

**Academia de Științe a Republicii Moldova
Institutul de Matematică și Informatică**

Cu titlu de manuscris
C.Z.U.: 004:65.012.4

Oleg Burlaca

Sistem de management al conținutului pentru Web

Specialitatea: 01.05.04 - Modelare matematică, metode matematice, produse program

Teză de doctor în informatică

Conducător științific:
Constantin Gaidric, dr. h. prof. univ.

Consultant științific:
Svetlana Cojocaru, dr.

Chișinău
2006

Cuprins

Introducere

Capitolul 1. Sisteme de management al conținutului 5

- 1.1. Preliminarii 6
- 1.2. Managementul conținutului 7
- 1.3. Ciclul de viață al managementului conținutului 8
- 1.4. Clasificarea CMS 10
- 1.5. CMS: priorități și lacune 14
- 1.6. Căi de dezvoltare CMS 17
- 1.7. Formularea problemei 18

Capitolul 2. Un model de sistem de management al conținutului

- 2.1. Introducere 20
- 2.2. Etapele de elaborare a site-urilor web 20
- 2.3. Modelul paginii de conținut al site-ului structurat 21
- 2.4. Model relațional-ierarhic de gestiune a datelor 24
- 2.5. Modelul hipertext al site-ului 29
- 2.6. Arhitectura sistemului 29
- 2.7. Aplicația client al sistemului 31
- 2.8. Procesul de generare a site-ului 35
- 2.9. Sistemul de management al conținutului NeoSite 39
- 2.10. Concluzii 41

Capitolul 3. Aplicațiile sistemului NeoSite 43

- 3.1. Domeniul de aplicabilitate 43
- 3.2. Portofolio 43
- 3.3. Studiu de caz: site-ul CNAA 44
 - 3.3.1. Site-urile anterioare ale CSA/CNAA 45
 - 3.3.1. Teze digitale 45
 - 3.3.3. Analiza și structurarea informației 47
 - 3.3.4. Elaborarea modelului de date și a structurii site-ului. Implementare 49
 - 3.3.5. Sistemul de căutare 53
 - 3.3.6. Statistica site-ului. Evaluare 54
- 3.4. Concluzii 55

Concluzii și recomandări 57

Bibliografia 59

Introducere

Actualitatea temei:

Societatea informațională este societatea în care producerea și consumul de informație este cel mai important tip de activitate, informația este recunoscută drept resursă principală, tehnologiile informației și comunicațiilor sînt tehnologii de bază, iar mediul informațional, împreună cu cel social și cel ecologic – un mediu de existență a omului.

Societatea informațională reprezintă o nouă etapă a civilizației umane, un nou mod de viață calitativ superior care implică folosirea intensă a informației în toate sferele activității și existenței umane. Societatea informațională permite accesul larg la informație al membrilor săi, un nou mod de lucru și de cunoaștere.

Noile tehnologii digitale reduc esențial costul de acces, stocare și transmitere a informației. Sistemele informatice joacă un rol important în procesele de accelerare a schimbului de informații între organizații și diseminarea eficientă a informației către consumator sau, dacă vorbim în contextul e-guvernării, către cetățean.

În secolul XX, cel mai mare eveniment tehnologic și social în același timp a fost apariția Internetului. Drăgănescu [1] afirmă că „în esență societatea informațională este societatea care se bazează pe Internet”. Web-ul a devenit un nou mediu de publicare a informației. Site-urile web sînt folosite pentru a promova companiile și produsele sale, presta servicii și informații, facilita comunicarea. Pentru site-uri medii și mari, se pune problema managementului conținutului lor. De aceea s-au implementat sisteme de management al conținutului (*engl., Content Management System, prescurtat: CMS*). Intrările sistemului CMS sînt informațiile ce trebuie publicate, ieșirile sînt paginile HTML care vor fi văzute de vizitatori.

Managementul conținutului este procesul de aprovizionare sistematică și structurată, de creare, prelucrare, administrare, prezentare, procesare, publicare și reutilizare a conținutului [2].

Există o gamă largă de CMS-uri [3], dar nici o soluție nu este în stare să gestioneze toate tipurile de conținut. Noțiunea de “cel mai bun” nu există. Rezultatele chestionarului [4] au stabilit că problemele majore întâlnite la proiectarea sau implementarea sistemelor de management al conținutului sînt:

- softul comercial este prea costisitor și necesită mult timp pentru a fi implementat și ajustat;
- dificultăți la integrare cu alte sisteme;
- determinarea cerințelor față de sistem;
- integrarea conținutului existent;
- instruirea autorilor și a editorilor.

Mai mult ca atît, a fost stabilit [5] că 61% din companiile ce utilizează sisteme de management a conținutului se bazează pe actualizarea manuală a site-urilor lor. Utilizatorii solicită: [4]

- sisteme compacte și simple;
- documentație cu idei concrete de organizare a proiectului și utilizare la maximum a posibilităților sistemului;
- interfețe adaptabile conform necesităților fiecărui utilizator în parte;
- interfețe comode de creare și actualizare a conținutului.

Trecerea la societatea informațională implică sporirea volumului de informații și accelerarea comunicării, problema managementului eficient al conținutului devine tot mai importantă. Din simple cărți de vizită, site-urile web au devenit astăzi proiecția activității organizațiilor în spațiul Web. Este necesară elaborarea unor modele de CMS care ar soluționa problemele menționate și ar accelera procesele de creare și mentenanță a site-urilor Web.

Scopul și obiectivele tezei:

Elaborarea și implementarea unui model de sistem de management al conținutului pentru site-uri complexe și bine structurate care ar permite dezvoltarea și mentenanța site-urilor Web cu resurse minime.

Obiectivele tezei au constituit:

1. Studiul CMS-urilor destinate site-urilor complexe, al arhitecturii lor și depistarea deficiențelor acestora care vor fi luate în considerație la elaborarea unui model nou de CMS;
2. Elaborarea unui model de date flexibil în baza cerințelor și specificului impuse de aplicațiile Web;
3. Elaborarea unei interfețe generale de management al conținutului pentru modelul de date propus;
4. Elaborarea unei arhitecturi scalabile CMS care ar garanta evoluția soluției CMS implementate (de exemplu, adăugarea de funcționalități noi, capacitatea de integrare);
5. Elaborarea unui model de site Web care ar reflecta, cu cheluieli minime de programare, informația din modelul de date;
6. Implementarea practică a modelelor propuse într-un CMS;
7. Elaborarea de site-uri web cu ajutorul CMS-ului propus.

