlastné rozhranie.
TikiWiki	Súčasť inštalácie	<ul style="list-style-type: none"> - Je možné vytvoriť, blokový element*, ktorý obsahuje definovaný počet prvkov jedného alebo viacerých <i>web-feed</i>. - Prvky <i>web-feed</i> kanálu je možné vložiť i do wiki-stránky.
Mambo	FeedGator – počiatočná verzia, alfa 0.3	<ul style="list-style-type: none"> - Hlboká integrácia: prvky <i>web-feed</i> sú importované do zvolenej sekcie/kategórie. - Pokročilé funkcie ako filtrácia a oprava obsahu na validné XHTML.
	Súčasť inštalácie – Newsfeed	<ul style="list-style-type: none"> - Vlastné rozhranie pre prezentáciu, zobrazuje definovaný počet prvkov <i>web-feed</i>. <i>Web-feed</i> je možné zaradiť do kategórií (lineárny zoznam, nezávislý od kategórií štandardných komponentov). - Je možné vytvárať blokové elementy* pre <i>web-feed</i>. Táto možnosť funguje nezávisle od predchádzajúcej.
Wordpress	FeedWordPress	<ul style="list-style-type: none"> - Hlboká integrácia: prvky <i>web-feed</i> sú importované ako nové príspevky a je im voliteľne priradená kategória a autor. - Ak prvok obsahuje informácie o kategórii, táto môže byť automaticky vytvorená a použitá. - Ak prvok obsahuje informácie o autorovi, je možné vytvoriť automaticky nového užívateľa. - Podporuje RPC-XML ping, po prijatí správy aktualizuje určený kanál, alebo všetky kanály. (Wordpress ale aj iné CMS, sú schopné takúto správu vygenerovať po pridaní nového obsahu, čo umožňuje distribúciu obsahu, bez potreby opakovaného prístupu k <i>web-feed</i> súboru na vzdialenom serveri.)

*Viac o blokových elementoch v kapitole o prezentácii obsahu.

3.6 Prepojenia rôznych CMS

Špeciálny druh štruktúry tvoria pingback a trackback. V tomto prípade nejde o štruktúru obsahu v rámci konkrétneho CMS, ale o prepojenie medzi rôznymi CMS.

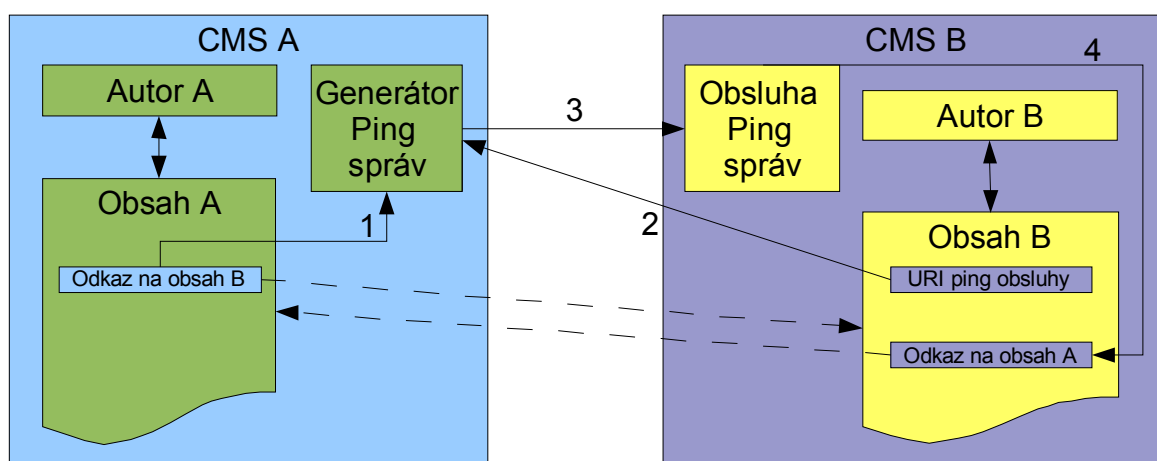
Definícia: Pingback je model pre komunikáciu a notifikáciu medzi webovými stránkami. Základnou myšlienkou je *Ping správa*, v zmysle *Obsah A sa vzťahuje ku, respektíve je prepojený na, Obsah B*. Obsah je v tomto prípade reprezentovaný štandardným URI.

Scenár použitia: autor obsahu A si praje oboznámiť autora obsahu B o tom, že vytvoril niečo relevantné k obsahu B. A pošle *ping správu* B. Výsledkom je, že B môže automaticky zobraziť odkazy na svoj obsah a tým sa vytvorí pevné prepojenie medzi obsahom A a B [18], [19].

Pingback predstavuje rozšírenie a zlepšenie *trackback* modelu. Na jednej strane zjednodušuje mechanizmus detekcie podpory, teda či CMS B podporuje prijímanie *pingback*. Na druhej strane pridáva možnosť kontroly, B môže mechanicky skontrolovať či ide o validný odkaz. V prípade *trackback* musí autor obsahu A zadať URI obsahu B manuálne, *pingback* využíva URI z odkazov priamo v obsahu A. Toto riešenie ma zásadné výhody:

1. CMS B ma možnosť kontroly. Súčasťou *pingback* správy je URI obsahu A i obsahu B, teda CMS B môže skontrolovať či sa odkaz na obsah B v obsahu A skutočne nachádza.
2. Keďže *pingback* využíva URI priamo z odkazov v obsahu A, posielanie *ping správ* je transparentné a nevyžaduje akciu zo strany autora A.
3. CMS B publikuje odkaz na obsah A, vznikne obojsmerné prepojenie.

Využitie prítomnosti odkazov v obsahu je súčasne i nevýhodou Pingback modelu, nakoľko vzniká problém pre iný ako textový druh obsahu.



Obrázok 13: Trackback, Pingback

1. Pingback: CMS A vyberie odkazy z obsahu.
TrackBack: Užívateľ zadá relevantné odkazy manuálne.
2. CMS A nájde URI pre obsluhu *ping správ* v obsahu B (URI musí byť v oboch prípadoch súčasťou obsahu).
3. CMS A pošle *ping správu* CMS B.
4. Pingback: CMS B voliteľne skontroluje či obsah A existuje, respektíve či existuje odkaz na obsah B.
CMS B vytvorí odkaz na obsah A a vloží ho ako súčasť obsahu B (nakoniec), alebo ako príspevok v diskusií.

Táto technológia vyšla z prostredia *blogov*. Z CMS v našom prehľade ju implementuje Wordpress, rozširujúci modul je dostupný pre Drupal.

4 Formy prezentácie obsahu

V tejto kapitole sa pozrieme na metódy prezentácie obsahu. Základnou jednotkou je v tomto prípade publikácia. Bez ujmy na všeobecnosti sa zameriame len na elektronické formy publikácií, nakoľko ostatné formy sú z nich, dnes prakticky bez výnimky, generované automaticky.

Elektronická publikácia má formát, ktorý definuje význam jednotlivých častí publikácie (prakticky bitov). Okrem formátu, je z pohľadu CMS dôležitý vzťah medzi publikáciou a obsahom, teda, ktoré komponenty, prípadne elementy sú súčasťou ktorej publikácie.

Postup pri vytváraní publikácie sa skladá z nasledujúcich krokov.

1. Výber obsahu (obsahov alebo ich častí).
2. Prevod obsahu z vnútornej reprezentácie obsahu do fyzickej reprezentácie daného formátu. V tomto kroku je dôležité zachovanie štruktúry – sémantiky obsahu.
3. Výber výstupného formátu.
4. Doručenie publikácie.

Podľa toho, či sú publikácie generované na požiadanie, alebo sú vygenerované dopredu, rozdelujeme CMS na:

- **online** – Všetky CMS v našom prehľade okrem Bricolage. Samozrejme generovanie stránok je pomerne pomalá a náročná operácia (v závislosti od obsahu), preto aj online CMS často automaticky vygenerovanú stránku uložia do cache.
- **offline** – Bricolage.

4.1 Výber obsahu

Pre výber obsahu poskytujú CMS niekoľko možností:

- Konkrétne komponenty (triviálna možnosť).
- Na základe vnútornej štruktúry obsahu (teda elementov).
- Na základe vonkajšej štruktúry (časť hierarchie komponentov, komponenty v určitej kategórii, komponenty z určitého časového rámca).
- Na základe *logického pohľadu*, ktorý definujeme ako logickú formulu nad predchádzajúcimi možnosťami.

Každá z možností definuje *logickú publikáciu*.

Tabuľka 28: Netriviálne možnosti výberu obsahu

CMS	
Drupal	<ul style="list-style-type: none"> - <i>Views</i> dovoľujú vytvoriť komplexný dotaz nad obsahom v CMS. - Premenné dotazu sú elementy komponentov. - Operácie sú podľa vybraného elementu: <ul style="list-style-type: none"> • textové: je rovný, obsahuje slovo (slová), začína slovom, • číselné: >, <, =, <=, >= • kategórie: je/nieje v kategórii. - Súčasne môžeme zvoliť aj spôsob prezentácie, teda ktoré elementy komponentu publikovať v akom poradí a ako formou (tabuľka, zoznam, ap). - Zaujímavou funkciou je, že niektoré z definovaných kritérií môžeme zverejniť a tým vytvoriť interaktívny nástroj pre filtrovanie obsahu. <i>Views</i> sú identifikované pomocou <i>cesty</i> (časť URL, po doméne). - <i>Diskusia</i> súvisiaca s komponentom.
EzPublish	<ul style="list-style-type: none"> - Poskytuje pomerne komplikovaný nástroj. V rámci obsahu komponentov typu <i>Folder</i>, definujeme odkazy na iné komponenty. Odkazy sú pri publikácii nahradené komponentami, pričom môžeme vybrať z niekoľkých možností prezentácie: nahradenie, vertikálny/horizontálny zoznam (v prípade, že komponent bol tiež typu folder) a podobne.
Mambo	<ul style="list-style-type: none"> - Len podľa príslušnosti do určitej kategórie. - Podobne ako Drupal ale do menšej miery, dovoľuje nastaviť prezentáciu.
Plone	<ul style="list-style-type: none"> - <i>Topic/Smart Folder</i>, funguje prakticky rovnako ako Drupal, poskytuje však prepracovanejší systém pre definíciu operácií. - Operácie sú na rozdiel od Drupal voliteľne textové, číselné, zoznamové (obsah elementu je/nieje v zozname) a dátumové. - <i>Diskusia</i> súvisiaca s komponentom. - Modul <i>CompositePack</i> dovoľuje vytvoriť publikáciu voľnou kompozíciou niekoľkých existujúcich komponentov.
TikiWiki	<ul style="list-style-type: none"> - Chronologický zoznam pre komponenty druhu <i>blog</i>. - Zoznamy objektov v kategóriach. - Zoznam <i>podobných</i> wiki-stránok - <i>Diskusia</i>
Typo3	<ul style="list-style-type: none"> - Chronologický zoznam pre komponenty v module správy (news) - <i>Diskusia</i>
Wordpress	<ul style="list-style-type: none"> - Chronologický zoznam komponentov <i>archív</i> (podľa mesiacov) - Zoznam komponentov podľa kategórie a tagov - <i>Diskusia</i>

4.2 Prevod reprezentácie obsahu

V tejto kapitole sa pozrieme na tok obsahu z CMS do publikácií. Na začiatku máme určitú logickú publikáciu, ktorú chceme previesť do fyzickej publikácie, teda do explicitnej postupnosti bitov. Prevod pozostáva z týchto krokov:

1. Prevod formátu elementov. V prípade komplexných textových elementov je tento prevod najčastejšie na úrovni jazyka, teda z jazyka komplexného typu do jazyka formátu. Multimediálne elementy ako obrázok, audio a video sú konvertované do vhodného rozlíšenia, respektíve vhodného formátu, prípadne sú vložené ochranné známky (napríklad vodotlač alebo DRM).
2. Prevod štruktúry obsahu. V tomto prípade sú štruktúrne informácie:
 1. nahradené dizajnom.
 2. prevedené do formy vhodnej pre daný formát. Napríklad odkazy sú v prípade HTML prevedené do formy webových odkazov. V prípade tlačovo orientovaného formátu sú nahradené číslom strany (ak je obsah, na ktorý sa odkazujeme súčasťou publikácie). Vonkajšia štruktúra komponentov môže byť použitá k automatickému vytvoreniu tabuľky obsahu alebo indexu.
3. Obohatenie publikácie o dizajn.

4.2.1 Šablónové systémy

Prevod je najčastejšie implementovaný ako funkčný modul. Vzhľadom na pomerne veľké množstvo publikácií je nepraktické vytvárať funkčný modul pre každú publikáciu. Riešením je funkčný modul určitým spôsobom parametrizovať, vďaka čomu ho môžeme použiť pre niekoľko publikácií. Takéto funkčné moduly sú v praxi nazývané šablónové systémy, pričom parametre sú označované ako *šablóny*.

1. Šablóny obsahujú kostru výslednej publikácie a poskytujú možnosti ako konkrétne časti kostry nahradiť alebo doplniť o časti logickej publikácie. K dotiahnutiu predchádzajúceho cieľu sa používa programovací jazyk.
2. Šablónové systémy interpretujú šablóny.

Podľa jazyka, ktorý používajú delíme šablónové systémy na systémy s

- úplným programovacím jazykom (PHP, JAVA, python, Perl, C#).
- vlastným, obmedzeným programovacím jazykom.
 - Poskytujú iba niektoré programovacie konštrukcie ako podmienku a cyklus.

Na základe smeru toku informácií rozdeľujeme šablónové systémy na:

- **Pipeline** – CMS iniciuje vytvorenie publikácie, pričom používa šablónový systém.
- **Callback** – šablónový systém využíva programovacie rozhranie CMS k získaniu logickej publikácie a následnému vytvoreniu fyzickej publikácie.

Callback šablónové systémy majú využitie v prípade, že CMS neposkytuje funkcie pre modelovanie logických publikácií. Aj t v tomto prípade je však rozumné oddeliť funkčnú časť pre modelovanie do samostatných šablón a ostatné šablóny z nich použiť *pipeline* spôsobom.

Tabuľka 29: Porovnanie šablónových systémov

	Pipeline	Callback
Výhody	- Bez ohľadu na použitý jazyk implementujú iba prezentačnú logiku. - Umožňuje vytváranie šablón delegovať na užívateľov bez alebo len z obmedzenou znalosťou programovacieho jazyka.	- Jednoduchší a „prírodzenejší“ model.
Nevýhody	- Konceptuálne zložitejší model.	- Spájajú prezentačnú a aplikačnú (definíciu logickej publikácie).

Tabuľka 30: Šablónové systémy

Šablónový Systém	Jazyk	Typ	Poznámky
PHPTemplate	PHP	Pipeline	- Štandardné PHP, rozdiel je v tom, že šablóna nereaguje na HTTP požiadavky ale pracuje len s niekoľkými premennými z CMS. Premenné sú napríklad komponenty alebo elementy obsahu.
TAL	Vlastný jazyk, Python, PHP, Perl	Obe možnosti	- Šablóna je validné XML, nakoľko funkčné časti sú definované v rámci atribútov elementov. Odporúča sa sa použiť vlastný jazyk, avšak dovoľuje i použitie štandardných programovacích jazykov v rámci výrazov (pre podmienky a cykly). - Vlastný jazyk poskytuje definíciu premenných, podmienok a cyklov a zabudované funkcie.
Smarty	Vlastný	Pipeline	- Premenné, podmienky, cykly, zabudované funkcie. Rozšíriteľný pridaním modulov. - Podporuje kompiláciu šablón
TypoScript	Vlastný	Obe možnosti	- Špecifický druh šablónového systému. Založený je na myšlienke deklarácie objektov. Objekty sú preddefinované, napríklad PAGE, MENU, TEXT a podobne. Objekty sa môžu skladať z ďalších objektov, tieto sú špecifikované ako prvky poľa. Navyše majú objekty atribúty (pre text je to value – hodnota). - Poskytuje niekoľko zabudovaných funkcií napríklad pre odstránenie HTML. - Výber obsahu umožňuje objekt CONTENT, ktorý má atribúty pre tabuľku a SQL výraz. Pre takto vybrané elementy je použitý objekt s rovnakým menom (východzie nastavenie). Na tomto objekte potom definuje vzhľad jednotlivých elementov. - Dovoľuje použiť PHP (priamo alebo volaním funkcie). - Dovoľuje použiť jednoduché HTML kostry, v ktorých sú pomocou jednoduchej syntaxe označené sekcie. Sekcie sú potom štandardné TypoScript objekty.
XLST	Vlastný	Pipeline	- Štandard pre transformáciu XML súborov do iných formátov (nielen XML!). Na rozdiel od iných riešení operuje priamo na štruktúre vstupu. - Premenné, podmienky, zabudované funkcie, volania (cykly implementujeme ako rekurzívne volania). - Stavia na ďalších XML štandardoch (podmienky môžu byť v tvare XPath výrazov). - XLST je XML jazyk.
Mason	Perl	Obe možnosti	- Veľa zabudovaných funkcií, objektovo orientovaný model. Nejde o jednoduchý šablónový systém, na Mason sa môžeme pozerať ako na knižnicu určenú pre vytváranie publikácií. Najčastejšie je CMS použitý len ako úložisko obsahu. - Je však možné ho použiť ako pipeline šablónový systém. - Šablóny sú kompilované (systém ich teda interpretuje iba raz).

4.3 Výstupný formát

Niektoré jazyky pre definíciu formátu sme si už predstavili v predchádzajúcej kapitole. V tejto časti sa zameriame hlavne na webové rozhranie z pohľadu výstupného formátu, keďže cieľom našej práce je prehľad webových CMS a súčasne je web najdôležitejším formátom. Na záver uvádzame stručný prehľad ďalších podporovaných formátov.

4.3.1 Webová stránka

Webové rozhranie plní v CMS úlohu interaktívneho prístupového bodu. Medzi najdôležitejšie funkcie patrí navigácia, vstup a výstup obsahu, autentifikácia užívateľa, správa rôznych aspektov CMS (vytváranie užívateľov, vytváranie záloh, konfigurácia ap.). V tejto časti sa zamierame na využitie webového rozhrania ako publikačného formátu.