Noutatea științifică a rezultatelor:

Noutatea științifică o constituie abordările de elaborare a CMS-urilor și metodelor de proiectare a site-urilor web:

- Modelul de date relațional-ierarhic (descriș în §2.4) în care relațiile sunt reprezentate sub formă de arbore. Relațiile de grad N sînt reprezentate prin $N-1$ relații binare. Adăugarea de noi tipuri de conținut și stabilirea de noi tipuri de relații între noduri nu necesită schimbări esențiale în structura bazei de date. Acest fapt asigură flexibilitatea modelului;
- Elaborarea în paralel al modelului date și modelului hipertext al site-ului a redus considerabil efortul pentru stabilirea corespondenței dintre adresa URL (*engl. Uniform Resource Locator*) a paginii accesate și nodului respectiv din modelul de date. În același timp, aceasta facilitează extragerea eficientă a informației din baza de date pentru generarea paginilor;
- Modelul de pagină web care se bazează pe relațiile dintre nodurile de conținut din care este generată o pagină poate fi folosit la proiectarea site-urilor complexe și asigură legătura cu alte compartimente ale site-ului. Unul din mecanismele principale ale acestui model este metoda de filtrare contextuală pentru interconectarea paginilor și implementarea căilor alternative (indirecte) de navigare pe site. Aplicarea acestei metode accentuează legăturile paginii cu alte compartimente ale site-ului și contribuie la conștientizarea poziției conținutului paginii în spațiul informațional al site-ului. Ca urmare, devine mai clară structura site-ului și se reduce timpul necesar vizitatorului pentru adaptare.

Semnificația teoretică și aplicativă:

Semnificația teoretică constă în introducerea modelului de date compus din două structuri ierarhice: a conținutului și a relațiilor. La nivel aplicativ, modelul poate fi folosit la elaborarea sistemelor de management al datelor structurate. Principiul care stă la baza interfeței de gestiune a datelor poate fi folosit la elaborarea unor interfețe comode pentru editoarele de documente XML. Un alt aspect teoretic este structura abstractă a paginii web și procedeul de filtrare contextuală.

Principiile și procesul de creare a site-urilor descrise în teză vor fi utile la proiectarea site-urilor complexe ce conțin informație structurată. Procedul de filtrare contextuală propus poate fi aplicat pentru orice site. Cu ajutorul CMS-ului elaborat au fost create mai multe site-uri complexe (descrise în §3.2) ce sînt întreținute de persoane slab inițiate în domeniul tehnologiilor informaționale din cadrul organizațiilor respective. CMS-ul elaborat poate fi propus studiourilor Web ca platformă de creare a site-urilor. Arhitectura CMS-ului propus poate fi folosită la elaborarea sistemelor complexe flexibile și scalabile.

Aprobarea rezultatelor:

Rezultatele expuse în teză au fost publicate în [6, 7] și comunicate în cadrul conferințelor:

- BiT+ International Conference on Information Technologies, anii 2001 – 2005 (Chișinău, Moldova);
- SACCS 2004 - International Symposium on Automatic Control and Computer Science, (22 - 23 octombrie 2004, Iași, România);
- IV International Conference on Microelectronics and Computer Science (septembrie 15-17, 2005, UTM, Chișinău, Moldova);
- seminarele Institutului de Matematică și Informatică al A.Ș.M.

Conform modelului de CMS propus în teză a fost implementat un CMS numit NeoSite cu ajutorul căruia au fost elaborate 15 site-uri care au luat premii de gradul I și II în cadrul concursului WebTop - concurs național al celor mai semnificative Web site-uri create în Moldova și pentru Moldova.

Capitolul 1. Sisteme de management al conținutului

1.1. Preliminarii

Definiții

Există mai multe definiții ale termenului “gestiunea conținutului”, deseori ele reprezintă puncte de vedere ale unor analiști influențați de politica de marketing al organizațiilor în care activează. În [8] găsim definițiile companiilor Gartner, Forrester, Giga, Meta Group, CAP Ventures. Meta Group definește “gestiunea conținutului” astfel: “... un amalgam complex de funcționalitate, ce include achiziția, administrarea, asamblarea, recenzarea și aprobarea, publicarea, păstrarea și protecția informației destinate necesităților internet, intranet sau extranet ale unei organizații”. Însă acestea sînt definiții specializate ce nu vor fi în stare să cuprindă sistemele care vor apărea în viitor. În plus, notiunea de CMS se intersectează cu cea a sistemelor de gestiune a documentelor, sistemelor de gestiune a cunoștințelor, sistemelor de integrare a aplicațiilor industriale, sisteme e-commerce și portaluri.

Definiții mai generale ale managementului conținutului:

“Un set de sarcini sau procese pentru administrarea conținutului de la apariție pînă la arhivare” (Ovum, <http://www.ovum.com/>).

“Un domeniu nou preocupat de facilitarea administrării și accesării informației în cadrul mediilor corporative” [9].

În realitate, CMS este mai mult un concept decît un produs sau tehnologie. Un concept ce cuprinde o serie de procese care vor sta la temelia viitoarei generații de site-uri web (de dimensiuni mari) în care conținutul va fi gestionat de autorii acestuia ci nu de tehnicieni. Scopul principal al CMS este integrarea și automatizarea sporită a proceselor ce contribuie la diseminarea eficientă și efectivă a informației în Internet.

Istoria sistemelor de management al conținutului

Istoria CMS începe odată cu inventarea scrisului, crearea unor formate portabile de documente bazate pe hîrtie (papyrus), asamblarea mai multor pagini în cărți, adunarea cărților în biblioteci (de exemplu, în or. Alexandria), tipărirea în masă a cărților datorită apariției mașinii de dactilografiat, catalogarea cărților (aranjarea alfabetică după autor sau titlu), clasificarea cărților în categorii (după subiect, domeniu, etc.).

Managementul conținutului poate fi privit ca soluția generală a problemei de creare a conținutului (pentru producător), și cea de furnizare a conținutului (pentru consumator, utilizator).

Managementul conținutului constă în transmiterea conținutului potrivit persoanei potrivite la timpul și costul potrivit. [10]

Cu toate că CMS poate descrie și un proces manual de lucru, noi vom folosi acest termen pentru descrierea softului care asistă oamenii în crearea, managementul, transmiterea și navigarea conținutului.