Webová stránka sa skladá z nasledujúcich častí alebo elementov:

- Obsahové elementy – definované logickou publikáciou.
- Navigačné elementy:
 - menu – zoznam alebo hierarchia odkazov.
 - *breadcrumbs* – cesta k aktuálnej webovej stránke od koreňa webového portálu
 - vyhľadávací formulár.
- Špeciálne elementy: prihlasovací formulár, *pool* – prieskum, reklama, logo a mnohé ďalšie.

Obsahové elementy

Pri vytváraní statických elementov je úlohou CMS konverzia formátu elementov komponentu do HTML. V prípade komplexných textových elementov ide o preklad medzi jazykom elementu a HTML. Multimediálne elementy ako obrázky, audio a video sú konvertované do vhodného rozlíšenia, respektíve vhodného formátu, prípadne sú vložené ochranné známky (napríklad vodotlač alebo DRM). Narozdiel od obrázkov, ktorých podpora je v HTML zahrnutá, je publikácia audio a video obsahu problematická. Najjednoduchším riešením je vložiť jednoduchý odkaz, čoraz častejšie sa však využívajú prehrávače vložené do webového prehliadača. Jednou možnosťou je použiť HTML element `<object>`, avšak toto riešenie je problematické vzhľadom na prenositeľnosť. Populárnym riešením sú prehrávače založené na technológii Adobe Flash.

Navigačné elementy

Ako sme povedali webový portál je vlastne množina webových stránok. Samozrejme jednotlivé webové stránky by nemali zmysel ak by sa k nim užívateľ nemohol dostať. Je teda nutné aby CMS umožnili vytváranie prepojení medzi webovými stránkami. (Je dôležité si uvedomiť, že štruktúra webových stránok, môže ale nemusí súvisieť so štruktúrou samotného obsahu.) Na druhej strane je dnes pomerne časté, že sa užívateľ dostane na ľubovlnú stránku webového portálu, bez toho aby najskôr prešiel istým, definovaným vstupným bodom (napríklad vďaka webovým vyhľadávačom).

Pod navigáciou teda chápeme dve základné úlohy, umožniť užívateľovi nájsť konkrétny obsah a umožniť užívateľovi orientáciu v štruktúre webových stránok portálu.

Pre vytvorenie navigačného elementu máme dve možnosti:

- Na základe vonkajšej štruktúry obsahu. Často je tento element to v prípade CMS, ktoré vyžadujú explicitnú štruktúru obsahu, poskytovaný automaticky.
- Voľne definovaný graf odkazov. Niektoré CMS poskytujú webový nástroj pre definíciu takéhoto elementu.

Definícia: Základným typom navigačného elementu je zoznam alebo hierarchia odkazov. Pre tento typ elementu sa zaužíval názov *menu*.

Definícia: Cestu v grafe v štruktúry obsahu od koreňa až po konkrétnu webovú stránku, nazývame ako *breadcrumb*.

Podľa použitia delíme navigačné elementy na

- **Globálne**, alebo primárne. Používajú sa na rozdelenie stránok podľa určitého zamerania.
- **Lokálne**, alebo sekundárne. Definujú štruktúru v rámci logických množín stránok, rozdelených podľa primárnych elementov.
- **Kontextové**. Ich úlohou je zobrazit' odkazy na podobné obsahy v kontexte obsahu aktuálnej webovej stránky.

Globálne a lokálne typy navigačných elementov sú najčastejšie implementované práve pomocou menu, založenom na statickej štruktúre obsahu.

Pre kontextové elementy je vhodné použiť dynamickú štruktúru obsahu. Napríklad ako zoznam kategórií alebo tagov (*tag cloud*), prípadne ako breadcrumb v rámci taxonómie.

Tabuľka 31: Porovnanie riešení navigácie

CMS	Poskytuje nástroj	Blokový element	
Drupal	Áno	Áno	- Automaticky: zoznam kategórií a <i>breadcrumb</i> podľa kategórií - Manuálne: Primárne menu: jednoduchý zoznam odkazov. Sekundárne menu: Ľubovolný počet hierarchických menu. Viditeľné iba pre konkrétne stránky, filtrované podľa odkazov. (Vlastnosť blokových elementov.)
EzPublish	Nie	Nie	- Automaticky: Menu a <i>breadcrumb</i> podľa explicitnej statickej štruktúry komponentov. Keďže menu je definované po úrovniach hierarchie, rôzne úrovne môžu mať rôznu prezentáciu.
Magnolia	Nie	Nie	- Automaticky: Menu podľa explicitnej statickej štruktúry komponentov.
Mambo	Áno	Áno	- Automaticky: breadcrumb - Manuálne: Ľubovolný počet hierarchických menu. - Viditeľné iba pre konkrétne stránky (stránky sú jednoznačne identifikované prvkami menu). (Vlastnosť blokových elementov.)
MediaWiki	Nie	Nie	- Manuálne: Existuje iba základné menu, ktoré je nutné definovať v konfiguračnom súbore.
MoinMoin	Nie	Nie	- Manuálne: Existuje iba základné menu, ktoré je nutné definovať v konfiguračnom súbore.
Plone	Čiastočne	Nie	- Manuálne: Primárne menu je zoznam odkazov, je nutné vytvoriť v ZMI,

CMS	Poskytuje nástroj	Blokový element	
			prípadne je možné použiť prvú úroveň statickej štruktúry automaticky. - Automaticky: Sekundárne menu a <i>breadcrumb</i> podľa explicitnej statickej štruktúry komponentov.
TikiWiki	Áno	Áno	- Automaticky: jednoduché menu (zoznam) podľa typu komponentov. - Manuálne: Ľubovoľný počet dvoj-úrovňových menu. - Prvky viditeľné len pre konkrétne skupiny, respektíve ak užívateľ má oprávnenie. - Menu je možné zobraziť len pre konkrétne skupiny. (Viditeľnosť celého menu je vlastnosťou blokového elementu.)
Typo3	Čiastočne	Nie	- Automaticky: Menu podľa explicitnej statickej štruktúry komponentov. Keďže menu je definované po úrovniach hierarchie, rôzne úrovne môžu mať rôznu prezentáciu.
WordPress	Čiastočne	Áno	- Manuálne: Existuje jediné konfigurovateľné hierarchické menu, do ktorého je možné zaradiť komponenty typu <i>page – stránka</i> . - Automaticky: Systém kategórií môže byť voliteľne zobrazený ako menu, pričom štruktúra menu kopíruje štruktúru kategórií. - Zoznam kategórií a tagov pre komponenty.

ZMI - Zope Management Interface, Rozhranie správy aplikačného serveru Zope.

Stavba webovej stránky

Webové stránky sú v najčastejšie vytvárané pomocou šablón, vďaka čomu môžeme ich vzhľad meniť podľa našich potrieb. Súčasne však CMS implementujú i možnosť logickej stavby webovej stránky. V tomto prípade šablóny definujú oblasti alebo pásma, následne môžeme elementy stránky priradiť jednotlivým pásmam pomocou grafického nástroja CMS. Možností je veľa, najčastejšie sa však stretáme s týmito:

- Špeciálne pásmo pre obsahový element, niekoľko pásiem pre blokové elementy. Do tejto skupiny patria Drupal, Mambo, TikiWiki, Wordpress.
- Ľubovoľný počet pásiem. Komponent je priamo webovou stránkou a jeho elementy sú pri vytváraní priradené konkrétnym pásmam. Magnolia, Typo3.

Definícia: CMS Drupal, Mambo a TikiWiki, definujú špeciálny druh elementu webovej stránky – *blokovaný element*. Medzi *blokované elementy* patria navigačné a špeciálne elementy.

Poradie elementov v pásmach je v prípade blokovaných elementov kontrolované vlastnosťou *weight* (Drupal), respektíve *module order* (Mambo), *order* (TikiWiki). Blokované elementy sú nazývané *Block* (Drupal) a *module* (Mambo, TikiWiki). Niektoré sú dodávané štandardne, poskytujú ich i rozširujúce moduly. Zaujímavú možnosť poskytuje Drupal, *views* môžu byť totiž aj blokovým elementom.

V tabuľke 32 uvádzame prehľad využitia *pásiem* a blokovaných elementov.

Tabuľka 32: Umiestnenie blokových elementov

CMS	Pásmo	
Drupal	Neobmedzený počet pásiem, definovaný v šablóne.	- Obmedzenie pre vybrané stránky - Obmedzenie pre druhy obsahu
Mambo	Neobmedzený počet pásiem, definovaný v šablóne.	- Obmedzenie podľa aktívneho prvku menu (teda URL). - Obmedzenie podľa práv užívateľa.
TikiWiki	Ľavé, pravé a obsahové pásmo. Blokove elementy je možné umiestniť len v ľavom alebo pravom pásme.	- Obmedzenie pre skupiny. - Užívatelia môžu definovať viditeľnosť.
Wordpress	Pásmo pre blokove elementy (tzv. sidebar) a pásmo pre obsah.	

Vytvorenie finálnej webovej stránky je aplikácia šablóny na logickú stránku (ktorá sa skladá z logickej publikácie, navigačných a špeciálnych elementov). V tabuľke 33 uvádzame použité šablónové systémy ako aj priradenia šablón.

Tabuľka 33: Použitie šablónových systémov pre webové rozhranie

CMS	Šablónový systém	Priradenie
Bricolage	HTML::Mason, HTML::Template PHP::Template	Komponent.
Drupal	PHPTemplate	Obmedzený počet šablón. Je však možné priradiť špecifické šablóny podľa cesty. Samostatné šablóny: - globálna stránka - obsahové elementy: logická publikácia. - blokove elementy Prezentácia (HTML) konfiguračných nástrojov je súčasťou PHP kódu konkrétneho nástroja.
Plone	TAL	- Podľa typu komponentu. - ArcheTypes poskytujú štandardné šablóny je ich však možné preťažiť pri definícii komponentu. - Webová stránka je vytvorená z niekoľkých šablón. Na najvyššej úrovni je hlavná šablóna, ktorá postupne „volá“ šablóny pre jednotlivé elementy stránky.
EzPublish	Rozšírená verzia Smarty.	Podľa typu komponentu.
Mambo/Joomla	Vlastný typ	Jedna šablóna.
Typo3	TypoScript	Priame priradenie komponentom, alebo komponenty dedia šablónu od predchodcu v statickej štruktúre. Navyše je možné pridať rozširujúce šablóny, ktorých definície nahrádzajú zdedené definície. - Šablóny prakticky definujú tak logickú ako aj fyzickú publikáciu.
MediaWiki	-	
Magnolia	JSP	Komponent.
MoinMoin	-	Jedna šablóna
TikiWiki	Smarty	Podľa typu komponentu.
Wordpress	-	Jedna šablóna

Poznámka

Tak MediaWiki ako MoinMoin poskytujú špeciálny šablónový jazyk. Šablóny sú vytvárané ako štandardné wiki stránky. Následne môžeme takto definované šablóny použiť v ľubovolnej inej wiki stránke (teda aj v ďalšej šablóne). Tieto šablóny však nie sú nástrojom na transformáciu do fyzickej publikácie, ich funkciou je rozšírenie wiki jazyka, napríklad o konzistentné formátovanie často sa opakujúcich častí textu wiki stránky.

4.3.2 Ďalšie výstupné formáty

Tabuľka 34: Výstupné formáty

CMS	Formát	Výber obsahu
Bricolage	Veľké množstvo formátov.	- Komponent. Bricolage je úplne formátovo nezávislé CMS.
Drupal	HTML pre tlač (module print), DocBook (beta)	- Komponent - Statickú štruktúru <i>knihu</i> . Definuje vlastný jazyk DXML pre <i>knihy</i> . Vyžaduje PHP 5.
EzPublish	PDF, MS Doc, OpenDocument	- Komponent
Mambo	PDF, HTML pre tlač	- Komponent
MoinMoin	DocBook, RAW*	- Komponent - Umožňuje import/export ľubovolnej množiny wiki-stránok (vo formáte wiki jazyka) ako zip súboru. Vhodné len pre prenos obsahu medzi rôznymi inštaláciami MoinMoin.
Plone	HTML pre tlač	- Komponent
TikiWiki	PDF, RAW*, HTML pre tlač	- Ľubovlnú množinu wiki-stránok. - Komponent

- textový formát ktorý obsahuje text ako pri editovaní (tj. wiki jazyk) .

4.4 Doručenie publikácie

Prakticky všetky CMS okrem Bricolage a Magnolia obsluhujú HTTP požiadavky priamo. Bricolage namiesto toho exportuje obsah na iné HTTP servery, a to pomocou FTP. Magnolia využíva vlastný protokol (obe časti sú dodávané ako súčasť CMS). Výhodou tohoto prístupu je okrem zvýšenej bezpečnosti CMS aj možnosť použiť niekoľko HTTP serverov pre obsluhu užívateľských požiadaviek.

4.5 Internacionalizácia a lokalizácia

Internacionalizácia a lokalizácia sú prostriedkom k adaptácii produktov ako publikácií, hardware, software pre iné národné a kultúrne podmienky. Kým predmetom internacionalizácie je sprístupnenie produktov v prakticky všetkých podmienkach, lokalizácia je zameraná na adaptáciu pre konkrétne podmienky.

Príklad zameraní **internacionalizácie**:

- reprezentácia národných znakov v texte
- spôsob zobrazovania textu – zľava doprava, sprava doľava
- formátovanie dátumu a času
- formátovanie čísiel – oddeľovač desatinných miest
- miery (metrický systém a imperiálny systém)

Príklad zameraní **lokalizácie**:

- preklad textu do národného jazyka (resp. variantov národného jazyka)
- poradie pri triedení textových reťazcov (napríklad v slovenčine je ch považovaný za znak)

Definícia: Konkrétne nastavenia predchádzajúcich atribútov pre dané národné a kultúrne podmienky označujeme ako *lokálny kontext*. (V prostredí .NET ide o *CultureInfo*, v JAVA je to objekt *Locale*). Lokálny kontext je definovaný pomocou kódu pre jazyk (ISO 639) a kódu pre krajinu alebo región (ISO 3166), napríklad en-US, en-GB, en-NZ, en-AU.

Predmetom internacionalizácie a lokalizácie je v CMS na jednej strane samotný obsah a na druhej strane užívateľské rozhranie CMS (napríklad názvy ovládacích prvkov). Toto rozdelenie je dôležité nakoľko v CMS môže byť lokálny kontext pre užívateľské rozhranie a lokálny kontext pre obsah odlišný. Súčasne môžeme chcieť nastaviť atribúty internacionalizácie a lokalizácie samostatne, teda napríklad používať metrické miery a anglický jazyk.

4.5.1 Internacionalizácia

Ako sme povedali v predchádzajúcej kapitole, ide o adaptáciu pre ľubovoľné národné a kultúrne podmienky. Z toho vyplýva, že:

1. CMS musí použiť také metódy pre vstup, uloženie a prezentáciu obsahu, ktoré sú vhodné v každých podmienkach.
2. CMS musí automaticky prispôbiť prezentáciu obsahu zvolenému lokálnemu kontextu.

Znaková sada

Pozrime sa aké je riešenie prvej úlohy. Historicky je s každým jazykom asociovaná *znaková sada* – kódovanie. Ide o jednoduchú tabuľku, ktorá určuje akú numerickú hodnotu v dátovej reprezentácii má konkrétny znak jazyka. Pre dátovú reprezentáciu znakov používame

jeden byte, pre niektoré jazyky sú potrebné multi-byte kódovania. Pokus o prezentáciu textu v nesprávnom kódovaní vedie k nezobrazeniu niektorých znakov, alebo použitiu nesprávnych znakov. Problém kódovania je, že ide o externú informáciu, súčasne mnohé formáty súborov neumožňujú túto informáciu uložiť (Napríklad čisto textový dokument, alebo formát ZIP). Na druhej strane autori často zabúdajú túto informáciu pridať aj keď to formát umožňuje (HTML). Webový prehliadač MS Internet Explorer preto implementuje heuristiku, ktorá na základe štatistiky výskytov znakov v danom jazyku zvolí kódovanie. Riešením tejto situácie je štandard Unicode. Ide o model pre reprezentáciu ľubovoľného jazyka. Unicode definuje niekoľko znakových sád, najznámejšie a najpoužívanejšie sú:

- **UTF-8** – multi-byte kódovanie, výhodou je, že ASCII znaky sú rovnaké, nula nie je reprezentáciou žiadneho znaku, všetky multi-byte znaky sú reprezentované jedným až štyrmi bytami. Štandardne sa používa v XML.
- **UTF-16** – kódovanie, ktoré reprezentuje každý znak dvomi bytami. Výhodou je rýchly prístup k znakom, nakoľko na rozdiel od UTF-8 nemusíme dekódovať predchádzajúce znaky. Na druhej strane je tento formát nekompatibilný s predchádzajúcimi a vyžaduje rozsiahle modifikácie existujúceho software. Používa sa vo MS Windows NT.

Riešenie je teda použiť unicode pri:

- Vstupe, výstupe – používať unicode kompatibilné formáty.
- Uložení – používať unicode kompatibilné databázy, tabuľky, alebo iné dátové sklady.
- Spracovaní – používať správne programovacie prostriedky (v PHP ide o mb_string rozšírenie).

Formátovanie číselných hodnôt

Prispôbenie prezentácie obsahu lokálneho kontextu, je komplikovanejšie. Z praktického hľadiska je nutné identifikovať časti obsahu, ktoré sú lokálne závislé. Druhým krokom je použitie takých funkcií pre prezentáciu obsahu, ktoré berú ohľad na lokálny kontext. Napríklad pri prevode dátumu z číselnej reprezentácie do textovej sú pozícia dňa, mesiaca, roka, ako aj znak oddeľovača, alebo meno dňa (v týždni) či mesiaca závislé od lokálneho kontextu.

Najčastejšie sa môžeme stretnúť s týmito riešeniami:

- Lokálny kontext je odvodený od jazyka, ktorý sa používa.
- Formátovanie číselných hodnôt je nezávislé od lokálneho kontextu.

RTL

Špecifickou požiadavkou je v rámci lokálneho kontextu zobrazovanie sprava doľava, označované ako RTL. Dôležité je, že nejde iba o zarovnanie, RTL jazyky vyžadujú vertikálne zrkadlové zobrazenie aj logických prvkov vid' príklad 12. Možnosti sú dve:

- Špecifikovať RTL v šablóne publikácií. Vyžaduje rôzne šablóny pre jazyk, alebo komplexné šablóny s integrovanou logikou, súčasne to nerieši problém RTL v užívateľskom rozhraní.
- Podpora RTL priamo v CMS.

Výpis 12: Príklad RTL

Zľava doprava	Sprava doľava – RTL
Krok 1. Krok 2. Krok 3.	Krok 3. Krok 2. Krok 1.
Hlavná stránka > Fakulta FMFI > Katedra X	Katedra X < Fakulta FMFI < Hlavná stránka

Tabuľka 35: Internacionalizácia

CMS	Znaková sada (podpora Unicode)	RTL	Formátovanie číselných hodnôt
Bricolage	Áno (Poskytuje možnosť zvoliť ľubovoľnú znakovú sadu - kódovanie)	Nie	Čiastočne. - Časový a dátumový formát sa nastavuje nezávisle od jazyka.
Drupal	Áno	Čiastočne	Nie (podpora je plánovaná pre verziu 6)
EzPublish	Áno	Nie	Áno, podľa jazyka. (Dátum, mena, oddeľovač desatinných miest, názvy dní a mesiacov.)
Plone	Áno	Áno, podľa jazyka.	Áno, podľa jazyka.
Mambo	Čiastočne	Nie	Nie
Magnolia	Áno	Áno	
MoiMoin	Áno		
Typo3	Áno	Nie	Nie
MediaWiki	Áno		Globálne nastaviteľný formát dátumu.
TikiWiki	Áno	Čiastočne (nieje úplne funkčné)	Nie
Wordpress	Áno	Nie	Nie

4.5.2 Lokalizácia webového rozhrania CMS

V tejto kapitole sumarizujeme jazyky poskytované jednotlivými CMS. Tam kde to bolo možné sme sa pokúsili odhadnúť počet reálne použiteľných prekladov. Napriek tomu však toto porovnanie nieje objektívne nakoľko rozsah potrebného prekladu je rôzny pre rôzne použitia. Súčasne sme v rámci tohto prehľadu uvažovali iba oficiálne, respektíve oficiálne podporované preklady.

Praktický štandard pre preklad v open-source software je formát PO (MO je jeho skompilovaná forma) [56]. Výhody použitia tohto formátu sú:

- Mnohé vyzreté nástroje pre vytváranie a modifikáciu.
- Umožňuje použiť existujúce preklady.
- Dovoľuje definovať preklady pre plurál (a „plural-y“ v jazykoch kde je to potrebné) ako i ďalšie gramatické formy.

Tabuľka 36: Lokalizácia CMS

CMS	Počet podporovaných jazykov (viac ako 60% textov)	Formát	
Bricolage	16(7)	Vlastný, Perl	
Drupal	50(35)	PO	http://drupal.org/project/Translations
EzPublish	30 (závisí od verzie)	Vlastný	http://ez.no/download/translations
Plone	55(40)	PO	http://plone.org/development/teams/i18n/existing-translations
Magnolia	25(?)	Vlastný	http://www.magnolia.info/wiki/Wiki.jsp?page=I18nTranslatorInformation
Mambo	38(?)	Vlastný	http://extensions.joomla.org/component/option,com_mtree/task,listcats/cat_id,1837/Itemid,35/
Typo3	45(18)	Vlastný xml	Pridanie jazyka je zabudované do CMS.
MediaWiki	148(32)	Vlastný PHP	Dodávajú sa všetky ako súčasť inštalácie.
MoinMoin	25(6)	PO (štandardné wiki stránky)	http://moinmoin.wikiwikiweb.de/SystemPages/Setup
TikiWiki	30(23)	Vlastný, PHP	http://tikiwiki.org/TikiWikiInternationalizationAndLanguages
Wordpress	53	MO	http://codex.wordpress.org/WordPress_in_Your_Language

4.5.3 Viacjazyčný obsah

Vytváranie viacjazyčného obsahu

Pre podporu vytvárania viacjazyčného obsahu máme niekoľko možností.

1. Použiť viac inštancií CMS.

Nevýhody:

- Náročné na prostriedky, z technické ako aj ľudského hľadiska.
- Duplikácia práce.
- Potenciálne vedie k nekonzistentnému obsahu.

2. Použiť oddelené statické štruktúry v CMS.

Nevýhody:

- Podobne ako 1, výhodou je, že všetok obsah je v jedinom CMS, čo zjednodušuje správu (napríklad správa užívateľov, procesov).

3. Preklad na úrovni komponentov. V tomto prípade existujú dve možnosti:

1. CMS podporuje viacjazyčné komponenty. Jednotlivé jazykové mutácie nieje možné spravovať nezávisle: spoločná história, spoločná kontrola prístupu. Na druhej strane sa zjednodušuje prepojenie medzi komponentami.
2. Každá jazyková mutácia je samostatný komponent. CMS musí zabezpečiť použitie správnej mutácie (pre danú publikáciu a pre odkazy medzi komponentami).

Tabuľka 37: Vytváranie viacjazyčného obsahu

CMS	Typ	Podpora	Poznámky
Bricolage	2. 3.1.	Zabudovaná	- Definovať špeciálne typy elementov pre každý jazyk. Následne definovať komponent ako štruktúru týchto elementov.
Drupal	3.2.	Modul i18n	- Komponenty, ktoré predstavujú rôzne jazykové mutácie sú rovnakého typu (tj. rovnaké elementy).
Plone	3.2.	Modul LinguaPlone	- Rovnako ako pre Drupal, jazykové mutácie zdieľajú typ.
EzPublish	3.1.	Zabudovaná	- Dovoľuje zobrazit' predlohu v inom jazyku pri prekladaní. - Elementy komponentu majú atribút či môžu byť prekladané.
Magnolia	2, 3.1	Áno	- Preklad na úrovni statickej štruktúry. Samostatná hierarchia komponentov (stránok) pre jazyk. - Preklad je možný aj na úrovni elementov, štandardne dodávané komponenty túto funkciu neposkytujú.
Mambo/Joomla	3.1.	Joom!Fish	- Dovoľuje zobrazit' predlohu v inom jazyku pri prekladaní. - Samostatné rozhranie pre preklad.
MediaWiki	1.	Jazykové mutácie vyžadujú samostatné inštalácie a použitie interwiki odkazov. Viac informácií nájde čitateľ v [52].	
MoinMoin	3.1	Áno	- Komponent môže obsahovať súčasne obsah v rôznych jazykoch.
TikiWiki	3.2	Zabudovaná	- Komponentu je priradený jazyk. - Komponent je možné označiť ako preklad iného komponentu. (Komponenty druhu wiki stránka)
Typo3	3.1.	Zabudovaná	- Elementy jazykových mutácií komponentu môžu byť odlišné, avšak dovoľuje aj preklad elementov podľa predlohy. Teda vyberieme element existujúceho komponentu a zvolíme jazyk, do ktorého chceme prekladať, následne systém vygeneruje predlohu.
Wordpress	3.	Niekoľko modulov, beta verzie.	Závisí od modulu

Publikácia viacjazyčného obsahu

Túto funkciu je podobne možné implementovať niekoľkými spôsobmi a na niekoľkých úrovniach. Obmedzíme sa na webového rozhranie.

1. Manuálny výber jazyka: CMS poskytuje element webového rozhrania, ktorý dovoľuje užívateľom výber jazyka.
2. Automatický výber jazyka: detekcia podľa IP adresy užívateľa, alebo jazyka webového prehliadača.
3. Podporu je nutné doplniť (rozširujúcim modulom alebo v šablóne).

Podľa úrovne:

1. Globálne nastavenie. Celá webová stránka a všetky ďalšie.
2. Na úrovni komponentov.

Okrem predchádzajúcich funkcií musia CMS riešiť problém **nedostupnosti** určitej jazykovej mutácie.

1. Použije sa obsah v špeciálnom predvolenom jazyku.
 1. Globálny predvolený jazyk.
 2. Špeciálne nastavenie pre každý jazyk (napríklad čeština na slovenčinu a naopak).
2. Chybový oznam (napríklad: „Obsah nieje vo zvolenom jazyku dostupný!“).
3. Obsah nieje zobrazený.


CMS	Výber jazyka	Úroveň	Nedostupný obsah
Bricolage	Nie		
Drupal	Obe možnosti modul i18n	Obe možnosti	- Obsah nieje zobrazený - Obsah vo zvolenom jazyku a súčasne v preddefinovanom (jednoduchšia implementácia). - Predvolený jazyk.
EzPublish	Obe možnosti	Globálne	- Predvolený jazyk.
Magnolia	Nie		
Mambo/Joomla	Obe možnosti	Joom!Fish	- Predvolený jazyk a/alebo chybový oznam.
MediaWiki	Nie	Jazykové mutácie vyžadujú samostatné inštalácie a použitie interwiki odkazov. Viac informácií nájde čitateľ v [52].	
MoinMoin	Automaticky	Globálne	- Predvolený jazyk.
Plone	Obe možnosti modul LinguaPlone	Globálne	- Predvolený jazyk a/alebo chybová správa
Typo3	Manuálne: Modul sr_language_menu	Globálne	
TikiWiki	Obe možnosti	Obe možnosti	- Predvolený jazyk.
Wordpress	Obe možnosti	Závisí od modulu	Závisí od modulu

4.6 Web-feed export

Technológiu *web-feed* sme predstavili v kapitole o vstupe obsahu. V tejto časti sa pozrieme ako sa používa pre výstup obsahu. *Web-feed*, ktoré sú generované CMS môžu byť použité ako vstup pre iné CMS, alebo môžu byť agregované samostatnou aplikáciou. Zaujímavé výsledky vzhľadom k použitiu *web-feed* poskytuje štúdia [21]. Najdôležitejšie výsledky zhrnieme v niekoľkých bodoch.

- Relatívne malá časť užívateľov (12%) internetu pozná technológiu *web-feed* (v tomto prípade pojem RSS), respektíve ju vedome využíva (4%).
- Až 27% užívateľov internetu, konzumujú obsah vďaka *web-feed*, bez toho aby si boli toho vedomí.
- Najpopulárnejšou metódou pre prístup k obsahu sprostredkovaným *web-feed* sú špecializované webové stránky ako MyYahoo, MyMSN. Druhou najrozšírenejšou metódou je integrácia s webovým prehliadačom (Firefox *livebookmarks*).
- 50% užívateľov si vyberá *web-feed* z ponuky nástroja, ktorý používa pre konzum. 37% hľadá *web-feed* samostatne, 35% resp. 21% využíva špeciálne tlačítko na webovej stránke (oranžové tlačítko *RSS*, *XML*, respektíve *Add To My Yahoo*).

Web-feed	Podcast	Špecializované webové portály
  		  

 Tento obrázok má najväčšiu podporu prehliadačov a webových stránok.

Z tejto štatistiky plynú dva dôležité závery pre tvorcov CMS. Umožniť programom nájsť a klasifikovať poskytované *web-feed*-y. Vďaka tomu môžu byť tieto *web-feed*-y poskytované tretími stranami (Google, MSN, Yahoo). Umožniť konkrétnemu užívateľovi jednoduchú identifikáciu obsahu, ktorý je dostupný vo forme *web-feed*-u.

Relatívne jednoduché riešenie je založené na využití štandardných prostriedkov HTML, konkrétne *link* elementu.

Výpis 13: *Autodiscovery*

```
1. <html>
2. <head>
3. ...
4. <link rel="alternate" type="application/rss+xml" title="Titul" href="webfeed.xml" />
5. ...
6. </head>
7. ...
8. </html>
```

Atribút *type* môže byť *application/xml*, alebo *application/rss+xml* pre RSS a *application/atom+xml* pre ATOM.

Atribút *href* (riadok 4. výpisu 13) definuje URL (relatívne k URL webovej stránky) *web-feed*-u. Vďaka tomu môžu vyhľadávacie programy jednoducho identifikovať *web-feed*, súčasne toto riešenie umožňuje webovým prehliadačom upozorniť užívateľa na jeho existenciu, prípadne poskytnúť relevantné akcie (Firefox – pridanie do *livebookmarks*, Podobnú funkciu poskytuje Opera, Konqueror – vloženie *web-feed*-u do špecializovanej aplikácie *Akregator*). Táto technika sa nazýva automatické objavenie (*autodiscovery*) a je podporovaná prakticky všetkými webovými prehliadačmi pre PC platformy.

Tabuľka 38: Porovnanie podpory *web-feed* ako výstupného mechanizmu

CMS	Formát	Úroveň podpory	A	B	C	D
Drupal	RSS 2.0	- Dodáva modul <i>web-feed</i> pre <i>views</i> , teda rovnaké logické publikácie ako pre webové stránky. - <i>Diskusia</i> - Súčasťou <i>web-feed-u</i> sú kategórie obsahu. - Modul pre administrátorov <i>AdminRSS</i> . Poskytuje zoznam nových obsahov, ktoré čakajú na publikovanie.	X	X	X	X
EzPublish	RSS 1.0 RSS 2.0	- <i>Web-feed</i> je definovaný ako množina dvojice (komponent, typ komponentu). Voliteľne môžu byť použitý i nasledovníci komponentu (podľa statickej štruktúry). - Pre každú dvojicu dovoľuje vybrať elementy, ktoré budú vo <i>web-feed-e</i> použité (ako meno a popis) - Ľubovoľný počet <i>web-feed</i> .	X	X		X
Mambo	RSS 0.91, 1.0, 2.0 ATOM 0.3	- Podľa sekcie (globálne nastavenie, nie je možné vybrať konkrétne stránky, ktoré budú <i>web-feed</i> poskytovať) - Modul <i>rssxt</i> .	X	X		X
MoinMoin	RSS 1.0	- Posledné zmenené wiki-stránky. Konfigurovateľný počet (max 100). prvky sa môžu odkazovať na stránku rozdielov, alebo môžu zmeny priamo obsahovať (v <i>description</i> elemente prvku).	X			X
Plone	RSS 1.0 (RSS 2.0, ATOM)	- Voliteľne pre každý komponentu typu adresár. Osahom <i>web-feed</i> sú komponenty v adresári (štandardne iba meno a úvod, je však možné konfigurovať zmenou šablóny) - Táto funkcia nieje pri inštalácii automaticky zapnutá. - Modul <i>qRSS2Syndication</i> dodáva veľa dôležitých funkcií, napríklad <i>web-feed</i> pre iné typy komponentov, <i>web-feed</i> pre podstrom statickej štruktúry a podcast podporu.	X	X	X	
TikiWiki	RSS 0.91, 1.0, 2.0 ATOM 0.3 OPML	- Konfigurovateľne pre každý typ komponentu: wiki-stránky, blog, fórum, galéria (okrem wiki-stránok tak globálne ako individuálne, napríklad <i>web-feed</i> pre všetky blogy, ako i konkrétny blog). - Nepodporuje <i>web-feed</i> pre kategórie.	X	X		X
Typo3	RSS 1.0	- Najaktuálnejší obsah v rámci celého CMS. - Relatívne zložitá konfigurácia. - Modul <i>rss_content</i>	X			
WordPress	RSS 2.0	- Pre najaktuálnejšie komponenty globálne a pre každú kategóriu. - Podcast je podporovaný modulom <i>podpress</i> .	X	X	X	X

A) Súčasťou *web-feed-u* je obsah.

B) Prvky *web-feed-u* obsahujú len časť obsahu. Konfigurovateľný počet slov alebo viet. Prípadne je možné využiť explicitné rozdelenie obsahu.

C) Podporuje *podcast*.

D) Podporuje *autodiscovery*.

Okrem textového výstupu je možné *web-feed* použiť i pre multimediálny obsah. *web-feed*, ktorý obsahuje URL odkaz na audio súbor sa nazýva *podcast*. Podobne ako v prípade textových *web-feed-ov*, existujú špecializované aplikácie, ktoré umožňujú jednoduchým spôsobom tento obsah konzumovať. Medzi najrozšírenejšie patrí program spoločnosti Apple – iTunes. (Termín *podcast* pochádza od produktu iPod, ku ktorému je program iTunes štandardne dodávaný.)

Pre wiki je *web-feed* používaný ako metóda pre notifikáciu užívateľa o zmene. Samotná zmena má však hneď niekoľko atribútov ako napríklad druh (aktualizácia, vymazanie).

Tieto dva príklady poukazujú na rozšíriteľnosť *web-feed*-ov. Riešenie je v prípade RSS 1.0, 2.0 ako i ATOM založené na využití *XML namespace*.

Tabuľka 39: Príklady použitia *XML namespace* vo *web-feed*

Druh	Príklad elementov	XML Namespace
Všeobecné metadáta [22]	creator, subject	http://purl.org/dc/elements/1.1/
Wiki	version, status, diff, history	http://purl.org/rss/1.0/modules/wiki/
Media	Group, content, rating, description, thumbnail	http://search.yahoo.com/mrss
iTunes	Keywords, author, copyright, owner, ttl, category	http://www.itunes.com/dtds/podcast-1.0.dtd

Analýza *XML Namespace* presahuje rámec tejto práce, preto sa obmedzíme na popis využitia vo *web-feed*. *XML Namespace* predstavujú slovník povolených elementov a atribútov XML súboru. XML súbor *web-feed*-u obsahuje elementy predvoleného *XML namespace*, môže však obsahovať i elementy dodatočných. Niektoré *XML namespace* (predvolený ako aj niektoré dodatočné) sú súčasťou definície konkrétneho *web-feed* štandardu. Priradenie konkrétnych elementov, konkrétnym *XML Namespace* je zabezpečené využitím prefixov. (Napríklad konvencia pre *Dublic Core* elementy je dc, teda elementy XML budú najčastejšie v tvare dc:creator, dc:subject vid' Tabuľka 39.) Vďaka tejto vlastnosti je zabezpečená kompatibilita, keďže klienti môžu jednoducho ignorovať elementy tých *XML Namespace*, ktoré nepoznajú. Súčasne poskytujú *XML Namespace* prakticky neobmedzené možnosti rozšírenia.

Princíp funkcie *web-feed*-ov v prípade CMS – CMS sme už popísali. Pozrime sa teraz akým spôsobom prebieha konzum v prípade špecializovanej aplikácie.