Menționăm evenimentele care au marcat dezvoltarea CMS:

Anul (aproximativ)	Evenimentul
1975	CM pe mainframe-uri: publicarea electronică
1984	CM pe calculatoare personale: publicare desktop
1990	Utilizarea tehnologiei Client-Server
1995	CM pe Web: publicare web

Cronologia detaliată a istoriei CMS este descrisă în [11].

Publicarea electronică a accelerat exponențial crearea conținutului. Astfel, la sfârșitul anilor '80 apăruse fenomenul de "supraîncărcare informațională", situație agravată de apariția calculatoarelor personale, utilizarea cărora nu permitea un control centralizat.

La începutul anilor '90, calculatoarele personale au început să fie unite în rețea, ce a favorizat apariția aplicațiilor centralizate construite în baza principiilor client-server. Aceasta a dat posibilitatea restabilirii controlului asupra conținutului electronic, ca urmare a apărut epoca managementului documentelor.

Situația a început să se schimbe pe la mijlocul anilor '90 odată cu creșterea popularității Internetului. Către anul '89 Internetul conținea deja milioane de pagini web, și a devenit o afacere serioasă. Managementul documentelor a ieșit din modă, oferind locul managementului de conținut web (*engl. web content*).

Dar euforia din domeniul Tehnologiei Informaționale (TI) a trecut odată cu prăbușirea NASDAQ și dot.com-urilor în 2000. S-a revenit la soluții mixte ce conțineau documente obișnuite (pe hârtie) și electronice (conținut web) cu accentul pe dispozitive fără fir (*engl. wireless*), fluxuri audio/video (*engl. streaming*), și alte forme de conținut electronic. Ritmul implementării de soluții ale comerțului electronic B2C (afacere către consumator) (*engl. B2C – business to customer*) a scăzut, în schimb a crescut interesul față de comunicarea automatizată a conținutului electronic în afaceri prin intermediul rețelelor comerciale XML B2B (*engl. XML - eXtensible Markup Language, B2B - Business to Business*). Actualmente există mai multe variații de management al conținutului, care au la bază aceleași principii dar cu scopuri diferite:

- a) Managementul conținutului web;
- b) Managementul cunoștințelor;
- c) Managementul documentelor;
- d) Managementul bunurilor digitale (*engl. Digital Asset Management*);
- e) Managementul dosarelor (*engl. Records Management*).

1.2. Managementul conținutului

Managementul conținutului este procesul de aprovizionare sistematică și structurată, de creare, prelucrare, administrare, prezentare, procesare, publicare și reutilizare a conținutului [2].

În această definiție generală prin CM (managementul conținutului) se înțelege o succesiune de acțiuni, în care accentul se pune pe procesul în sine, și nu pe metodă. Noțiunea de CM este prea vastă pentru a putea formula definiții precise [8]. Ce înseamnă CM se întrebă mai des producătorii de CMS, fiindcă este o problemă de marketing cum să poziționezi un produs pe piață. Consumatorul de obicei înțelege în felul său conceptul de CM și nu este interesat de definiția specificată de producător din moment ce soluția oferită îi rezolvă problema sa. În [8] găsim interpretarea actuală a noțiunii de CM și modul în care producătorii de CMS de astăzi înțeleg și definesc CM. Menționăm că unele definiții sînt niște puncte de vedere ale analiștilor influențați de

politica de marketing al organizațiilor în care activează, deaceia ele reflectă mai mult specificul CMS-urilor acestor organizații.

O abordare mai pragmatică [9] determinată de Louis Rosenfeld, un arhitect al informației, coautorul cărții “Information Architecture” [12], realizează că CM este de fapt un domeniu și nu o definiție. Este un domeniu nou preocupat de facilitarea managementului și accesului la informație în medii corporative.

Deseori CM este confundat cu managementul documentelor. În §1.1 vorbeam despre ascensiunea managementului documentelor în domeniul sistemelor informaționale în ultimele decenii, mai apoi a urmat evoluția către managementul conținutului. Vom descrie managementul documentelor și al conținutului ca cititorul să observe delimitările și să înțeleagă domeniul abordat în teză.

Un document este o unitate de conținut, care poate fi identificată și manipulată ca o entitate, privită ca informație ce se referă la o anumită temă [13]. Documentele sînt structurate slab, pentru utilizarea de către oameni. Gartner Group a estimat că pînă la 90% din informația unei corporații se conține în documente.

Managementul documentelor înseamnă crearea, organizarea, utilizarea, manipularea și dispoziția documentelor în scopuri organizaționale [14]. În managementul documentelor electronice, aceste activități sînt efectuate utilizînd tehnologiile informaționale și aplicații soft.

Managementul conținutului este caracterizat de o varietate de instrumente și metode pentru colectarea, procesarea și furnizarea conținutului de diferite tipuri [15]. Se poate spune că managementul documentelor este o submulțime a CM. CM mai conține și conținut ce nu poate fi considerat sau recunoscut ca document din cauza utilizării diverselor tehnologii. Bob Boiko [16] consideră că CM este destinat pentru gestiunea publicațiilor Web și posibilitatea reutilizării conținutului. După Bob Boiko, managementul documentelor pune accentul pe gestiunea fișierelor, pe cînd CM se concentrează pe gestiunea componentelor, adică a unităților de conținut. Componentele și metadatele asociate permit managementul fluxului de lucru (*engl. workflow*) și crearea conglomeratelor (*engl. aggregations*), care reprezintă colecții de conținut existent (documente sau părți din acestea). Șabloanele definesc componentele, scripturile și conținutul static (logoul companiei, paneele de navigare) care sînt combinate pentru a crea *conținut web*.

1.3. Ciclul de viață al managementului conținutului

Scopul CM constă în furnizarea informației necesare utilizatorului în contextul unei acțiuni: de exemplu, consultarea catalogului de produse pentru luarea unei decizii chibzuite, căutarea de informație despre medicamente de către un doctor pentru alcătuirea unei prescripții. Rolul CM constă în optimizarea *comunicării* dintre persoana care crează informația și persoana-consumator de informație (în contextul unei acțiuni sau a unui proces). Însă comunicarea nu este scopul final: comunicarea catalizează procesul de lucru, iar procesul propulsează afacerea.

Rezultatul final al unei comunicări eficiente este luarea unei decizii. Dacă consumatorul obține informația de care are nevoie, el cumpără (sau nu) un produs. Dacă doctorul este asigurat că are suficientă informație pentru a prescrie o rețetă, boala pacientului este luată sub control. Cu cît comunicarea este mai optimizată, cu atît procesul de luare a deciziilor devine mai rapid și eficient, ca în final organizația respectivă să devină mai flexibilă și productivă.