- Nájdenie.** Užívateľ nájde *web-feed*, ktorý ho zaujíma a to použitím vyhľadávacieho programu (napríklad <http://www.feedster.com> , <http://www.syndic8.com>), alebo použije URL ktoré nájde na webovej stránke CMS.
- Prenos.** URL *web-feed*-u je prenesené do špecializovanej aplikácie (často označovaná ako *web-feed* agregátor). Znovu je možné použiť automatizované riešenie. Webový prehliadač využije *autodiscovery* a registračnú databázu aplikácií (zjednodušene si môžeme túto databázu predstaviť ako zoznam dvojíc *mime-type*, aplikácia) hostiteľského operačného systému na prenesenie URL. Druhé riešenie, užívateľ musí preniesť URL manuálne (i v tomto prípade je možné použiť *autodiscovery*, teraz na strane špecializovanej aplikácie, ak užívateľ preniesie iba URL webovej stránky a nie *web-feed*-u).
- Použitie.** Špecializovaná aplikácia v určených intervaloch, konfigurovateľných užívateľom, alebo definovaných *web-feed*-om (táto možnosť nieje v súčasnosti rozšírená) stiahne xml súbor asociovaný s *web-feed*-om. Následne skontroluje aktuálnosť jednotlivých prvkov. V závislosti na konkrétnej aplikácii môže byť obsah prvkov automaticky stiahnutý, nové a nečítané (respektíve inak nekonzumované) prvky sú zvýraznené.
- Konzum.** Konzum obsahu je definovaný konkrétnou aplikáciou. V prípade textového obsahu je časté využitie webového prehliadača, v prípade *podcast* je možnou akciou prenos obsahu na prenositeľný prehrávač.

5 Porovnanie správy rolí a procesov

5.1 Bezpečnosť

Pojem bezpečnosti môžeme rozdeliť do troch skupín, často označované aj ako AAA. Kontrola prístupu (access control), autentifikácia, audit.

5.1.1 Kontrola prístupu

Na začiatok sa pozrieme na existujúce modely kontroly prístupu. V ďalšej časti podáme prehľad o implementácií kontroly prístupu v jednotlivých CMS, ich výhodách a nedostatkoch.

Definícia: Pod *zdrojom* rozumieme akýkoľvek objekt, ku ktorému sa vzťahuje kontrola prístupu.

Definícia: Pre každý zdroj definujeme množinu akcií – *operácií*.

Definícia: *Oprávnenie* je dvojica zdroj, operácia.

Definícia: *Užívateľ* je osoba, alebo automatický agent (proces), ktorý vykonáva operácie na zdrojoch.

Definícia: *Skupina* je množina užívateľov. Skupiny môžu vytvárať orientovaný graf, pričom skupina preberá užívateľov predchodcov.

Definícia: *Rola* je množina oprávnení. Podobne môžu i roly vytvárať orientovaný graf, množina oprávnení konkrétnej roly je zjednotenie oprávnení predchodcov.

Spôsob akým pre daného užívateľa definujeme oprávnenia, nazývame modelom kontroly prístupu. Existuje niekoľko zaužívaných modelov, ktoré si v krátkosti predstavíme.

DAC (Discretionary Access Control)

V tomto modeli je pre každý zdroj rozhodnutie o povolení alebo zamietnutí operácie založené na identite užívateľa. S každým zdrojom asociujeme zoznam dvojíc (užívateľ, operácia). Rozšírením je možnosť použiť namiesto užívateľa skupiny, teda dvojice (skupina, operácia). Užívateľ pritom môže patriť do niekoľkých skupín, a súčasne skupiny môžu vytvárať hierarchiu.

Príkladom DAC je UNIX, kde je tento zoznam obmedzený na dvojice (vlastník, operácia), (skupina, operácia) a odvodená skupina ostatný (ostatný, operácia). Ide o takzvaný *vlastnícky* DAC, nakoľko o zozname rozhoduje vlastník zdroja.

Všeobecným modelom je ACL (Access Control List), ktorý nekladie obmedzenia na druh a počet dvojíc.

RBAC (Role Based Access Control).

Ako už vyplýva z názvu, je tento model založený na rolách. Každý užívateľ je asociovaný s niekoľkými rolami, podobne roly sú asociované s niekoľkými užívateľmi. Roly najčastejšie tvoria hierarchiu. Základným rozdielom oproti skupinovým ACL, je možnosť definovať obmedzenia. Obmedzenia sú statické alebo dynamické. Pomocou obmedzení môžeme zúžiť množinu rolí, v ktorých užívateľ vystupuje (napríklad užívateľ nemôže súčasne vystupovať v roli autora a editora).

Bez obmedzení sú jednoduché RBAC a skupinové hierarchické ACL prakticky ekvivalentné v schopnosti definovať kontrolu prístupu [53].

MAC (Mandatory Access Control)

Je model, v ktorom sú všetkým zdrojom priradené označenia (label), ktoré definujú senzitivnosť (teda potrebný stupeň ochrany) zdroja. Súčasne sú užívateľom podobne priradené stupne. Rozhodnutie o povolení určitej operácie je založené na porovnaní stupňa užívateľa a zdroja. Stupeň užívateľa môže byť modifikovaný predchádzajúcimi akciami. Práve posledná vlastnosť je hlavným rozdielom oproti DAC, kde oprávnený užívateľ môže zmeniť stupeň ochrany zdroja (kopírovaním, zmenou ACL), kým v prípade MAC ak užívateľ čítal vysoko senzitivný dokument, systém mu zabráni vytvoriť dokument s nižším stupňom ochrany.

Zdroje v CMS

V rámci CMS je objektom komponent alebo element obsahu, ale aj šablóna a podobne. Súčasne je z praktických dôvodov často ako zdroj uvažovaná aj množina komponentov (ktorú môžeme odvodiť napríklad od štruktúry). Oprávnenie pre množinu je následne automaticky priradené jej prvkom.

Tabuľka 40: Zdroj

CMS	Element	Komponent	Štruktúra
Bricolage	Áno	Áno	Áno, pomerne komplikovaný systém, dovoľuje vytvárať množiny objektov.
Drupal	Áno	Áno	Áno, kategórie (modul taxonomy_access)
Plone	Áno	Áno	Áno, statická štruktúra
EzPublish	Nie	Áno	Áno, statická štruktúra
Mambo/Joomla	Nie	Áno	Čiastočne
Typo3	Áno	Áno	Áno, statická štruktúra
MediaWiki	Nie	Áno	Áno, komplexné riešenie, na základe modulu namespace
Magnolia	Nie	Áno	Áno, statická štruktúra
MoinMoin	Nie	Áno	Nie
TikiWiki	Nie	Áno	Áno, kategórie
Wordpress	Nie	Nie	Nie

Rozdelenie užívateľov

Niektoré druhy CMS, definujú dve nezávislé skupiny užívateľov. Prvá skupina sú užívatelia, ktorí sa podieľajú na vytváraní obsahu, kým druhá skupina sú užívatelia, pre ktorých je obsah určený. Tieto skupiny sú najčastejšie označované ako producenti (backend) a konzumenti (frontend).

Tabuľka 41: Rozdelenie užívateľov CMS

Jediná skupina užívateľov	Producenti/Konzumenti	Iba producenti
Drupal	EzPublish	Bricolage
Plone	Magnolia	Wordpress
MediaWiki	Mambo	
MoinMoin	Typo3	
TikiWiki		

Implementácia kontroly prístupu

V tejto časti sa pozrieme na konkrétne implementácie kontroly prístupu. Niektoré všeobecné atribúty uvádzame v tabuľke 42. Následne pre niektoré systémy s komplikovanou implementáciou podávame širší popis a sumarizujeme výhody a nevýhody daného prístupu.

Bricolage

Kontrola prístupu je založená na množinách zdrojov nazývaných skupiny. Skupiny sa rozdeľujú podľa typu zdroju, ktorí obsahujú. Na začiatku máme všeobecné skupiny pre každý typ zdroju (napríklad všetky typy elementov, všetky komponenty, všetky kategórie). Pre komponenty definujeme skupinu jednoduchým priradením. Novú skupinu vytvoríme, tak že zvolíme typ zdrojov (napríklad textové komponenty – *story*) a následne vyberieme zdroje. Na skupinách existuje tieto základné operácie: čítať, modifikovať, vytvárať, publikovať a stiahnuť (prakticky opak publikácie). Oprávnenie priradíme skupinám užívateľov (áno ide prakticky len o špeciálny typ skupiny).

Výhody:

- Vysoká granularita oprávnení.
- Pochopiteľný systém skupín – všetky zdroje (nielen obsah) sú spravované rovnakým spôsobom.
- Založené na logických operáciách.

Nevýhody:

- Skupiny musia byť explicitné zoznamy zdrojov.

Ezpublish

Poskytuje systém s vysokou granularitou a prepracovaným užívateľským rozhraním. Rolám priradujeme oprávnenia – kritéria explicitne. Vyberieme modul a funkciu (prípadne funkciu ešte ďalej obmedzíme na typy zdrojov ap.) a potom zdroj respektíve časť hierarchie zdrojov. Z praktického hľadiska ide o jednoduché RBAC.

Výhody:

- Jednoduché ale silné užívateľské rozhranie.
- Logický systém kritérií.
- Založené na logických operáciach.
- Všetky zdroje sú objekty v ezPublish a sú spravované rovnakým spôsobom.

Nevýhody:

- Roly sú nezávislé a tvoria jednoduchú množinu.

Plone

Poskytuje prepracovaný systém kontroly prístupu. Každý typ zdroja (a špeciálne i každý typ komponentu) uverejní dostupné oprávnenia. Oprávnenia následne priradujeme rolám, a to buď explicitne na danom zdroji alebo je oprávnenie zdedené z predchodcu zdroja. Navyše, pre každý zdroj môžeme definovať takzvané lokálne roly. Lokálna rola pozostáva z dvojice užívateľ alebo skupina a existujúca rola. Lokálnu rolu môžu zdediť nasledovníci zdroja. Oprávnenia pre užívateľa a daný zdroj sú následne zjednotením oprávnení z rolí a lokálnych rolí.

Výhody:

- Vysoká granularita
- Typy komponentov môžu definovať oprávnenia.
- Voliteľné dedenie oprávnení podľa statickej štruktúry.

Nevýhody:

- Pre niektoré nastavenia (vytvorenie a správa rolí) je nutné použiť ZMI, iné sú dostupné aj v Plone aj v ZMI.
- Priradenie lokálne roly je často spojené s tým, že užívateľ (ak nemá prístup do ZMI) nevie aké oprávnenia vlastne poskytuje.
- Roly a skupiny sú nezávislé a tvoria jednoduchú množinu.

Typo3

Kontrola prístupu pre komponenty (stránky) je podobná UNIX-ovému vlastníckemu DAC, teda oprávnenia sú priradené vlastníčkovi, skupine a ostatným. Existuje päť základných operácií na komponentoch:

- čítať
- modifikovať (pod modifikáciou stránky rozumie Typo3 globálne parametre).
- zmazať
- vytvárať komponenty pod aktuálnym komponentom
- modifikovať obsah komponentu. Právo modifikovať obsah na stránke, zaručuje iba potenciálnu možnosť modifikácie, nehovorí nič o druhu obsahu, ktorý smie užívateľ vytvoriť, respektíve akým spôsobom smie byť uverejnený.

Existuje špeciálna skupina užívateľov – administrátorov, ktorý majú neobmedzený prístup a navyše môžu vytvárať skupiny a jednotlivých *backend* užívateľov.

Z UNIX-u preberá myšlienku užívateľských pracovných adresárov. Pre každého užívateľa je nutné definovať vstupný bod do stromovej štruktúry obsahu. Užívateľ má následne v CMS prístup len k jemu pridelenej časti hierarchii stránok.

Definícia operácií je komplikovanejšia. Tak pre skupiny ako aj pre konkrétnych užívateľov je možné vytvoriť množinu im prístupných nástrojov. Nie je však možné povoliť respektíve zakázať priamo jednotlivé akcie nástrojov. Nástroje obsahujú ovládacie prvky a kontrola prístup k nástroju je definovaná ako množina viditeľných ovládacích prvkov. Pre niektoré ovládacie prvky je možné explicitne povoliť alebo zakázať určité hodnoty. To znamená, že ak má mať užívateľ právo publikovať obsah, musí mať prístup k položke *Zobraz/Schovaj*. Rozdelenie práv medzi skupinu a užívateľa je asymetrické. Napríklad spomenutý prístup k ovládacím prvkom nástrojov je možné nastavovať len pre skupinu. Na druhej strane vstupný bod *mount point*, je možné definovať pre skupinu aj pre konkrétneho užívateľa, pričom užívateľa môžu, voliteľne, nastavenie skupiny zdediť. Do určitej miery definuje skupina rolu užívateľa v zmysle RBAC.

K správe *frontend* užívateľov je nutné použiť odlišný nástroj. Kontrola prístupu je v tomto prípade zameraná iba na akciu zobrazenia stránky vo webovom prehliadači. Ktorí konzumenti majú ku stránke prístup je kontrolované pomocou atribútu stránky - skupina. Ak je nastavený, potom ku stránke, voliteľne k celému pod-stromu, majú prístup iba užívateľa, ktorí sú členmi tejto skupiny. Od verzie 4 dovoľuje Typo3 priradenie niekoľkých skupín ku stránke. Konzumenti a ich skupiny sú uložené v špeciálnom druhu stránky – *sysfolder*. Táto stránka nieje publikovaná. Počet *sysfolder*-ov nieje obmedzený a rovnako ako pre iné stránky sa k nim vzťahuje kontrola prístupu, tentokrát pre producentov obsahu. Vďaka tomu môžu producenti spravovať prístup konzumentov ku svojim stránkam samostatne. Interakcia konzumentov s interaktívnym obsahom nieje predmetom správy. Pre konzumentov je navyše typické definovanie častí stromu obsahu, ku ktorým nemajú prístup. Pre producentov platí opak, prístup je kontrolovaný pomocou vstupných bodov – *mount pointov*.

Výhody:

- Vysoká granularita oprávnení.
- Hierarchia skupín.
- Producenti môžu spravovať samostatné skupiny konzumentov.
- Dôležitou funkciou je zobrazenie všetkých stránok a prístupových práv v jednom pohľade.

Nevýhody:

- Pomerne vysoká komplexnosť.
- Nízkoúrovňové oprávnenia na databázových tabuľkách a ovládacích prvkoch.
- Obmedzené DAC pre komponenty.
- Dovoľuje priradiť oprávnenia konkrétnym užívateľom.
- Asymetrické rozdelenie definície užívateľa a skupiny.
- Niektoré parametre nie sú prístupné cez webové rozhranie (napríklad východzie nastavenie oprávnení na stránkach – *umask*).

MoinMoin

Skupiny sú wiki-stránky so špeciálnym menom, štandardne sa končiacom slovom *Skupina*, napríklad *AutorskáSkupina*. Ak je na novom riadku takejto wiki-stránky uvedené meno užívateľa, je zaradený do skupiny. Operácie sú: čítať, modifikovať, zmazať wiki-

stránku, vrátiť predchádzajúcu verziu a právo administrácie. MoinMoin neposkytuje špeciálny nástroj na definíciu práv, ACL sú súčasťou wiki-stránky, nie sú však publikované. Modifikovať ACL je teda možné priamo pri editovaní stránky. Užívatelia a skupiny, ktorí majú právo administrácie majú možnosť ACL vytvárať a modifikovať. Na najvyššej úrovni existujú ešte ACL, ktoré sú aplikované pred a po aplikácií ACL asociovaných s wiki-stránkou a ACL, ktoré sú použité ak stránka žiadne ACL neobsahuje. Modifikovať je ich možné len priamym zásahom do konfiguračného súboru.

Výhody:

- Jednoduché používanie.
- Dobrá granularita.

Nevýhody:

- Obmedzené na obsah.
- Nie je možné definovať ACL na množinách komponentov

TikiWiki

Každá funkčná časť (modul) poskytuje množinu operácií. Tieto môžeme priradiť skupinám, zdrojmi sú následne všetky typy komponentov kontrované daným modulom.

Operácie môžeme explicitne povoliť priradením ACL kategóriám, ale aj konkrétnemu obsahu. ACL sú v tomto prípade zoznamy dvojíc skupina, operácia. Kategórie, ktoré ACL nemajú pridelené, dedia od svojich rodičov. Výsledné ACL pre komponent sú definované ako prienik ACL kategórií komponentu. V prípade, že existuje ACL na úrovni konkrétneho obsahu, rušia sa tým všetky ostatné ACL zdedené z kategórií ako i povolené akcie na úrovni skupín.

Alternatívne je možné v prípade wiki-stránok povoliť DAC model, kde vlastník stránky rozhoduje o kontrole prístupu priamo.

Existuje jediný špeciálny užívateľ administrátor s neobmedzenými právami. Je však možné vytvoriť skupinu, ktorá má povolené všetky akcie, táto je ale limitovaná ACL kategórií a obsahu.

Výhody:

- Vysoká granularita.
- Hierarchické skupiny.
- Založené na logických operáciách.

Nevýhody:

- Nakoľko kategórie nie sú povinné, čo znižuje ich použiteľnosť pre kontrolu prístupu.
- Komplexné pravidlá.

Drupal

V štandardnej inštalácii sú oprávnenia publikované modulmi, pre obsah sú limitované podľa typu komponentu a typu elementu. Rozširujúci modul umožňuje samostatne pre každú taxonómiu a každý pojem nastaviť práva vytvárať, čítať, modifikovať a zmazať komponent v danej kategórii. Ak je komponent súčasne popísaný viacerými kategóriami, výsledné práva sú *zjednotením* čiastkových. Zaujímavá je i možnosť nastaviť predvolené práva pre každú novú kategóriu.

Výhody:

- Založené na logických operáciach.
- S modulom *taxonomy_access* pomerne dobrá granularita.
- Oprávnenia na úrovni elementov.

Nevýhody:

- Bez rozširujúceho modulu, nízka granularita.
- *Taxonomy_access* definuje vlastnú skupinu operácií.
- Iné rozširujúce moduly vyžadujú často zmeny i ďalších moduloch.
- Rozširujúce moduly nie sú oficiálne podporované.

Mambo

Zdroje sú rozdelené do dvoch základných skupín. Prvou sú konkrétne komponenty, druhou je webová stránka definovaná prvkom menu. V oboch prípadoch existujú tri nezávislé podskupiny, a to *public*, *registered*, *special*.