Privite în acest mod, tehnologiile de management al conținutului trebuie să faciliteze: 1) colaborarea și 2) integrarea cu alte aplicații și procese de lucru.

- *Colaborare* înseamnă că conținutul este disponibil într-o formă convenabilă pentru etapele de creare, verificare și utilizare. Totodată aceasta înseamnă că controlul accesului este suficient de flexibil pentru a asigura lucrul unui grup de utilizatori, fiecare cu roluri și drepturi multiple în CM;
- *Integrarea* este similară colaborării, dar din punct de vedere al calculatorului. Gradul de integrare (drepturile de acces) al aplicațiilor exterioare variază.

Ciclul de viață al conținutului poate fi divizat în etape. În literatura de specialitate întâlnim divizări cu număr variat de etape, dar care descriu același proces:

Bob Boiko [16] accentuează trei „părți majore” a unui CMS: 1) colectare, 2) management (fluxuri de lucru, aprobare, controlul versiunilor, repozitoriu), 3) publicare.

Gerry McGovern [17], de asemenea, vede trei procese, dar le numește: 1) creare 2) editare 3) publicare.

JoAnn Hackos [18] descrie patru „componente”: 1) conținut 2) repozitoriu 3) asamblare/stabilirea legăturilor 4) publicare.

Aceste abordări au ca premisă instrumentele de bază ale CM: controlul accesului, sistemul de control al versiunilor, editare, fluxul de lucru, pregătirea tehnologică a procesului de lucru (*engl. staging*), personalizare și localizare.

Un raport recent [19] menționează cinci etape, și adaugă o etapă nouă: cea de negociere (*engl. transact*) care răspunde de legătura inversă (*engl. feedback*) cu vizitatorii site-ului.

Ciclul de viață al conținutului cuprinde următoarele etape:

- *Creare*. Crearea conținutului de către utilizatori, transformarea conținutului în alte formate, crearea datelor de suport (metadate);
- *Management*. Gestiunea conținutului de către utilizatori în cadrul unui repozitoriu, controlul de versiuni, controlul accesului la conținut (drepturi de citire/scriere);
- *Integrare*. Posibilitatea accesării conținutului de utilizatori în mai multe formate și din mai multe repozitorii. Îmbinarea conținutului cu alte aplicații și procese de lucru. Dacă în cadrul unei organizații documentele sînt gestionate de un sistem, elementele multimedia de altul, atunci capacitatea de integrare a CM oferă o interfață generală, comună de acces la mai multe repozitorii pentru utilizatori și aplicații. Mai mult ca atât, capacitatea de integrare poate reuni și procese de lucru;
- *Negociere*. Posibilitatea de a obține un profit sau de valorificare a conținutului. Pentru un editor comercial aceasta ar însemna utilizarea unor componente tehnologice pentru vânzarea conținutului pe Web. Exemplu: funcționalitate integrată de creare a campaniilor email: Web portalul de suport al consumatorilor ce în mod automat înștiințează un grup de clienți despre apariția unor produse noi ținînd cont de preferințele acestora;
- *Distribuție*. Publicarea conținutului în mai multe forme pentru uz intern (în cadrul CMS) și extern (WAP, HTML, XML, PDF, etc).

Odată cu maturizarea tehnologică a CM, la fel a evoluat și noțiunea de „obiect de conținut”. Schimbarea, în timp, aduce cu sine modificarea nivelului de complexitate, granularitate și diversitate la care a ajuns obiectul de conținut. Într-un sistem de management al documentelor, prin

obiect de conținut se subînțelege însăși documentul și metadatele asociate acestuia. Odată cu apariția CMS pentru Web (WCMS), „dimensiunea” (granularitatea) obiectelor gestionate a devenit mai mică: dacă în managementul documentelor un articol era un document, atunci în CMS acest articol este compus din mai multe unități: autorul, titlul articolului, capitole separate. Necesitățile reale sînt și mai complexe: obiectele de conținut pot fi obiecte compuse din alte obiecte sau reprezenta o agregare (interpelare) de mai multe obiecte.

Un document complex al unei organizații poate fi generat din mai multe obiecte de conținut extrase din locații diferite. Tehnologiile CM trebuie să ofere posibilitatea de lucru cu astfel de obiecte.

Volumul de informație care devine mai apoi conținut este impunător și greu de gestionat în totalitate, doar o parte din conținutul gestionat este reprezentat sub formă de obiecte compuse și flexibile. Organizația de talie medie se confruntă cu tipuri de conținut complex și neomogen, gestionat de mai multe sisteme în care metadatele se descriu în mod diferit. Necesitatea de integrare a conținutului și proceselor de lucru a crescut odată cu transferul acestor procese pe Web.

Un criteriu al completitudinii soluției de management al conținutului este flexibilitatea, granularitatea cu care obiectele de conținut de diferite tipuri sînt gestionate pe parcursul etapelor de management al conținutului. Succesul unui CMS este determinat de compatibilitatea integrării conținutului din cadrul sistemului cu alte aplicații și procese de lucru.

În tendința de răspîndire și adaptare al aplicațiilor în spațiul Web și alte zone, un factor important este integrarea. Pentru a putea “aproviziona” aceste aplicații cu conținut, tehnologiile de management al conținutului trebuie să suporte o infrastructură distribuită.

Extindere și Specializare în domeniul CMS

În ciclul de viață al unei soluții soft putem identifica două tendințe de dezvoltare: de extindere sau specializare al funcționalității sistemului. Pentru CMS, procesul de specializare îl urmărim în sistemele complexe ce se bazează pe servicii de integrare. Exemple: managementul fluxului de documente, managementul de portaluri, sisteme de management al relațiilor cu clienții. Extinderea se manifestă prin diversificarea metodelor de furnizare a conținutului: incorporarea funcționalităților de creare, publicare și syndicare ca parte componentă a unei platforme mai mari. În fig.1 sînt prezentate soluțiile specializate ce au la bază managementul conținutului.

1.4. Clasificarea sistemelor de management al conținutului

Noțiunea de CMS este greu de definit [8], multe produse pretind a fi soluții CMS complete: jurnale personale (personal weblogs), wiki-uri (wiki este un site web care permite redactarea sa de către vizitatori și este destinată editării colective), portaluri de noutăți.