Užívatelia sú rozdelení do dvoch základných skupín *frontend* a *backend*. Toto rozdelenie však nekorešponduje s rozdelením konzument – producent. V rámci *backend* skupiny existuje jednoduchá a fixná hierarchia skupín, ktoré majú priradené práva k administrácii rôznych aspektov CMS. Vo *frontend* skupine existuje podobná hierarchia. *Registered* skupina má priradené právo konzumovať obsah v doméne *registered*. Pridelenie práv podľa skupín je pevne definované:

- Author: právo vytvárať nový obsah a právo modifikovať vlastný.
- Editor: ako Author a navyše právo modifikovať každý obsah.
- Publisher: ako Editor a navyše právo zverejniť obsah.

Súčasne tieto tri skupiny spolu so skupinami v *backend* môžu pristupovať k obsahu v doméne *special*. Užívateľ je zaradený do práve jednej skupiny.

Výhody:

- Jednoduchosť.

Nevýhody:

- Extrémne obmedzené možnosti pre definíciu kontroly prístupu.
- Užívateľské roly a ich oprávnenia sú vopred definované.

Zhrnutie

Tabuľka 42: Kontrola prístupu

CMS	Nositeľ oprávnení	Hierarchia	Priradenie užívateľov	Oprávnenia
Bricolage	Skupina	Nie	N skupín	- ACL na množinách objektov.
Drupal	Rola	Nie	N rolí	- Typ komponentu, elementu - ACL pre kategórie, komponenty dedia z kategórií.
Plone	Rola	Nie	N rolí N skupín skupina = N rolí	- ACL komponentov, prípadne dedené zo statickej štruktúry
EzPublish	Rola	Nie	N rolí N skupín skupina = N rolí skupiny sú hierarchické	- Typ komponentu, elementu - ACL komponentov, prípadne dedené zo statickej štruktúry
Mambo/Joomla	Skupina	Čiastočne	1 skupina	- Komponenty
Typo3	Skupina/Užívateľ	Áno	N skupín	- Pre komponenty vlastnícke ACL. - Typ elementu (element je možné vytvoriť/modifikovať len ak má užívateľ oprávnenie modifikovať komponent)
MediaWiki	Skupina	Áno	N skupín	- ACL pre namespace, komponenty dedia ACL z namespace.
Magnolia	Skupina	Áno	N skupín	- ACL komponentov, prípadne dedené zo statickej štruktúry
MoinMoin	Skupina/Užívateľ	Nie	N skupín	- ACL komponentov
TikiWiki	Skupina	Áno	N skupín	- ACL komponentov, prípadne dedené zo štruktúry kategórií. - Typ komponentu, ak ACL neexistuje
Wordpress	Skupina	Nie	1 skupina	Nie

Kontrola prístupu mimo CMS

Jednou z úloh kontroly prístupu je rozhodnúť o tom kto môže aký obsah konzumovať. Nehovorí však akým spôsobom je možné obsah ďalej použiť. V tomto smere existujú dve techniky.

Prvou je kontrolu využitia obsahu spojiť so samotným obsahom, hovoríme o takzvanom *Digital Rights Management*, digitálna správa práv. Myšlienkou je zabezpečiť, aby bol obsah používaný iba presne definovým spôsobom. Ako príklad uveďme právo *vytlačiť* určitý dokument, alebo časový interval, v ktorom je možné video obsah konzumovať. Žiadne z CMS v našom prehľade integrované DRM neposkytuje. Jedným z dôvodov je i negatívny postoj opensource komunity k DRM.

Druhou možnosťou je explicitne uviesť povolené druhy použitia obsahu pomocou licencie. Licencia môže byť jediná pre všetok obsah prístupný cez CMS, alebo priradená zvlášť každému elementárnemu obsahu. Integrovanú správu licencií poskytuje TikiWiki a

MediaWiki. Štandardne však túto funkciu môžeme implementovať v ľubovoľnom CMS, ktoré podporuje typy komponentov a to pridaním povinného elementu – zoznamu povolených licencií.

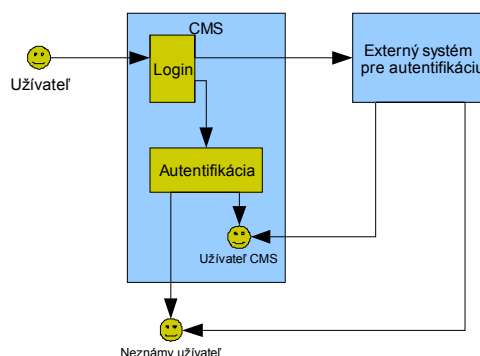
5.1.2 Autentifikácia a vytváranie nových užívateľov

Definícia: Autentifikácia je proces verifikácie identity užívateľa. Všeobecnejšie je to proces, ktorý umožňuje jednej strane dokázať svoju identitu druhej strane.

Autentifikácia užívateľa voči počítaču, môže byť založená na:

1. niečo čo užívateľ vie, najčastejšie meno a heslo,
2. niečo čo užívateľ má, čipová karta,
3. niečo čo užívateľ je, biometrické dáta: odtlačok prsta a podobne.

V prípade CMS v našom porovnaní (prakticky však väčšina CMS) sa využíva prvá metóda. Autentifikácia existuje v CMS v dvoch druhoch, priama a nepriama. Priama autentifikácia využíva techniky HTTP protokolu. Nakoľko sa heslo prenášané v textovej podobe poskytujú niektoré CMS podporu SSL. Nepriamu autentifikáciu je žiadateľné využiť v prípade, že CMS je nasadzovaný v rámci existujúcej informačnej infraštruktúry organizácie. CMS v tomto prípade deleguje autentifikáciu na existujúci mechanizmus. V druhom prípade spravuje užívateľov priamo CMS, čo zahŕňa autentifikáciu, teda kontrolu mena a hesla, ako aj vytváranie, modifikáciu a mazanie užívateľov.



Obrázok 14:

Vytvorenie užívateľa je možné explicitnou akciou oprávneného užívateľa, okrem toho CMS podporujú registráciu cez webové rozhranie. Užívateľ vyplní registračný formulár a následne:

- je prihlásený a môže okamžite využívať povolené služby. V tomto prípade je nutné použiť ochranu voči programom, ktoré sú schopné vyplniť takýto formulár automaticky. Klient teda musí vykonať operáciu, ktorú nieje možné efektívne automatizovať (v zmysle aktuálne dostupných informačných technológií), napríklad zadať kontrolný reťazec, ktorý je prezentovaný ako obrázok prípadne zvuk.
- systém pošle na zadanú e-mailovú adresu e-mail, ktorý obsahuje jedinečný webový odkaz. Aktivácia konta je podmienená prístupom k tejto stránke prostredníctvom webového prehliadača. Neaktivované kontá sú v systéme zvýraznené, a je možné ich automaticky zmazať ak neboli aktivované v určenom časovom okne. Takéto riešenie však zabráňuje automaticky generovaným kontám len čiastočne. Program, ktorý kontroluje e-mailový server dokáže vytvárať kontá automaticky.

- o jeho požiadavke musí rozhodnúť oprávnený užívateľ. Až potom sa môže prihlásiť.
- Rozšírené sú moduly, umožňujúce filtrovať e-mailové adresy, teda vytvárať filtre, ktoré zabránia vytvoreniu nového užívateľského konta, s určitou e-mailovou adresou. Podobne možno vyžadovať jedinečnosť užívateľských e-mailových adries.

Tabuľka 43: Autentifikácia a bezpečný kanál

CMS	Priamo	Nepriamo	Bezpečný kanál (SSL)
Bricolage	HTTP	- LDAP súčasť inštalácie	- V konfiguračných súboroch. - Celý portál, alebo iba autentifikácia a zmena užívateľského konta.
Drupal	HTTP	- LDAP modul	- V konfiguračných súborov. - Len celý portál.
EzPublish	HTTP	- LDAP súčasť inštalácie	- Transparentné. Odporúčame použiť mod_rewrite Apache serveru, pre nezávislé používanie SSL pre autentifikácia.
Mambo	HTTP	- LDAP patch	- Transparentné. - Len celý portál.
MoinMoin	HTTP		- Len celý portál.
Plone	HTTP	- LDAP modul	- Len celý portál, existuje však modul pre obmedzenie na autentifikáciu.
TikiWiki	HTTP - challenge/response (konfigurovateľné)	- LDAP súčasť inštalácie	- Konfigurovateľné. - Pre prihlásenie, voliteľne i pre celý portál.
Typo3	HTTP - challenge/response (vždy)	- LDAP modul	- Transparentné. - Len celý portál.
Wordpress	HTTP		- Transparentné. - Len celý portál.

5.1.3 Audit

Pod pojmom audit rozumieme verifikáciu stavu CMS, teda dodržiavanie definovaných pravidiel, ako aj odhalenie problémov. Na jednej strane CMS musí poskytovať nástroj na sledovanie akcií, ktoré užívatelia v rámci CMS vykonávajú, teda priradenie zodpovednosti, na druhej strane musí CMS upozorniť na výskyt (potenciálnych) problémov. Relatívne priamočiarou implementáciou je záznam akcií (log). Záznam akcií predstavuje zoznam správ generovaných pri určitých udalostiach rôznymi komponentami CMS. Druhou možnosťou je využiť správu histórie obsahu, táto je však limitovaná na priradenie zodpovednosti, ide teda o komplementárnu funkciu.

Tabuľka 44: Sledovanie akcií

CMS	Záznam	Záznam
Bricolage	Druh udalosti, dátum, užívateľ	- Nezávisle pre každý objekt.
Drupal	druh udalosti, dátum, popis, užívateľ, akcie	- Prehľadný záznam akcií správy a prístupu k obsahu (vytváranie, modifikácia) - Filtrovanie
EzPublish	-	-
Mambo	-	- Žiadny webový nástroj - Využíva zabudované nástroje oznamovania chýb prostredia PHP, je možné nastaviť iba úroveň oznamovaných chýb.
MoinMoin	-	- Len história wiki-stránok.
Plone		- Len záznam chýb.
TikiWiki	druh udalosti, dátum, popis, užívateľ, webový prehliadač, IP Adresa	- Záznam je obmedzený na akcie správy CMS. - Pre wiki-stránky je možné použiť históriu ako audit nástroj, vytváranie a modifikácie ostatného druhu obsahu nie sú zaznamenávané.
Typo3	druh udalosti, dátum, popis, užívateľ	- Prehľadný záznam, sledovanie prístupu k obsahu je na úrovni tabuliek relačnej databázy. - Filtrovanie
Wordpress	-	-

5.2 Správa procesov

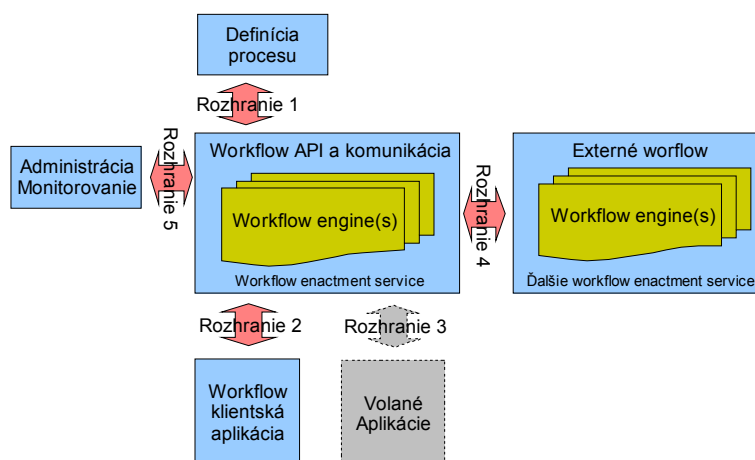
Kolaborácia pri vytváraní obsahu ako i kvalita obsahu sú dva dôležité faktory prečo nasadiť CMS. Pozrime sa na možný postup pri vytváraní finálnej publikácie: autor v niekoľkých krokoch vytvorí nový článok, editor skontroluje obsah, prípadne navrhne prepracovanie. V ďalšom kroku dizajnér vloží obrázky, prípadne navrhne zalomenie textu. Finálny článok je zodpovedným užívateľom publikovaný. Prezentovaný postup je príkladom *business process*.

Definícia: *Workflow* je automatizovaná forma časti alebo celého *business process*, počas ktorej sú dokumenty, informácie alebo úlohy presúvané medzi účastníkmi, podľa množiny procedurálnych pravidiel.

Definícia: *Aktivita* je najmenšia časť práce, reprezentujúca logický krok v procese.

Definícia: Aktivity sú spojené *prechodmi*, ktoré môžu byť nepodmienené – ukončenie aktivity vedie vždy k spusteniu inej, alebo podmienené – vykonanie závisí od jednej alebo viacerých *prechodových podmienok*.

Definícia: *Workflow management system* definuje, vytvára a spravuje vykonávanie workflow, pomocou software, vykonávanom na jednom alebo niekoľkých *workflow engine*, pričom je schopný interpretovať definíciu procesov, ovplyvňovať účastníkov, v prípade potreby využíva nástroje a aplikácie informačnej technológie.

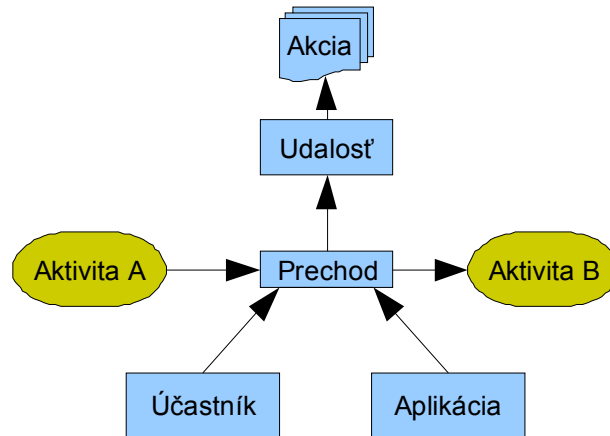


Obrázok 15: WFMS- Referenčný-Model podľa WfMC

Definícia: Workflow engine poskytuje vykonávanie *business processu*. [24]

Vzťah CMS a *workflow engine* môže byť dvoch druhov, externý a interný. V prvom prípade je *workflow engine* nezávislou aplikáciou, ktorá cielene komunikuje s CMS a inými aplikáciami. Inými slovami iba časť vykonávaného workflow je zameraná na CMS. V druhom prípade je *workflow engine* komponentom CMS, všetky workflow sú špecifické pre CMS. CMS rozdeľujeme na:

- **Workflow-based**, všetky akcie sú súčasťou niektorého workflow, *workflow engine* je kľúčovým komponentom.
- **Workflow-enabled**, workflow je voliteľným komponentom.



Obrázok 16: Schéma workflow

Cielom workflow je teda formalizovať *business process*. Hlavným cieľom je zvýšiť kvalitu, vďaka presnému postupu, ako aj efektívnosť procesu, priradením a monitorovaním potrebných akcií užívateľom.

Obrázok 16 predstavuje zjednodušenú schému workflow. Workflow sa teda skladá z aktivít, ktoré sú prepojené prechodmi.

Každý prechod môže byť:

- explicitný, vykonaný užívateľom.
- implicitný, vyvolaný automaticky aplikáciou.

Navyše každý prechod predstavuje zdroj udalosti, ku ktorému sa môžu viazať akcie. Aktivity a prechody teda tvoria orientovaný graf.

V prípade, že aktivita má viac ako jeden vychádzajúci prechod nastávajú dva prípady:

- *AND split* – súčasne prechod do všetkých cieľových aktivít.
- *XOR split* – podmienený prechod do práve jednej cieľovej aktivity.
- *OR split* – podmienený súčasný prechod do ľubovoľnej podmnožiny cieľových aktivít.

Opačný prípad, keď aktivita má niekoľko vstupných prechodov je komplikovanejší:

- *AND join* – Aktivita sa začne až po použití všetkých vstupných prechodov. Čo sa stane v prípade, že sa niektorý prechod nepoužije nieje definované a závisí od implemntácie.
- *OR join* – Aktivita sa vykoná po použití ľubovoľného prechodu. Predpokladom je, že prechody, ktoré spája nemôžu byť vykonané paralelne.

Definícia: Pod *inštanciou* workflow rozumieme konkrétne dokumenty, informácie či akékoľvek iné dáta vytvorené a používané v priebehu vykonávania aktivity.

Implementácia workflow managementu v CMS môže byť zaradená do dvoch skupín. Prvou je obmedzená forma, CMS poskytuje niekoľko preddefinovaných workflow, druhou je neobmedzená forma, kde CMS poskytuje nástroj na definíciu workflow. Prvá forma je implementovaná najčastejšie ako model vytvorenie-kontrola-publikácia. CMS poskytuje nástroje na definíciu účastníkov (najčastejšie pomocou rolí) a obsahu, ktorý je takto spravovaný.

5.2.1 Jednoduché workflow

Tabuľka 45: Obmedzená verzia workflow

CMS	
Drupal	- Vytvorenie je jednorázová akcia, neexistuje <i>draft</i> aktivita. - Kontrola je zabezpečovaná modulom moderation queue. - Publikácia je riešená atribútom Published.
EzPublish	- Ľubovoľný počet <i>draft</i> verzii komponentu, oprávnený užívateľia môžu <i>draft</i> verziu publikovať.
Mambo	- Existuje špeciálna aktivita novovytvoreného obsahu koncept – <i>draft</i> . V rámci tejto aktivity môže modifikovať obsah iba autor. - Kontrola je riešená interným notifikačným systémom, zodpovedný užívateľ dostane správu o novom obsahu. Aktivita kontrola je pevne priradená užívateľskej role. - Publikácia – ako Drupal
Typo3	- Komponent (page) má operáciu zmeny viditeľnosti. Priradením oprávnenia k tejto operácii dostávame jednoduché dvojtavové workflow – vytvorenie a kontrola. - Ďalšou funkciou je systém workspace [54]. Môžeme vytvoriť ľubovoľný počet workspace. V rámci workspace je nový a modifikovaný obsah nezávislý. K workspace pridáme množinu užívateľov (alebo skupín), v rolách editorov, kontrolórov a vlastníkov. Kontrolóri a/alebo vlastníci workspace môžu nový a modifikovaný obsah selektívne publikovať. Typo3 poskytuje silné nástroje pre správu workspace ako aj obsahu vo workspace. Napríklad dovoľuje graficky porovnať obsah vo workspace s aktuálnou publikovanou verziou, zavesenie poznámok k obsahu a umožňuje komunikáciu medzi užívateľmi vo workspace.
Wordpress	- Vytvorenie a <i>draft</i> aktivita, kontrola, publikácia ako v prípade Mambo. - Podobne ako Drupal existuje zoznam obsahov, ktoré čakajú na kontrolu a publikáciu. - Podobne ako Mambo sú jednotlivé aktivity pevne priradené užívateľským rolám.

5.2.2 Komplexné workflow

Plnohodnotnú implementáciu workflow managementu v súčasnosti podporujú Bricolage, Drupal, Plone a TikiWiki.

Bricolage

Workflow management je založený na metafore „stolov“. Každý komponent (ako aj binárne súbory – media a šablóny) existuje na niektorom zo stolov. Medzi stolmi sa objekty presúvajú tak, že užívateľ spraví *check-out*, modifikuje objekt, *check-in* na iný alebo rovnaký stôl. So stolom sú asociované ACL, pričom užívateľ môže spraviť *check-out* ak má oprávnenie písať pre stôl a *check-in* ak má oprávnenie čítať. Prakticky sú stoly aktivity, pričom prechody sú implicitné, odvodené z ACL. Práve jeden stôl je začiatkový.

Výhody:

- Jednoduchá ale silná metafora.
- Integrované s kontrolou prístupu, ako aj s vytváraním obsahu. Obsah je možné vytvárať a modifikovať len v rámci konkrétneho workflow.

Nevýhody:

- Nedovoľuje priradiť akcie prechodom (nakolko však podporuje všeobecné akcie, je možné tento nedostatok obísť).
- Prechody sú implicitné, založené na oprávneniach užívateľov, čo mierne zvyšuje komplexitu.

Drupal

Inštanciou workflow je vždy iba jeden konkrétny komponent. Workflow sú založené na dvoch rozširujúcich moduloch *workflow* a *actions*. Modul workflow poskytuje rozhranie pre definíciu aktivít (state) a prechodov. Súčasne umožňuje definovať množinu rolí, ktoré sú oprávnené prechod medzi aktivitami vyvolať. Pomocou modulu *actions*, ako už meno prezrádza, môže oprávnený užívateľ definovať akcie. Zámerom autorov Drupal je, aby jednotlivé moduly systému zverejňovali množiny povolených akcií. Príkladom akcie je napríklad publikácia obsahu, zverejnenie na úvodnej stránke, alebo odoslanie e-mailu. K danému prechodu je potom následne možné priradiť množinu akcií. Úlohou v rámci aktivity je modifikácia konkrétneho komponentu, respektíve metadát. Aktivitou teda nemôže byť napríklad vytvorenie užívateľa, alebo zálohovanie databázy. V tomto zmysle je teda konkrétny obsah inštanciou workflow.

K prehľadu o tom, ktorý obsah sa v akej aktivite nachádza je možné použiť modul *views*. Tento modul umožňuje vytvárať zoznamy obsahov podľa filtra. V kooperácii s modulom workflow je možné vytvoriť zoznam obsahov v konkrétnej aktivite.

Výhody:

- Jednoduchosť vytvárania workflow, ako aj pridávanie akcií. Modul *views* umožňuje jednoduchý prístup k obsahu vo workflow.
- Preddefinované akcie a API pre vytváranie akcií.
- Moduly môžu dodávať vlastné definície workflow (napríklad modul *i18n* dodáva workflow pre preklad)

Nevýhody:

- Workflow modul v súčasnosti neposkytuje možnosť súčasného vykonávania aktivít. Komponent je zaradený práve jedného workflow.
- Problematická je kontrola prístupu k obsahu v jednotlivých aktivitách.

Plone

Workflow management ako aj niekoľko jednoduchých workflow je súčasťou inštalácie. Workflow je priradené podľa typu komponentu. Inštanciou workflow je komponent. Vytvárať a modifikovať workflow môžeme pomocou grafického nástroja v ZMI. Aktivity (state) môžu byť spojené s ľubovoľným počtom iných aktivít pomocou prechodov, ktoré sú buď automatické alebo manuálne. Komponenty vo workflow začínajú v aktivite označenej ako začiatočná, ktorá musí byť práve jedna. Pre každú aktivitu môžeme nastaviť ACL, ktoré voliteľne nahrádza štandardné ACL. Podobne môžeme asociovať ACL aj s prechodmi. Prehľadný zoznam komponentov v rámci workflow poskytuje worklist.

Výhody:

- Interaktívny nástroj na vytváranie.
- Integrované s kontrolou prístupu.
- Používanie workflow je integrované do užívateľského rozhrania.
- Poskytuje API pre rozšírenie funkcionality. Pre každý prechod môžeme definovať funkciu (v jazyku python), ktorá sa má vykonať pred a po prechode. V rámci workflow môžeme definovať premenné, globálne pre celé workflow ako aj asociované s konkrétnym komponentom.
- Definíciu workflow môžeme exportovať (a importovať) ako zdrojový kód v jazyku python.

Nevýhody:

- Problematické priradenie ACL prechodu (je nutné vypísať meno operácie a roly).
- Žiadne preddefinované akcie, je nutné ich naprogramovať.
- Aj keď poskytuje automatické prechody, systém vyberie vždy práve jeden.

TikiWiki

TikiWiki dodáva workflow management ako súčasť inštalácie. V porovnaní s Drupal poskytuje viac funkcií, ale za cenu zvýšenej komplexnosti. Základ predstavuje *proces*. K procesu sú priradené aktivity, a to automatická a interaktívna. K obom druhom aktivít je nutné vytvoriť vykonateľný kód, a samozrejme definovať prechody. Podobne ako Drupal existuje počiatková akcia, narozdiel od Drupal však definuje i cieľovú akciu (tak počiatkových ako i cieľových aktivít môže byť viac). K interaktívnej aktivite je navyše nutné vytvoriť i grafické prostredie (pomocou šablónového systému). Ďalšou výhodou oproti Drupal je i možnosť definovať paralelné aktivity. Táto funkcia je zabezpečená pomocou aktivít typu *split* a *join*, pričom aktivita typu *split* je spojená prechodmi s množinou ďalších aktivít, ktoré budú prebiehať paralelne. Aktivita typu *join* je vlastne opakom *split*, je spojená s množinou paralelných aktivít, súčasne zabezpečuje, že ďalší prechod je možný až po ich skončení.

Výhody:

- Výhody sú pomerne veľká množina funkcií, je možné vytvárať verzie procesov, graficky znázorniť daný proces, ako i import a export definície procesu.
- V priebehu definície procesu, systém kontroluje aktuálny stav a oznamuje chyby, ako napríklad neexistujúca cesta medzi počiatkovou a cieľovou aktivitou.

Nevýhody:

- Potreba znalosti PHP a Smarty ako i architektúry TikiWiki, keďže úloha, ktorá ma byť v rámci akcie vykonaná, musí byť buď naprogramovaná.
- Negatívne musíme ohodnotiť i užívateľské rozhranie, ktoré je síce plné funkcií, ale neintuitívne. Mnohé funkcie sú reprezentované iba obrázkom, niektoré pokročilé funkcie by si zaslúžili samostatné rozhranie.
- Účastníkov je nutné vybrať zo všetkých registrovaných, nieje možné znovupoužitie definovaných skupín v rámci TikiWiki.

Zhrnutie

Žiadny zo systémov neposkytuje možnosť vytvoriť workflow pre vyšší logický celok, ako napríklad vytvorenie obsahu a z neho niekoľkých publikácií. Súčasne je systém workflow obmedzený na správu obsah (až na Bricolage). Ako čiastočnú nevýhodu vnímame nemožnosť vytvoriť paralelne aktivity (AND split) (až na TikiWiki), čo však vzhľadom na fakt že predmetom workflow je komponent nieje až také podstatné. Žiadna implementácia nepodporuje model externého *workflow-engine*.

Veľkou výhodou by bola možnosť využitia niektorého štandardného jazyka pre definíciu workflow, napríklad XPD [25].

Tabuľka 46: Komplexné workflow

CMS	Inštancia	Nástroj na vytváranie	Pred-konfigurované / Povinné	Akcie
Bricolage	Komponent, Šablóna, binárny súbor (media).	Áno	Áno/Áno	Áno (samostatné)
Drupal	Komponent	Áno	Nie/Nie	Áno
Plone	Množina komponentov, podľa typu.	Áno	Áno/Nie	Áno, nutné naprogramovať
TikiWiki	Komponent	Čiastočne	Nie/Nie	Áno

5.2.3 Alternatívne možnosti správy obsahu

V tejto kapitole sa pozrieme na možnosti správy obsahu, ktoré nemajú formálny charakter procesu. V niektorých prípadoch sa systém *workflow* model môže prejavovať ako obmedzujúci faktor. Striktné dodržiavanie procesu totiž môže viesť k oneskoreniu uverejnenia obsahu, súčasne v prípade rozsiahlej skupiny autorov, nieje často možné kontrolovať každý obsah. Hlavne v prípadoch, kde autori a konzumenti obsahu predstavujú rovnakú skupinu, je možné proces kontroly obsahu delegovať na konzumentov. Tento model sa využíva v prípade diskusných fór ako i v prípade wiki. Úlohou CMS v tomto prípade je poskytnúť nástroje užívateľom ako nevhodný obsah objaviť a vhodným spôsobom upraviť. Nevhodný obsah môžeme rozdeliť do týchto kategórií: SPAM, vulgárny obsah, nesprávny obsah.

1. SPAM podobne ako v prípade e-mailu definujeme ako nevyžiadaný obsah. Narozdiel o e-mailového SPAM-u má niekoľko podôb.
 - Obsahový SPAM: najčastejšie reklama určitého produktu alebo služby.
 - Odkazový SPAM: obsah pozostávajúci prevažne z webových odkazov. Tento druh SPAM-u zneužíva spôsob vyhodnocovania relevancie webových stránok určitými vyhľadávacími programami.Spoločným menovateľom je, že tento druh obsahu je vytváraný alebo vkladný do CMS automaticky určitou aplikáciou. V súčasnosti existujú techniky ako zabrániť alebo obmedziť vstup obsahu z tejto kategórie. Prehľad riešení je však nad rámec tejto práce.
2. Vulgárny, *politicky nekorektný obsah*, prejavuje sa použitím určitých výrazov, podobne ako v prípade SPAM-u je možné takýto druh obsahu automaticky odhaliť a zabrániť vstupu (vyhľadávaním nevhodných výrazov a slov).

3. Nesprávny obsah je kategória, ktorú nieje možné automaticky odhaliť, nesprávnosť obsahu totiž plynie z významu.

V štúdií [12] autori uvádzajú štatistiku, podľa ktorej, obsah z prvej a druhej kategórie bol explicitne odstránený zo stránok Wikipédie v priebehu niekoľkých minút. Existujú však prípady kde obsah z tretej kategórie bol prístupný až niekoľko mesiacov. Prvým publikovaným prípadom bol článok o *John Seigenthaler*, v ktorom bol zámerne modifikovaný jeho životopis [26]. Ďalším prípadom boli články modifikované zamestnancami Kongresu USA (došlo k vymazaniu niektorých nepriaznivých informácií zo životopisu politikov) [27].

K objaveniu nevhodného obsahu poskytujú CMS notifikačné funkcie. Užívateľ môže definovať množinu obsahu, o ktorej zmene chce byť notifikovaný. Prakticky to znamená odoslanie e-mailu, notifikáciu pri prihlásení (takzvaný *Watch List*), alebo ako sme už načrtli v predchádzajúcej kapitole využitie *web-feed*.

Riešenie nevhodného obsahu v prípade wiki je relatívne jednoduché, obsah sa prepíše predchádzajúcou verziou. V prípade diskusných fór je možné i toto riešenie, okrem radikálneho odstránenia existuje i ďalšia metóda, založená na ohodnocovaní obsahu. Registrovaný užívateľ má určitý počet bodov, ktoré môže využiť k ohodnoteniu obsahu. Najčastejšie môže využiť bod k pozitívnemu ako aj negatívnemu ohodnoteniu. Súčet pozitívnych ako i negatívnych bodov definuje hodnotu obsahu. CMS následne umožňuje užívateľom definovať bodovú hranicu, obsah s hodnotením pod touto hranicou nieje prezentovaný. Počet bodov, ktoré má konkrétny užívateľ k dispozícii závisí od CMS. Pomerne rozšírené sú riešenia, kde je tento počet funkciou hodnotenia kvality obsahu vytvoreného daným užívateľom. Hodnotenie obsahu podporuje TikiWiki.

Tabuľka 47: Notifikácie

CMS	Notifikácia	Web	E-mail	Web-feed
Bricolage	Pre každý objekt v CMS nielen obsah. Komplexné výrazy pre definíciu podmienok.	Áno	Áno	Nie
Drupal	Komponent	Áno	Áno, modul Notify	Áno, modul Views
EzPublish	Komponent a podstrom statickej hierarchie	Nie	Áno	Nie
Mambo	Komponent	Áno	Áno	Nie
MediaWiki	Komponent	Áno	Áno	Nie
MoinMoin	Komponent	Áno		Áno, súčasťou sú zmeny
Plone	Zavisí od modulu, respektíve vlastnej implementácie			
TikiWiki	Komponent	Nie	Áno	Nie
Typo3	Podstrom statickej štruktúry v rámci workspace.	Áno	Áno	Nie
Wordpress	Komponent	Áno	Áno, modul Email notification	

5.3 Spolupráca pri vytváraní obsahu

Dôležitou vlastnosťou CMS je umožniť užívateľom kooperáciu pri vytváraní a správe obsahu. V tejto časti sa pozrieme na problémy a riešenia, ktoré v tomto ohľade CMS poskytujú.

5.3.1 Súčasná modifikácia obsahu

Táto časť je venovaná prehľadu riešení problému súčasnej modifikácie rovnakého obsahu viacerými užívateľmi. V tomto kontexte sa často spomína problém *stratenej aktualizácie (lost update)* [16]. Tento problém nieje limitovaný pre CMS, ide o inherentný problém systémov s distribuovaným vytváraním obsahu (napríklad *version control* aplikácie ako CVS a Subversion, MS SourceSafe).

Príklad:

1. Adam začne modifikovať dokument (v našom prípade ľubovoľný konkrétny obsah).
2. Eva začne modifikovať rovnaký dokument.
3. Adam ukončí prácu na dokumente a uloží ho.
4. Eva ukončí prácu na dokumente a uloží ho.
Doplňme, že výraz „začať modifikovať“ sa často označuje ako *checkout*, „ukončenie a uloženie“ sa nazýva *checkin*.

Z príkladu je vidno, že Adamove zmeny nebudú trvalé, lebo budú prepísané dokumentom, ktorý modifikovala Eva (ktorý Adamove zmeny ešte neobsahoval).

Riešenia:

1. Komunikácia mimo rámca CMS. Samozrejme toto riešenie z pohľadu CMS v skutočnosti problém vôbec nerieši, spomíname ho len pre úplnosť.
2. Neexkluzívny *checkout* s automatickou detekciou a manuálnym riešením konfliktov pri *checkin-e*. V tomto prípade CMS zabráni uloženiu zmien v kroku 4. Eva musí konflikt vyriešiť manuálne (teda spojiť zmeny Adama so svojimi).
3. Skoré varovanie o potenciálnych konfliktoch v budúcnosti. V kroku 2. je Eva oboznámená o *checkout-e* Adama. Samotný problém však nerieši, preto sa najčastejšie používa v spojení s technikou 2.
4. Exkluzívny *checkout*. CMS zabráni Eve modifikovať dokument pokiaľ Adam nevykoná *checkin*. Problémom zostáva čo v prípade, keď Adam zabudne na *checkin*, respektíve dôjde k chybe a Adam nemôže *checkin* vykonať. Najčastejšie preto existuje nejaký spôsob ako exkluzívny *checkout* obísť (časové obmedzenie, administrátorský príkaz).

Tabuľka 48: Riešenie problému stratenej aktualizácie

CMS	Metóda	
Bricolage	4.	<i>Checkout</i> a <i>check-in</i> sú explicitné akcie. Komponent nie je možné modifikovať bez <i>check-out-u</i> . Komponenty v stave <i>check-out</i> sú označené menom užívateľa.
EzPublish	3.	Správa oboznamujú užívateľa o prebiehajúcom <i>checkout-e</i> . Užívatelia môžu paralelne pracovať na vlastných verziách. Systém dovoľuje manuálne spojiť verzie, alebo prepísanie jednej verzie druhou.
Drupal	2.	Správa pri ukladaní oboznámi užívateľa o konflikte. Negatívne hodnotíme, že ani opakovaný pokus o uloženie nie je možný.
Mambo	4.	Graficky zvýraznený <i>checkout</i> komponentu. Administrátorský príkaz zruší všetky aktuálne <i>checkout-y</i> .
MediaWiki	2.	Časovo limitovaný <i>checkout</i> . Semi-automatické riešenie konfliktu. Pri konflikte vygeneruje MediaWiki grafický náhľad rozdielov a stránku s obidvoma konfliktnými komponentmi.
MoinMoin	2.	Časovo limitovaný <i>checkout</i> . Semi-automatické riešenie konfliktu. Pri konflikte vygeneruje MoinMoin špeciálnu verziu obsahu so zmenami od oboch užívateľov. Po odstránení konfliktu môže užívateľ obsah uložiť.
Plone	1.	Konflikty nie sú riešené. Problém bude/je vyriešený vo verzii Plone 3 [55].
TikiWiki	3.	Konfigurovateľný časový interval, v ktorom sú užívatelia notifikovaný správou o prebiehajúcom <i>checkout-e</i> . Konflikty nie sú riešené.
Typo3	3.	Správa oboznamujú užívateľa o prebiehajúcom <i>checkout-e</i> . Žiadne riešenie konfliktov.
Wordpress	1.	Konflikty nie sú riešené.

Riešenie Mambo a Typo3 vyžaduje explicitné ukončenie (*checkin*) aj v prípade, že užívateľ nijaké zmeny nevykoná. Toto však narušá základné postupy pri práci na internete, napríklad využitie tlačítka *spät*. Čiastočne by sa tento problém dal vyriešiť využitím *ECMAScript-u*. Pri pokuse opustiť stránku (akciou *spät*, explicitným zadaním novej adresy), by bol užívateľ vyzvaný k uloženiu, respektíve zamietnutiu zmien. Nevýhodou riešenia je, že vyžaduje podporu webového prehliadača. (Navyše riešenie je relatívne komplikované a špecifické pre každý prehliadač. Jednou z možností je použitie skrytej inštancie prehliadača, ktorá opakovane kontroluje adresu hlavnej inštancie.) Problém prerušenia spojenia alebo chyby prehliadača však nie je možné odhaliť.