Cererea imensă de pe piața CMS a atras mulți producători de soft: la data de 22 ianuarie 2006, în categoria „management de conținut” din directoriul DMOZ [20] erau 587 de companii producătoare CMS.

Marketingul online și distribuția produselor prin intermediul web-ului înseamnă cheltuieli mici necesare pentru a intra pe piața CMS. Dar mulți producători nu conștientizează că softul este doar o parte a unei soluții de management al conținutului.

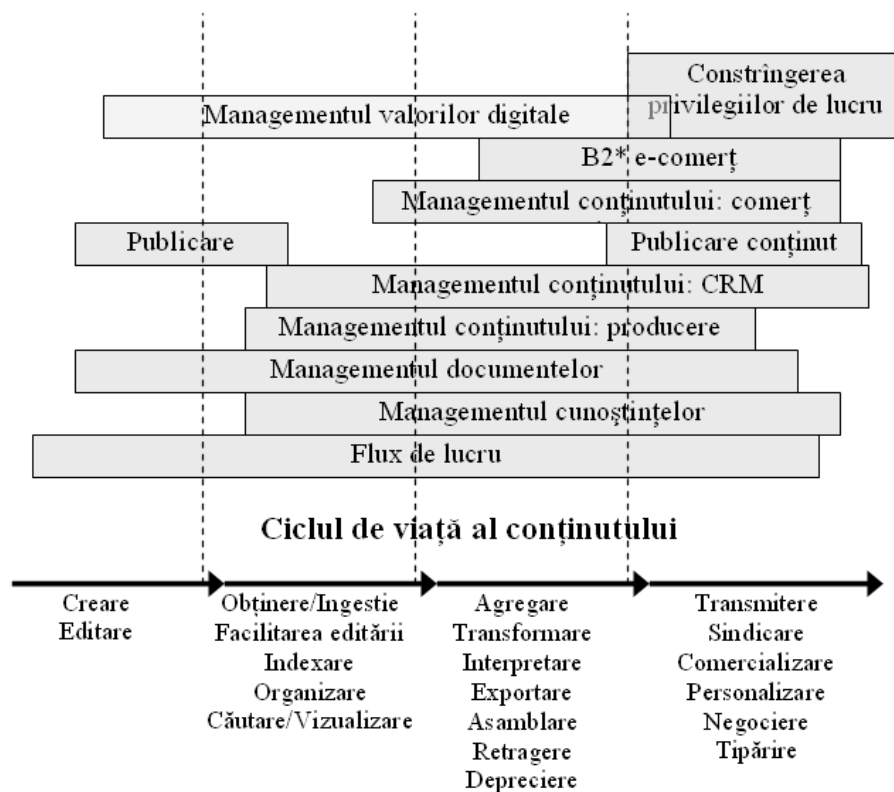


Figura 1: Extensii și specializări CMS

Sursa: [21]

Probabil cel mai important este faptul că majoritatea celor implicați în managementul site-urilor se confruntă cu probleme de control al procesului de publicare al conținutului. Ca rezultat, un număr impunător de organizații sau chiar indivizi au elaborat soluții CMS proprii. O parte din ei au decis să lanseze pe piață produsele sale prin vânzări directe sau darea în arendă (prestatori de servicii soft: Application Service Providers) .

Cu toate acestea, ținând cont de proveniența lor, putem defini câteva clase generale de CMS. Asistăm la o convergență a funcționalității oferite de CMS, însă produsele finale sînt orientate la piețe diferite. Tabelul ce urmează oferă o listă de abordări generale care stau la baza elaborării unui CMS:

Produsele “deschise” și mediile de dezvoltare CMS (*engl. framework products*) sînt mai preferabile din punct de vedere strategic, dacă se preconizează o dezvoltare continuă a site-ului web. Experiența arată că implementarea sistemelor mari costă mai mult decît prețul de achiziționare ale lor. Nu putem vorbi despre procurarea unui CMS, ci despre procurarea și adaptarea CMS.

Tip	Origine	Calități notorii	Exemple
Sisteme de management al documentelor	Soft destinat managementului depozitului de informație și publicarea internă a informației corporative.	Ciclul de viață al documentelor, procese de lucru, metadate, traducere.	Documentum, Panagon 2000, ChangingPages
E-noutăți / Publicații periodice	Instrumente pentru facilitarea publicării online a revistelor și noutăților, liste de discuții electronice	procese de lucru simplificat, publicarea rapidă a conținutului simplu, instrumente simple de creare a conținutului, livrarea la timp	Eroom, Expressroom, Conversant, SlashDot, Frontier

E-business / E-comerț	Softul din spatele magazinelor online și sistemelor de management al relațiilor cu clienții	Management simplificat al bazelor de date, personalizarea site-ului web, sisteme de tranzacții incorporate	Vignette, Broadvision, ATG Dynamo, Open Market
Sisteme de control al versiunilor	Soft de control al proceselor, controlul surselor între un grup de programatori	Redactare în baza rolurilor, controlul versiunilor, automatizarea proceselor de lucru, sisteme bazate pe șabloane	Content Management Studio, Interwoven TeamSite
Soft de complexitate medie bazat pe „steroidi”	Instrumente de creare a site-urilor dinamice: PHP, ASP, ColdFusion, JSP. Produse ca WebObjects, Cache Objects for the Web și Tango 2000	Livrare dinamică și medii simple lucru	Enhydra, Midgard (PHP), Dmind DSM (ColdFusion), SiteGenesis, Obtree.
Medii (frameworks) de management pentru Web	“generația a doua” de instrumentariu, elaborate din start pentru crearea dinamică a site-urilor web și gestiunea conținutului.	Inercarea de a cuprinde și îngloba toată funcționalitatea necesară, înlăturînd necesitatea de utilizare a instrumentelor auxiliare	ACS, eGrail, Engenda, Mediasurface, NetObjects Fusion, Spectra, Xpedio, Zope
Procesoare XML	A treia generație de produse CMS bazate pe tehnologiile XML	Control granular al conținutului și reutilizarea acestuia. Cu toate că multe din produsele de mai sus pot utiliza XML, produsele din această categorie sînt elaborate special pentru crearea site-urilor folosind XML pentru stocarea datelor și tehnologiilor derivate (XSLT, RDF) pentru managementul și livrarea acestuia.	Cocoon, Interwoven TeamSite Templating, Lychee, Rhythmix, Tamino, POET

Tabelul 1. Tipurile de CMS

Sursa: [21]

Din punct de vedere al raportului preț/funcționalitate, pot fi evidențiate 3 tipuri de CMS: 1) sisteme de tip cutie (*engl. box systems*); 2) platforme (medii) de dezvoltare; 3) sisteme atotcuprinzătoare (*engl. all-in-one solutions*).

Sisteme Cutie

Locul apariției: agențiile de publicitate.

Poziționarea de către producător: platformă care oferă utilizatorului mediu posibilitatea de creare și management al site-ului web.

Ideea de bază: Site-ul web este elaborat în exclusivitate de către consumator. Produsul este conceput ca o soluție ieftină.

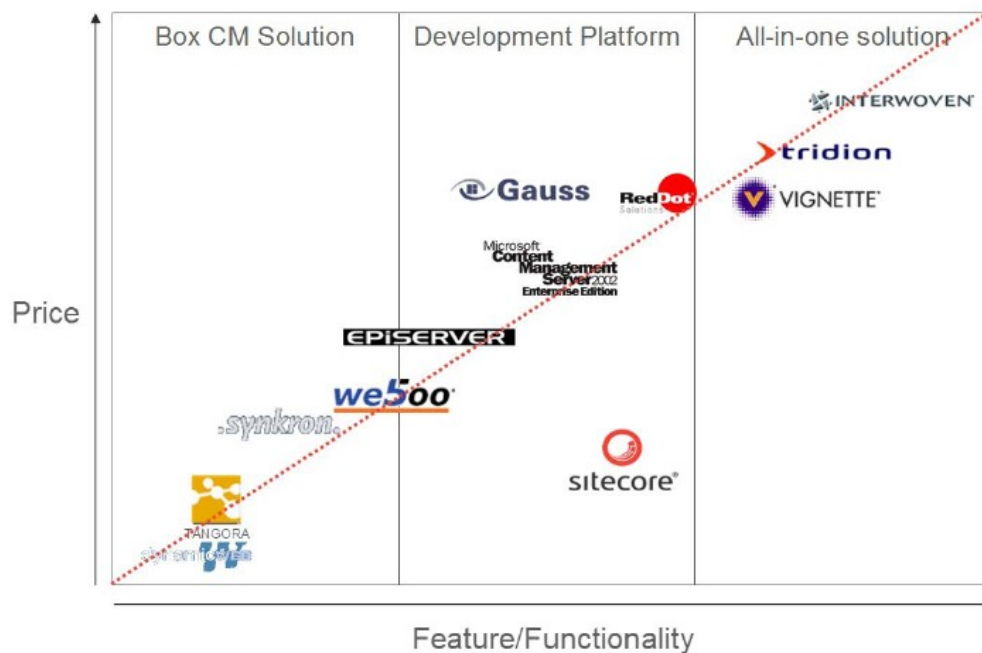


Figura 2. Preț vs. Funcționalitate
sursa: [22]

Avantajul principal al acestor sisteme: timp și preț redus de implementare a site-ului. Aceste produse sînt limitate din punct de vedere al designului, funcționalității și capacității de integrare. Sistemele cutie pot fi numite și instrumente de publicare, în sensul că ele rezolvă problema de prezentare a paginilor web în Internet.

Medii de dezvoltare (Development platforms)

Locul apariției: companiile din sectorul TI

Poziționarea de către producător: crearea unei platforme ce permite dezvoltatorilor de soft să elaboreze un site web care mai apoi să fie gestionat de către editori.

Ideea de bază: să ofere două instrumente. Unul pentru elaborarea site-ului sau a unei aplicații internet și altul destinat clienților pentru managementul conținutului.

Soluțiile ce au la bază un mediu de dezvoltare sînt oferite de consultanți TI. Managerii de conținut reînvoesc site-ul ca și în cazul sistemelor cutie (uneori chiar mai simplu). Aceste soluții nu au limitele sistemelor cutie, dezavantajul fiind costul inițial ridicat, deoarece este nevoie de programatori pentru a crea o soluție specifică în conformitate cu cerințele clientului. Ele sînt o alegere strategică pentru companiile care doresc mai mult decît o simplă publicare a informației în internet.

Sisteme atotcuprinzătoare

Locul apariției: producători ai sistemelor de gestiune a documentelor.

Poziționarea de către producător: un sistem pentru managementul tuturor tipurilor de conținut și procese de lucru în cadrul unei întreprinderi.

Ideea de bază: pe lângă managementul complet al conținutului, îndeplinirea a cît mai multe sarcini importante din activitatea întreprinderii.

Aceste sisteme nu se axează doar pe managementul conținutului în internet. Ele sînt o alegere excelentă pentru reconstrucția în întregime a infrastructurii TI dintr-o întreprindere. Cu toate că oferă posibilități de publicare a informației în internet, ele complică prea mult procesul de dezvoltare a site-ului. Costul acestor sisteme începe de la \$100,000 fără cheltuielile de consultanță.

1.4.1 Proiecte CMS cu cod deschis (open source)

Spre deosebire de soluțiile comerciale, CMS-urile cu cod deschis (*engl. open source*) nu încearcă să încorporeze funcționalități “la modă” pentru a deveni mai populare ci se concentrează pe comunitatea de utilizatori care au necesități bine definite față de managementul conținutului. Exemple de sisteme open source populare: Zope – server de aplicații pentru elaborarea CMS; Mambo – un CMS popular; WordPress – platformă de publicare personală (*engl. blog*).

În raportul [23] din ianuarie 2006 sînt descrise cele mai populare CMS-uri cu cod deschis.

Impactul Open Source în evoluția CMS

- Componentele modulare ce apar în rezultatul proiectelor comune vor determina organizarea funcțională a arhitecturii sistemelor și nu drepturile de proprietate sau durata licenței;
- Dovezi clare de aplicabilitate și compatibilitate cu anumite aplicații datorită distribuției de extensii și adaptări asociate cu fiecare implementare;
- Cu toate că proiectele CMS-urilor cu cod deschis sînt mai bine organizate și distribuite, lor le vor lipsi funcționalități de nivel înalt, cum ar fi: algoritmi moderni de căutare, suport multimedia, criptografie de ultimă oră, managementul valorilor digitale. Organizațiile care mizează pe avantajul în afaceri adus de CMS spre deosebire de concurenții săi, vor prefera soluțiile comerciale și garanțiile oferite prin procurarea acestora.