Z týchto dôvodov odporúčame využitie 2. metódy v kombinácii s 3. pričom *checkout* by bol sledovaný v určitom časovom intervale. Vylúčenie stratenej aktualizácie je v tomto prípade závislé od konkrétnej implementácie, napríklad dĺžky intervalu, v ktorom sa obsah považuje za v stave *checkout*. Dôležité sú aj nástroje pre riešenie potenciálnych konfliktov.

5.3.2 História modifikácií

V predchádzajúcej kapitole sme hovorili o probléme stratenej aktualizácie. Ak takýto prípad nastane, je dobré ak CMS poskytujú možnosť vrátiť sa k predchádzajúcej verzii. Samozrejme mechanizmus návratu k starším verziám má využitie i v iných prípadoch. Príkladom môže byť neskoro zistená nevhodná zmena, ale aj prisúdenie zodpovednosti za konkrétny obsah. Využiť správu verzií je možné i na štúdium správania užívateľov.

Jednoduchým riešením je časté zálohovanie obsahu. Nevýhodou je problematický prístup ku konkrétnemu obsahu (napríklad v prípade, že sa zálohuje celá databáza), súčasne nieje často technicky možné vytvoriť zálohu pri každej zmene. Riešenie, ktoré CMS poskytujú je sledovanie histórie pre jednotlivé obsahy.

Ako sme už povedali história obsahu a jednoduchý návrat je jednou zo základných vlastností wiki. Preto nieje prekvapujúce, že tak TikiWiki ako MoinMoin túto funkciu podporujú. (História niektorých článkov Wikipédie má dnes viac ako 1000 prvkov.) Domnievame sa, že správa histórie komponentov i keď v obmedzenej podobe, je dôležitou vlastnosťou CMS.

Tabuľka 49: História modifikácií

CMS	
Bricolage	<ul style="list-style-type: none">- Grafické porovnanie ľubovoľných verzií.- Nahradenie aktuálnej staršou verziou.- Nová verzia vznikne až po <i>checkin-e</i>.- Komentár
Drupal	<ul style="list-style-type: none">- Voliteľne dovoľuje vytvárať <i>revízie</i> obsahu. Administrátor môže vytváranie revízií vyžadovať.- Neposkytuje možnosť porovnania <i>revízií</i>.
EzPublish	<ul style="list-style-type: none">- Grafické porovnanie, ľubovoľných verzií.- Jednotlivé verzie okrem aktuálne publikovanej je možné zmazať. Staršie verzie je možné duplikovať a následne upravovať.- Prehľadné, silné a ľahko pochopiteľné rozhranie.
MoinMoin	<ul style="list-style-type: none">- Zobrazuje posledných 100 verzií, archív verzií je však neobmedzený.- Komentár- Grafické zobrazenie rozdielov ľubovoľných dvoch verzií.
MediaWiki	<ul style="list-style-type: none">- Grafické zobrazenie rozdielov ľubovoľných dvoch verzií.
Plone	<ul style="list-style-type: none">- Globálna história všetkých zmien (nielen obsahových).- Vytváranie verzií je možné doplniť pomocou vytvorenia nového workflow.
TikiWiki	<ul style="list-style-type: none">- Konfigurovateľný počet verzií.- Podpora malej zmeny (<i>minor version change</i>) manuálne.- Komentár- Grafické porovnanie, ale len s aktuálnou verziou.
Typo3	<ul style="list-style-type: none">- Grafické porovnanie bezprostredne nasledujúcich verzií.- Umožňuje spojiť zmeny niekoľkých verzií do jednej (pre potreby porovnania ako i z dôvodu šetrenia miesta).

Ďalšie dôležité funkcie správy verzií sú komentár a detekcia malej zmeny. Komentár je krátky text, ktorý môžu autori použiť ako popis zmeny, napríklad dôvod prečo zmenu vykonali, alebo aké ďalšie akcie sú nutné. Malá zmena (*minor version change*) je označenie, ktoré znamená, že zmeny oproti predchádzajúcej verzii obsahu sú minimálne, napríklad zmena dizajnu alebo gramatické opravy. Štandardne je použitie tohto označenia plne v rukách autora, nakoľko je zrejme že i malá syntaktická zmena môže byť zásadná z hľadiska významu. Zmysel táto funkcia nadobúda pri spolupráci viacerých autorov, ktorí podľa

komentáru, respektíve nastavenia malej zmeny, môžu následne vykonať určité akcie (kontrolu, vloženie vlastných zmien) alebo zmenu ignorovať (napríklad v prípade gramatických chyb).

6 Riešenie ukladania, porovnanie fyzickej reprezentácie

6.1 Vyhľadávanie

Kvalitné vyhľadávanie je dôležitou funkciou CMS, nevyhnutnou v prípade, že obsah v CMS nieje prístupný vyhľadávacím programom tretích strán (google, msn, yahoo). Vyhľadávacie funkcie musia riešiť dve úlohy, poskytnúť dostatočne silný nástroj pre definíciu hľadaného obsahu a súčasne vyhodnotiť relevanciu nájdeného obsahu. V rámci prvej úlohy je pomerne časté riešenie použitie vyhľadávacieho jazyka, spravidla ide o vyhľadávanie podľa slov, ktoré môžu alebo nemusia byť oddelené bielymi miestami, prípadne spojené logickými operátormi napríklad AND a NOT.

Doména vyhľadávania môže byť celý text (fulltext), v tomto prípade je okrem vyhľadávania konkrétnych slov, časté i rozšírenie o frázy (teda postupnosť slov, ktoré sa v texte musia vyskytovať v presnom poradí). Doménou môžu byť i metadáta, kategórie, tagy, kľúčové slová, možná je i kombinácia oboch prístupov.

Vyhodnotenie relevancie je obmedzené na syntaktické vlastnosti textu napríklad vzdialenosť hľadaných slov v texte, počet výskytov slov. Možné je i využiť logickú štruktúru textu, teda výskyt slova v nadpise priradí konkrétnemu obsahu vyššiu relevanciu ako iný výskyt.

Štandardne sa vyhľadávanie implementuje pomocou indexovania. Pri vyhľadávaní teda CMS nemusí vyhodnocovať každý obsah, ale využije pripravený index. Prehľad techník indexovania je nad rámec tejto práce, čitateľov odkazujeme na [28], [29]. Prakticky je možné indexovanie implementovať v CMS dvoma spôsobmi.

1. **Priamo**, CMS obsahuje modul ktorý, opakovane indexuje modifikovaný obsah. V tomto smere je populárny projekt Apache Lucene. Ide o JAVA implementáciu stratégie indexovania ako aj vyhľadávania. Existujú však derivované implementácie v mnohých ďalších jazykoch (C, C++, .NET, PHP, Python, Delphi/Pascal).
2. **Nepriamo**, CMS sa spolieha na indexovanie externým programom, najčastejšie sa využíva index poskytovaný relačnou databázou.

Výpis 14: Vyhľadávanie textu v relačnej databáze

```
1. SELECT menoStípca FROM menoTabulky WHERE menoStípca LIKE pattern
```

Pattern je reťazec znakov. Výsledkom sú riadky kde sa reťazec vyskytuje oddelený bielymi miestami. Znak '%' zastupuje množinu ľubovoľných znakov kým '_' práve jeden. Výsledkom môže byť teda i časť slova.

Tabuľka 50: Indexovanie, vyhľadávaci jazyk a relevancia nájdeného obsahu

CMS	Index	Jazyk	
Bricolage			
Drupal	Priamo	Logické operátory, kategórie, druh obsahu, frázy	- Relevantnosť je odvodená od logických konštrukcií HTML (tagy <h1>,<h2>), vzdialenosti slov, počtu komentárov ku konkrétnemu obsahu. - Obsah je indexovaný v definovaných časových intervaloch.
EzPublish	Priamo	Logické operátory, frázy, regulárne výrazy	- Pre každý element je možné nastaviť či má byť indexovaný. Okrem celého textu dovoľuje vyhľadávať podľa dátumu a typu komponentu. - Je možné zúžiť vyhľadávanie na podstrom statickej štruktúry.
Mambo	Nepriamo	Logické operátory, frázy	- index databázy
MediaWiki	Nepriamo	Logické operátory, Frázy	- Používa index databázy, optimalizované pre jednotlivé verzie MySQL a PostgreSQL, dotaz je predtým filtrovaný a upravený. - Podľa názvu alebo celého textu komponentu. - Je možné zúžiť podľa statickej štruktúry – namespace.
MoinMoin	Priamo	Logické operátory, frázy, regulárne výrazy,	- Relevantnosť je odvodená od logických konštrukcií: odkaz, názov stránky. Špeciálne operátory umožňujú vyhľadávať stránky, ktoré sa odkazujú na stránky s určeným názvom. Vyhľadávanie podľa celého textu alebo iba názvu stránky. -Experimentálna podpora Lupy (derivácia Lucene v jazyku Python).
Plone	Priamo (Zope)	Logické operátory, Frázy	- Obsah: podľa celého textu, popisu (úvodu), alebo mena. - Metadáta: autor, kľúčové slová, dátum, typ komponentu, aktivita vo workflow. - Zobrazuje percentuálnu relevantnosť obsahu.
TikiWiki	Nepriamo	Logické operátory	- index databázy
Typo3	Nepriamo	Logické operátory	- Vyhľadávanie podľa celého textu, alebo podľa kľúčových slov priradených komponentom (stránka musí byť typu <i>Advanced</i>).
Wordpress	Nepriamo	Frázy,	- Vyhľadávanie podľa názvu a textu obsahu. Dovoľuje iba vyhľadávanie podľa slov (všetky musia byť prítomne) alebo frázy (všetky slová v exaktnom poradí).

6.2 Fyzická reprezentácia

Pod fyzickou reprezentáciou rozumieme formát elementárnych obsahov ako i spôsob uloženia. V zásade existujú dve možnosti, využiť relačnú databázu, alebo použiť súborový systém hosťiteľského operačného súboru.

Tabuľka 51: Implementácia CMS

CMS	Uloženie	Prostredie	Poznámky
Bricolage	PostgreSQL	mod_perl (Apache 1.x)	- Používa mnoho rozširujúcich perl modulov.
Drupal	MySQL, PostgreSQL	PHP	- Problémom sú niektoré moduly. Často totiž neobsahujú špecifikáciu potrebných databázových tabuliek pre oba varianty (hlavne rozširujúce moduly), alebo využívajú metódy dostupné len v jednej variante (v neprospech PostgreSQL). - Interne používa vlastné SQL API, ktoré abstrahuje od vykonávania zložitejších SQL operácií (napríklad INNER JOIN). - Použitie niekoľkých databáz súčasne. (Aj kombinácia MySQL a PostgreSQL je možná.) - Odporúčame použitie MySQL 4.0 alebo 4.1. V prieskume [35] 87% užívateľov používa MySQL.
EzPublish	MySQL	PHP (4.x)	- Použitie niekoľkých databáz súčasne
Mambo	MySQL	PHP	- Vo vývoji sa nachádza modul, založený na ADODB vďaka, ktorému by bolo možné použiť i iné relačné databázy (alfa verzia).
Magnolia	Java content repository	JSP	- Dodáva sa s Apache JackRabbit JCR.
MediaWiki	MySQL, PostgreSQL	PHP (5.x)	- Je možné použiť súčasne niekoľko databáz.
MoinMoin	Súborový systém	Python	- Wiki-stránky zodpovedajú súborovým zložkám, pričom jednotlivé verzie konkrétnej wiki-stránky sú uložené ako súbory v tejto zložke (meno súboru je verzia).
Plone	Zope objektová databáza	Zope/Python	- Pri definícii komponentu, môžeme zvoliť aj inú formu uloženia, napríklad relačnú databázu.
TikiWiki	ADODB	PHP	
Typo3	MySQL ADODB	PHP	- Použitie niekoľkých databáz súčasne. - Optimalizovaný prístup k MySQL, prístup k iným relačným databázam cez ADODB.
Wordpress	MySQL	PHP	

Väčšina z prezentovaných CMS sú označované ako takzvané LAMP riešenia, Linux operačný systém, Apache web-server, MySQL databáza a PHP zdrojový jazyk aplikácie. Výhodou CMS založených na PHP je ich prenositeľnosť, keďže modul pre PHP existuje pre i pre ďalšie serveri (Lighttpd, IIS).

ADODB je knižnica napísaná v PHP, ktorá predstavuje abstraktnú databázovú vrstvu. Vďaka tomu môže CMS využiť, ktorúkoľvek relačnú databázu, podporovanú ADODB. Samozrejme každá úroveň abstrakcie, predstavuje určitý pokles výkonu. Možným riešením je kritické časti prepísať do kompilovaného jazyka. Práve toto riešenie autori ADODB využili, časť knižnice je možné nahradiť skompilovaným modulom. Porovnanie výkonu a podporované databázy nájde čitateľ v [30].

Plone je založená na aplikačnom serveri Zope. Zope nie je možné priamo porovnať so serverom ako Apache alebo Tomcat. Jedným z komponentov Zope je CMF – content management framework. Plone je len pomerne tenkou aplikáciou nad bohatým rozhraním Zope. Prakticky je možné použiť Zope ako CMS samostatne a existujú i ďalšie CMS založené na Zope, napríklad Silva.

Magnolia využíva na ukladanie špeciálny druh databázy nazývaný Java Content Repository. Ide o pripravovaný štandard pre Java, z praktického hľadiska je JCR druhom objektovej databázy. Existuje niekoľko rôznych poskytovateľov implementácie, vďaka čomu je možné podľa potrieb, pomerne jednoducho zvoliť inú implementáciu.

Cache a nadmerné zaťaženie

Prakticky všetky CMS poskytujú techniku *cache*, vygenerované webové stránky sa na určitú dobu uložia a v prípade požiadavky na túto stránku je odpoveďou uložená stránka. Tým sa znižuje celkové zaťaženie CMS.

Táto technika však nemôže úplne vylúčiť možnosť nadmerného zaťaženia CMS. Túto situáciu musia CMS zaznamenať a riešiť. Nadmerné zaťaženie nastáva keď CMS, (respektíve určitý komponent ako webový server) je nútený obsluhovať viac klientov ako dovoľujú technické prostriedky (napríklad voľná pamäť, šírka sieťového pásma). Výsledkom môže byť dočasná nedostupnosť pre niektorých klientov, v horšom prípade môže byť CMS úplne nedostupné. Túto situáciu označujeme ako odmietnutie obsluhy - (*Denial Of Service*, DOS). Najčastejšie tento problém vznikne ak sa určitý populárny webový portál odkzuje na webovú stránku CMS (relatívne zaužívaným pojmom pre tento scenár je *slashdotting*, podľa populárneho webu slashdot.org). Na druhej strane môže ísť aj o druh útoku, takzvaný *DOS* útok, ktorý nie je obmedzený len na CMS.

Tabuľka 52: Porovnanie riešení cache a nadmerného zaťaženia

CMS	Cache	Nadmerné zaťaženie
Drupal	Áno	- Administrátor môže zvoliť funkcie CMS, ktoré v prípade zvýšenej záťaže nebudú dostupné, zvlášť pre autentifikovaných a neautentifikovaných užívateľov. Štandardne dodávaný modul <i>throttle</i> .
Mambo	Áno	Nie
MoinMoin	Áno	Nie
TikiWiki	Áno	Áno - V prípade detekcie nadmerného zaťaženia je možné zobrazit' chybové hlásenie. Explicitne povolenie prístupu v takomto prípade je možné priradením špeciálneho práva. - Detekcia nadmerného zaťaženia je odvodená zo zaťaženia operačného systému a je konfigurovateľná.
Typo3	Áno	Nie

7 Záverečný prehľad

7.1 Rozdelenie CMS

V tejto časti sa pozrieme na logické skupiny CMS podľa zamerania a využitia. Uvádzame nutné funkcie pre každú skupinu a najvhodnejších kandidátov pre danú skupinu.

7.1.1 CMS ako nástroj pre vytvorenie webového sídla

Táto skupina CMS patrí medzi najväčšie. Pôvodne boli webové sídla vytvárané ako množina statických HTML stránok, avšak relatívne nízka cena CMS, jednoduchosť používania pre netechnických užívateľov a zabezpečenie konzistentného dizajnu spôsobili, že čoraz viac spoločností CMS využíva.

Požiadavky:

- Jednoduchosť, teda WYSIWYG nástroje, komponentom obsahu je webová stránka, najčastejšie stačí jednoduché workflow.
- Striktná statická štruktúra, umožňuje užívateľom orientáciu v obsahu, pridelenie zodpovednosti za určité časti štruktúry, ako aj automatické vytvorenie navigačných elementov.
- Flexibilný HTML výstup. CMS musí umožňovať vysokú mieru flexibility pri dizajne webových stránok.
- Internacionalizácia, keďže čoraz viac spoločností operuje medzinárodne.

Kandidáti:

Typo3, EzPublish, Magnolia, Mambo, spĺňajú všetky predchádzajúce požiadavky, navyše poskytujú mnohé ďalšie funkcie. Kým prvé tri CMS sú využiteľné pre prakticky neobmedzene veľké webové sídla, vzhľadom na jednoduchosť Mambo odporúčame použiť iba pre stredné a malé.

Osobné webové sídla

Špeciálnou podskupinou sú osobné webové sídla. Najčastejšie využívané úzkou skupinou producentov obsahu, často však s pomerne veľkou skupinou konzumentov. To tejto kategórie patria napríklad *blogy*.

Požiadavky:

- Dynamická štruktúra, teda orientácia v čase, kategórie a tagy.
- Jednoduchá konfigurácia prezentácie, najčastejšie ako rozširujúci modul.

Kandidáti:

Wordpress, Drupal, čiastočne TikiWiki a Plone (s rozširujúcim modulom).

7.1.2 CMS ako online periodiká

Najčastejšie sa s CMS s tejto skupiny môžeme stretnúť ako s online verziami klasických periodik. Základnou požiadavkou pre CMS je schopnosť pracovať s veľkou množinou obsahu.