1.5. CMS: priorități și lacune

Managementul conținutului a devenit un domeniu important odată cu apariția Internetului și, ca consecință, a informatizării societății. Sistemele de management al conținutului au apărut datorită cererii de pe piață: companiile aveau nevoie de instrumente pentru publicarea informației. Interesul comercial a favorizat apariția unui număr impunător de CMS-uri [20], însă ele nu se bazau pe investigații științifice și nu veneau cu o schimbare de paradigmă, nu propuneau metode bine formalizate de rezolvare a unor clase de probleme, ci erau simple instrumente rudimentare care satisfăceau cerințelor pieței. Rolul crucial al informației în activitatea cotidiană a organizațiilor, necesitatea utilizării internetului (dependența de internet) și, ca urmare, importanța industrială a CMS a determinat interesul academic față de acest domeniu.

Valoarea unui CMS comercial depinde de operativitatea de implementare a inovațiilor din domeniul tehnologiilor informaționale. Astăzi accentul se pune pe facilitățile de integrare cu alte aplicații, deaceia CMS ce nu vor putea interacționa cu alte servicii și aplicații moderne vor deveni instrumente izolate și ineficiente. Din acest motiv, considerăm neesențială o comparare a posibilităților oferite de CMS actuale și identificarea lacunelor acestora. Interes științific prezintă ideea, abordarea ce stă la baza unui CMS și prioritățile obținute în urma aplicării metodologiei la elaborarea mai multor CMS-uri.

O listă exhaustivă de CMS cu posibilitatea de comparare în baza unei taxonomii [24] este accesibilă pe adresa <http://www.cmsreview.com/>. Un alt site ce descrie posibilitățile CMS și permite compararea pînă la 10 sisteme simultan este <http://www.cmsmatrix.org/>, care la 23 ianuarie 2006 avea în baza de date 509 sisteme de management al conținutului. O listă vastă de resurse CMS este indicată pe adresa <http://www.nmpub.com/cmsresources/>. Aceste referințe au fost menționate pentru a ghida cititorul care are nevoie de un CMS. Evaluarea CMS poate fi făcută în baza criteriilor descrise în [25].

Investigațiile științifice în managementul conținutului au un caracter fragmentar, în sensul că se axează pe segmente înguste ale domeniului (de exemplu, interacțiunea cu baze de date, interfața

utilizator, scalabilitate, arhitecturi distribuite, etc). Există puține abordări [26,27] și metode noi de construire a CMS. Considerăm că cauza principală constă în aceea că:

domeniul CMS este vast, deaceia astfel de abordări trebuie să fie abstracte, cu substrat teoretic, dar care se bazează pe o experiență practică suficientă. Și fiindcă domeniul este nou, prea puțini sînt cei, experiența și cunoștințele cărora le-ar permite argumentarea și impunerea unor modele noi de construire a CMS.

Lipsa unor principii clare de proiectare și întreținere ulterioară a site-urilor este una din cauzele datorită căreia 61% din companiile ce utilizează sisteme de management al conținutului se bazează pe actualizarea manuală a site-urilor lor [5]. Trebuie conștientizat faptul că elaborarea unui site, în afară de procurarea unui CMS, presupune și adaptarea acestuia. Din rezultatele chestionarului efectuat pe un eșanșion de 63 de persoane [4], 54% din cei chestionați au răspuns că CMS necesită mult timp pentru ajustare:

Care au fost problemele cu care v-ați confruntat la elaborarea/implementarea CMS ?

Softul comercial este prea scump	57.1%
Necesită mult timp pentru ajustare	54%
Lipsa de facilități pentru migrarea conținutului existent	50.8%
Nu este suficient de flexibil pentru a se acomoda designului necesar	47.6%
Este complicat de evaluat producătorii CMS	47.6%
Softul comercial necesită mult timp pentru implementare	44.4%
Dificultăți la integrarea cu alte sisteme	44.4%
Nu permite ajustări suficiente	41.3%

Sursa: [4]

Vom trece în revistă unele rezultate [28,29,30] cărora le corespund componente similare din CMS-ul propus în teză și lucrări [26,27] în care o atenție deosebită se acordă modelelor conceptuale ale site-urilor. Vom descri problemele comune CMS-urilor actuale.

XXL [28] – un sistem interactiv de proiectare a interfețelor utilizator bazat pe conceptul echivalenței dintre reprezentarea text și cea vizuală. XXL prezintă interfața în 3 moduri: textuală, grafică și idiomatică (*engl. widget*). Reprezentarea text corespunde codului sursă. Cea grafică este reprezentarea iconică a textului. Aceste două perspective constituie specificarea “abstractă” a interfeței. “Rezultatul” acestei specificări este înfățișarea idiomatică.

XcoP [29] - depozit de conținut XML (*engl. XML content repository*)

Depozit bazat pe sisteme obiectual-relaționale de gestiune a bazelor de date ce îmbunătățește managementul documentelor XML prin exploatarea caracterului structurat al acestora. Documentele sînt numite fragmente. Un fragment este compus din conținut textual și poate include la rîndul său alte fragmente. Un fragment reprezintă o înregistrare dintr-un tabel al bazei de date. Fragmentele pot fi procesate parțial, ceea ce facilitează reutilizarea și exclude redundanța datelor.

Newtenberg Engine [30] – platformă de modelare rapidă a site-urilor web bazată pe boxe. Definiția boxei este compusă din selectare, reprezentare și reguli stilistice (*engl. style rules*). Boxa este elementul principal folosit de arhitectii de informație la construirea paginilor. Unitatea elementară de conținut este numită “Eidox” și constituie abstracția unui document ce include conținut, proprietățile și istoria modificărilor (cine și cînd a redactat documentul, porțiunile modificate)

XcoP, Newtonberg Engine și NeoSite se aseamănă prin modul în care este stocată și utilizată informația. *Fragment* din XcoP, *eidox* din Newtonberg Engine și *datacell* [31] din NeoSite este același lucru - un atom de informație ce nu mai poate fi divizat.

Lucrările menționate sunt niște lucrări fragmentare, rezultatele cărora pot fi folosite în cadrul CMS. Pentru a obține un CMS viabil în care sunt integrate componente ce acoperă ciclurile de viață ale managementului conținutului, este necesară o arhitectură pe potrivă, ce are la bază un model conceptual.

Strudel [26] – specificare declarativă a conținutului și structurii site-ului.