Požiadavky:

- Silné nástroje na definíciu typu komponentu a jeho vnútornej štruktúry.
- Striktné oddelenie prezentácie a štruktúry.
- Bohaté metadáta a možnosti pre definíciu niekoľkých vonkajších štruktúru ako tagov a kategórií.
- Užívateľské rozhranie, orientované na úlohy. Na rozdiel od predchádzajúcej skupiny nemusí byť jednoduché, avšak rýchly prístup k funkciám je kritický.
- Prepracovaný systém workflow.
- CMS musí zabezpečiť možnosť publikácie do viacerých formátov.
- Aspoň čiastočná internacionalizácia je výhodou.

Kandidáti:

Bricolage, čiastočne Plone, Drupal. Keďže Bricolage bol pôvodne vyvíjaný práve online periodikom tento výber je jasný. Plone a Drupal poskytujú tak pomerne silné workflow ako aj definície typu komponentov. Pre rozsiahlu množinu obsahu odporúčame Bricolage, pre jednoduchšie periodiká Plone alebo Drupal.

7.1.3 CMS ako kolaboračná platforma

V rámci tejto skupiny sme identifikovali dve podskupiny.

- Prvá je orientovaná na kolaboráciu v rámci preddefinovaných skupín užívateľov.
- Druhá je zameraná na veľkú a rôznorodú skupinu nezávislých užívateľov.

Dôraz je na efektívnu komunikáciu, v oboch prípadoch konkurujú CMS hlavne e-mailu (a mailinglistom).

Požiadavky:

- Integrácia s e-mailom a/alebo poskytovanie komunikačného rozhrania napríklad správ v rámci CMS a hlavne diskusných fór.
- Podpora notifikácií o zmene je kritickou funkciou.
- Kvalitné vyhľadávanie.
- Jednoduché vytváranie obsahu, hlavným kandidátom sú wiki jazyky.
- V prvej skupine aj integrácia s dokumentami štandardných kancelárskych aplikácií.
- Veľkou výhodou je prístup k CMS cez súborovo-orientovaný protokol, nakoľko integruje CMS s užívateľským štandardným prostredím.
- Kritické sú i funkcie pre správu verzií a zabránenie alebo riešenie kolízií pri modifikáciách obsahu.
- Možnosť definície typov komponentov dovoľuje spravovať štruktúrovaný obsah (napríklad chyby, požiadavky na zmeny, často kladené otázky) a je tiež výhodou.
- V prípade, že chceme použiť CMS v prvej skupine, pre niekoľko nezávislých skupín, potrebujeme striktné statické štruktúry a kvalitnú kontrol prístupu.
- Kalendár a komponenty typu udalosť, dovoľujú plánovať kolaboráciu a sú tiež potrebnou funkciou.

Kandidáti:

V prvej skupine najmä Plone a čiastočne Drupal. V druhej skupine sú Drupal a všetky wiki CMS v našom prehľade. Príkladom použitia sú napríklad fóra pre podporu (napríklad komerčných produktov). V tomto prípade je dôležité zabezpečiť tok informácií zachytených v diskusiách do formálnejších komponentov obsahu. Príklad tohto toku je možno vidieť v CMS pre opensource software, napríklad veľká časť wiki stránok gentoo linux začína odkazom na originálnu diskusiu. Domnievame sa, že ide o kritickú úlohu. V tomto smere by sme radi videli funkcie ako „wikifi“, ktorá by na jeden krok skopírovala obsah diskusie do wiki stránky.

7.2 Subjektívne hodnotenie

Obsahom tejto časti sú naše subjektívne dojmy z inštalácie a používania CMS. Pri inštaláciách sú možnosti:

- Samostatná aplikácia, spustiteľná na hostiteľskom operačnom systéme.
- Webová aplikácia. Prvým krokom je skopírovanie inštalačného adresára do vhodného adresára webového/aplikačného servera. Druhým krokom je použitie webového prehliadača, pričom CMS poskytuje webovú aplikáciu pre počiatočnú konfiguráciu. Jediným externým krokom je najčastejšie vytvorenie databázy.

Domnievame sa, že obe možnosti sú ekvivalentné, uvádzame ich pre úplnosť.

Pri analýze používania sme sa zamerali na jednoduchosť používania ako aj jednoduchosť objavovania funkcií. Súčasne sme hodnotili dostupnosť konfiguračných nástrojov priamo v užívateľskom rozhraní CMS.

Výsledky sumarizujeme v tabuľke 53.

Tabuľka 53: Subjektívne hodnotenie

CMS	Inštalácia	Používanie
Bricolage	<ul style="list-style-type: none"> - Pomerne komplikovaná, potrebuje veľa dodatočných perl modulov. - Navyše obmedzená kompatibilita iba s Apache 1.x. - Po troch pokusoch využiť inštaláciu z balíčkov rôznych linuxových distribúcií, sme nakoniec inštalovali manuálne zo zdrojových súborov. 	<ul style="list-style-type: none"> - Prehľadné a príjemne rozhranie. - Niektoré nastavenia sú prístupné len z konfiguračného súboru.
Drupal	<ul style="list-style-type: none"> - Webová 	<ul style="list-style-type: none"> - Prehľadné a príjemne rozhranie. - Logická navigácia a predvídateľné umiestnenie funkcií.
EzPublish	<ul style="list-style-type: none"> - Webová, kontroluje dostupnosť potrebných modulov a navrhuje zlepšenia konfigurácie (PHP). 	<ul style="list-style-type: none"> - Menej prehľadné, často je dodatočná konfigurácia prístupná len cez modifikáciu konfiguračných súborov. - Problém predstavuje využívanie kontextového menu (ECMAScript).
Mambo	<ul style="list-style-type: none"> - Webová, upozorňuje na nebezpečné nastavenia 	<ul style="list-style-type: none"> - Prehľadné a príjemne rozhranie. - Nelogické je rozdelenie rozširujúcich modulov na viac typov, a previazanie logickej publikácie s prvkom menu.
Magnolia	<ul style="list-style-type: none"> - Jednoduchá inštalácia, skopírovaním pripravených súborov. - Inštaláčne programy. 	<ul style="list-style-type: none"> - Prehľadné a príjemne rozhranie. - Problém predstavuje využívanie kontextového menu (ECMAScript).
MediaWiki	<ul style="list-style-type: none"> - Webová 	<ul style="list-style-type: none"> - Málo nástrojov na konfiguráciu.
MoinMoin	<ul style="list-style-type: none"> - 	<ul style="list-style-type: none"> - Jednoducho prístupná dokumentácia, je súčasťou inštalácie. - Málo nástrojov na konfiguráciu.
Plone	<ul style="list-style-type: none"> - Poskytuje inštaláčne programy pre väčšinu platforiem. 	<ul style="list-style-type: none"> - Prehľadné a príjemne rozhranie. - Málo nástrojov na konfiguráciu, mnohé pokročilé nastavenia sú dostupné len v ZMI.
TikiWiki	<ul style="list-style-type: none"> - Webová 	<ul style="list-style-type: none"> - Extrémne neprehľadné, veľa času stráveného opakovaným hľadaním nastavení.
Typo3	<ul style="list-style-type: none"> - Webová 	<ul style="list-style-type: none"> - Menej prehľadné ale príjemne rozhranie. - Zobrazuje pomerne veľa informácií a akcií súčasne. - Problém predstavuje využívanie kontextového menu (ECMAScript). - Prístup k nízkej úrovni CMS napríklad k tabuľkám databázy. - Integrované nástroje na sťahovanie a inštaláciu rozširujúcich modulov.
Wordpress	<ul style="list-style-type: none"> - Webová 	<ul style="list-style-type: none"> - Prehľadné, príjemne a úlohovo-orientované rozhranie.

8 Záver

Problematika CMS je značne rozsiahla. Za hlavný prínos tejto práce považujeme podanie prehľadu o problematike. Prínosom práce je taktiež zmapovanie situácie v oblasti opensource CMS, ako aj analýza technológií používaných v rámci CMS.

CMS sa stávajú dôležitou až nevyhnutnou súčasťou organizácií. Prvým krokom pri nasadzovaní CMS je výber správneho systému, ktorý je komplikovaný veľkým množstvom kritérií. Prínosom práce je aj identifikácia a rozdelenie kľúčových požiadaviek.

Pre ďalšie smerovanie uvádzame nasledovných niekoľko bodov

- v práci sa venujeme prevažne textovému vstupu. Zaujímavé by bolo podať prehľad o možnostiach CMS zameraných na správu multimediálneho obsahu. Zaujímavými kandidátmi sú v tomto prípade projekty ako *Flickr*, *YouTube*, ap.
- námetom na implementačnú prácu by bolo automatické vytváranie typov komponentov ako aj ich rozhrania pre vkladanie obsahu.
- v práci okrajovo spomíname možnosti automatickej klasifikácie obsahu. Domnievame sa, že preskúmanie tejto problematiky by bolo vhodnou témou ďalšej práce.

9 Literatúra

- [1] Halm M.; Pelikan M.: Enterprise Content Management Systems: Beyond Digital Asset Management and Web Content Management Systems;
http://its.psu.edu/dmr/notes/ECM_finish-c.pdf
- [2] Wheeler D.: Secure Programmer: Validating input, 2003;
<http://www-128.ibm.com/developerworks/library/l-sp2.html>
- [3] CERT Coordination Center: Understanding Malicious Content Mitigation for Web Developers;
http://www.cert.org/tech_tips/malicious_code_mitigation.html
- [4] Cowan C.: Software security for open-source systems; IEEE J. Security & Privacy, 1(1):38—45, 2003;
http://wirex.com/~crispin/opensource_security_survey.pdf
- [5] Tiki Working Group Franck M.: INTERNET-DRAFT The Wiki Community, September 2004;
<http://tikiwiki.org/RFCWiki>
- [6] Snow M.: Wiki syntax standardization project proposed, 2004;
http://en.wikipedia.org/wiki/Wikipedia:Wikipedia_Signpost/2005-04-04/Syntax_standardization
- [7] Reitmayr E.: Editing Information in the German Wikipedia, 2006;
openusability.org/reports/get_file.php?group_id=109&repid=69
- [8] DOM Working group: Document Object Model;
<http://www.w3.org/DOM/>
- [9] Browning P.: Web Editors;
<http://www.geniisoft.com/showcase.nsf/WebEditors>
- [10] <http://www.mozilla.org/editor/midasdemo>
- [11] Stenback J.; Heninger A.: Document Object Model (DOM) Level 3 Load and Save Specification, April 2004;
<http://www.w3.org/TR/2004/REC-DOM-Level-3-LS-20040407/>
- [12] Viégas F.; Wattenberg M.; Dave K.: Studying Cooperation and Conflict between Authors with history flow Visualizations;
<http://opensource.mit.edu/papers/viegaswattenbergdave.pdf>
- [13] Goland Y.; Whitehead E. Irvine UC.; Faizi A.; Carter S.; Jensen D.: HTTP Extensions for Distributed Authoring – WEBDAV, Február 1999;
<http://www.ietf.org/rfc/rfc2518.txt>
- [14] Kline G.; Carrol J.; McBride B.: Resource Description Framework (RDF): Concepts and Abstract;
<http://www.w3.org/TR/rdf-concepts/>
- [15] King A.: The evolution of RSS;
<http://www.webreference.com/authoring/languages/xml/rss/1/index.html>
- [16] Nielsen H.; LaLiberte D.: Detecting the Lost Update Problem Using Unreserved Checkout;
<http://www.w3.org/1999/04/Editing/>
- [17] Chekuri Ch.; Goldwasser M.; Raghavan P.; Upfal E.: Web Search Using Automatic Classification, 1997;
http://theory.stanford.edu/people/wass/publications/Web_Search/
- [18] Langridge S.; Hickson I.: Pingback 1.0;
<http://hixie.ch/specs/pingback/pingback>

- [19] Pronet.: Developer Documentation : TrackBack Technical Specification;
http://www.sixapart.com/pronet/docs/trackback_spec#Sending_a_TrackBack_Ping
- [20] Regli T.; Donner C.: Placed vs. Dynamic Content, 2005;
<http://www.cmswatch.com/Feature/129-Web-Publishing>
- [21] Grossnickle J.; Board T.; Pickens B.; Bellmont M.: RSS-Crossing into the Mainstream, Október 2005;
publisher.yahoo.com/rss/RSS_whitePaper1004.pdf
- [22] Powell A.; Nilsson M.; Naeve A.; Johnston P.: DCMI Abstract Model, 2005;
<http://dublincore.org/documents/abstract-model/>
- [23] Dave, P., Bogen, P., Karadkar, U.P., Francisco-Revilla, L., Furuta, R., Shipman, F.: Dynamically Growing Hypertext Collections. Submitted to Proceedings of Hypertext
<http://www.csdl.tamu.edu/walden/reports/ht2004.pdf>
- [24] Muehlen M.; Allen R.: Stand-alone and embedded workflow management systems; 2000;
http://www.wfmc.org/standards/docs/MzM_RA_WfMC_WP_Embedded_and_Autonomous_Workflow.pdf
- [25] Kolektív autorov: Process Definition Inteface – XML Process Definition Language, Október 2005;
http://www.wfmc.org/standards/docs/TC-1025_xpdl_2_2005-10-03.pdf
- [26] Seigenthaler J.: A false Wikipedia 'biography', November 2005;
http://www.usatoday.com/news/opinion/editorials/2005-11-29-wikipedia-edit_x.htm
- [27] http://en.wikipedia.org/wiki/Wikipedia:Requests_for_comment/United_States_Congress
- [28] Elmasri R.; Navathe S.: Fundamentals of Database Systems, Addison-Wesley, 1994;
- [29] Brin S.; Page L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine;
<http://www-db.stanford.edu/pub/papers/google.pdf>
- [30] Lim J.: ADOdb Library for PHP, 2006;
<http://phplens.com/lens/adodb/docs-adodb.htm>
- [31] Eric van der Vlist; XML.com; May Building a Semantic Web Site;
<http://www.xml.com/pub/a/2001/05/02/semanticwebsite.html>
- [32] RSS-DEV Working Group: RDF Site Summary (RSS) 1.0, 2000;
<http://web.resource.org/rss/1.0/spec>
- [33] Kolektív autorov: Java Content Repository Standard JSR-170;
<http://jcp.org/aboutJava/communityprocess/maintenance/jsr170/index.html>
- [34] <http://pear.php.net/package-info.php?pacid=179>
- [35] <http://drupal.org/node/37268>
- [36] GNU General Public License
<http://www.gnu.org/copyleft/gpl.html>
- [37] GNU Lesser General Public License
<http://www.gnu.org/copyleft/lgpl.html>
- [38] Rozšírenie komponentov obsahu o kardinalitu a štruktúru v Plone.
<http://plone.org/products/compoundfield>
- [39] <http://plone.org/documentation/tutorial/archgenxml-getting-started>
- [40] McKay A.: The Definitive Guide to Plone, Apress, 2004
- [41] Morville P., Rosenfeld L.: Information Architecture for the World Wide Web, O'Reilly Media, 2002
- [42] Powers S.: Practical RDF, O'Reilly Media, 2003

- [43] Boiko B.: Content Management Bible 2. edícia, Wiley Publishing, 2005
- [44] Ray E.: Learning XML, O'Reilly Media, 2001
- [45] Kiepuszewski B.: Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows;
- [46] Eisenberg D.: OASIS OpenDocument Essentials;
<http://books.evc-cit.info/index.html>
- [47] Kolektív autorov: RDF Primer, 2004;
<http://www.w3.org/TR/rdf-primer/>
- [48] Kohsuke Kawaguchi: W3C XML Schema Made Simple, 2001;
<http://www.xml.com/pub/a/2001/06/06/schemasimple.html>
- [49] Walsh N., Muellner L.: DocBook: The Definitive Guide, O'Reilly, 2006;
- [50] Kolektív autorov: Delta encoding in HTTP;
<http://www.ietf.org/rfc/rfc3229.txt>
- [51] Garvey P., Gratacos M., Leon J., Smith C.: The Berkeley XML Application Infrastructure, 2003;
<http://groups.sims.berkeley.edu/CDE/report/go-five-final-report.html>
- [52] Kolektív autorov: Použitie a prepojenie viacerých inštancií MediaWiki;
http://www.mediawiki.org/wiki/Manual:Wiki_family
- [53] Barkley J.: Comparing Simple Role Based Access Control Models and Access Control Lists, NIST; 1997;
- [54] Skaarhoj K.: Inside Typo3; 2007;
http://typo3.org/documentation/document-library/core-documentation/doc_core_inside/
- [55] Ritz R.: Locking in Plone; 2006;
<http://plone.org/products/plone/roadmap/145>
- [56] Kolektív autorov: GNU gettext utilities;
<http://www.gnu.org/software/gettext/manual/gettext.html>

Zoznam pojmov

ADODB.....	108	Obsah.....	10
ATOM.....	58	Odmietnutie obsluhy.....	108
Autentifikácia.....	91	OWL.....	41
Automatické objavenie.....	80	Panel nástrojov.....	43
Blog.....	15	Permalink.....	41
Blokový element.....	71	Pingback.....	63
Business process.....	94	Pod-stránky.....	35
Cache.....	108	Podcast.....	81
Centrálne úložisko.....	51	Príloha.....	51
CMS.....	10	Problém stratenej aktualizácie.....	101
Checkin.....	101	Producent – Konzument.....	14
Checkout.....	101	RBAC.....	84
DAC.....	83	RDF.....	24
DAM.....	12	RSS.....	58
DHTML.....	43	Slashdoting.....	108
Diskusia.....	15	SPAM.....	99
Diskusné fórum.....	15	Structure.....	35
DMS.....	12	Tag.....	38
DocBook.....	56	Tag cloud.....	39
Dokument v dokumente.....	49	Trackback.....	63
DOM.....	43	Vyhľadávací jazyk.....	105
DOS.....	108	WCMS.....	13
DRM.....	90	Web-feed.....	58
EMS.....	13	Weblog.....	15
execCommand.....	45	Wiki.....	15
Indexovanie.....	105	Wiki jazyk.....	27
Integrovaná správa licencií.....	90	Workflow management system.....	94
Kniha.....	35	Workflow.....	94
Kolaboračné.....	14	Workflow engine.....	94
MAC.....	84	Workflow-based.....	94
Malá zmena.....	103	Workflow-enabled.....	94
Nadmerné zaťaženie.....	108		