Strudel aplică concepte din sisteme de gestiune a bazelor de date la procesul de construire a site-urilor web. Se încearcă separarea datelor de crearea și mentenanța structurii site-ului și prezentarea grafică (html). Se propun trei etape de construire a site-ului: 1) Este creat un model de date ce cuprinde toată informația afișată pe site 2) în baza acestui model, structura site-ului este definită în mod declarativ ca o interpelare (*engl. query*) a datelor 3) rezultatul evaluării interpelării este un graf al site-ului și reprezintă conținutul și structura acestuia 4) este aplicat un set de șabloane HTML pentru prezentarea grafică.

Limbajul de interpelări StruQL propus operează cu date semistructurate și asigură scalabilitatea sistemului. Strudel poate fi utilizat de persoane calificate și nu oferă mecanisme generice de introducere a informației în sistem și este utilizat doar în cercurile academice. Sistemul este interesant prin modalitatea de construire a site-ului ca un tot întreg însă necesită eforturi considerabile pentru definirea datelor și elaborarea definiției structurii în baza interpelărilor. Strudel poate fi aplicat doar site-urilor statice, nu este clar dacă există posibilitatea de reconstruire doar a unei părți a site-ului.

WebML [27] – sistem de notații pentru specificarea vizuală a site-urilor complexe la nivel conceptual.

Spre deosebire de modelarea datelor, care este o activitate consolidată, modelarea hipertext este o disciplină mai tânără căreia îi lipsește o bază organizată de concepte, notații și metode de proiectare. Limbajul WebML oferă instrumente și notații elementare pentru modelarea hipertext. Ca bază se ia modelul Entitate-Relație [32] ce folosește concepte simple și expresive pentru specificarea modelelor exprimate prin notații grafice intuitive. Modelul propus în WebML trebuie înțeles ca o prelungire a modelului Entitate-Relație, care dă posibilitate programatorului să extindă schema de date (*engl. data schema*) a aplicației cu specificarea hipertextului utilizat pentru publicarea și manipularea datelor. Componentele principale ale WebML sînt: paginile, unitățile și legăturile, organizate în structuri modulare numite zone și vederi ale site-ului (*engl. site views*). Unitățile sînt fragmente atomice de conținut, ce oferă alternative de combinare dinamică a conținutului extras din entități și relații al schemei de date. Totodată, unitățile permit specificarea formelor de introducere a datelor de către utilizatori. Din unități sînt formate paginile, care sînt elementele de interfață oferite utilizatorilor. Paginile și unitățile sînt conectate într-o structură hipertext. Legăturile reprezintă esența modelării hipertext: ele exprimă posibilitatea de navigare și transmitere de parametri de la o unitate la alta, necesari pentru generarea conținutului unei pagini.

Specificarea site-ului în WebML constă din 4 modele:

- modelul structural, exprimă conținutul site-ului sub formă de entități și relații;
- modelul compozițional, specifică paginile din care este compus hipertextul, și unitățile de conținut al acestor pagini;
- modelul navigațional, exprimă legăturile stabilite între pagini și unități de conținut pentru a forma hipertextul;
- modelul prezentațional, exprimă aranjamentul și aspectul grafic al paginilor.

WebML a stat la baza programului WebRatio (www.webratio.com), succesul comercial al căruia demonstrează importanța modelării conceptuale al site-urilor web. Accentul în WebRatio se pune pe modelare și reutilizarea implementărilor precedente, care sînt generate în mod automat sau semi-automat din modele de nivel înalt. Valoarea abordării propuse nu constă în componentele individuale ale sistemului, dar în definiția unei carcasi bine sistematizate în cadrul căreia elaborarea aplicațiilor Web poate fi organizată în concordanță cu principiile fundamentale ale ingineriei soft, și toate sarcinile sînt îndeplinite conform conceptelor, notațiilor și metodelor adecvate.

Modelarea conceptuală și generarea automatizată a site-urilor în baza unor modele generale reduce considerabil elaborarea unei soluții de management al conținutului, însă rămîne deschisă problema mentenanței. Regenerarea codului sursă în urma modificărilor din modelul conceptual devine problematică după ajustări manuale în codul existent. În [33] este descris un model conceptual de elaborare a aplicațiilor web cu ajutorul limbajului UML.

Majoritatea CMS-urilor actuale operează cu conținut în format XML, se accentuează posibilitatea de reutilizare a conținutului pentru generarea mai multor pagini. Cu toate acestea, procedeele de interconectare a conținutului din documente separate pentru generarea paginilor nu sunt extensibile. În sistemul EzPublish [34] există două tipuri de date: „relația obiectului” și „relațiile obiectului” cu ajutorul cărora între nodurile arborelului de conținut sînt stabilite relații. Nu există posibilitatea de specificare a atributelor relațiilor. Pot fi stabilite doar relații binare.

În sistemul Zope [35] există posibilitatea de definire a entităților (tipuri de noduri de conținut) în cadrul carcasi pentru managementul conținutului [36] (*engl. Content Management Framework*), dar nu este propus un mecanism de stabilire a relațiilor între noduri.

Majoritatea CMS-urilor, spre exemplu Midgard CMS (www.midgard-project.org) oferă interfețe de programare a aplicațiilor (*engl. API – Application Programming Interface*) către conținutul bazei de date, dar nu propun scheme generale ale bazei de date care vor putea fi aplicate la elaborarea site-urilor.

1.6. Căi de dezvoltare CMS

Elaborarea unei aplicații Web ce operează cu multe date necesită cunoștințe din diferite domenii pentru a face față diverselor sarcini: definirea structurilor de date pentru stocarea conținutului, conceperea interfețelor hipertext pentru navigarea și managementul conținutului, crearea stilurilor de prezentare, asamblarea de arhitecturi robuste și performante, integrarea cu aplicațiile existente și servicii externe.

Analizînd modalitățile existente de elaborare a aplicațiilor Web și a instrumentelor disponibile dezvoltatorilor, putem constata că principiile de inginerie soft nu sînt exploatate la maxim. Proiectanții de obicei elaborează aplicațiile Web prin aplicarea experienței și metodelor pe care le-au obținut elaborînd alte tipuri de sisteme: sisteme informaționale pentru întreprinderi, aplicații orientate obiect. Aceste abordări sînt adecvate pentru partea tradițională: proiectarea structurilor de date, implementarea logicii aplicației de partea serverului; însă ele nu i-au în considerație specificul aplicațiilor Web: furnizarea conținutului și serviciilor utilizînd o interfață hipertext.

Într-un studiu [37] efectuat cu scopul de a determina care din metodele ingineriei de dezvoltare a softului desktop sînt viabile pentru Web se menționează că: problemele majore întilnite la dezvoltarea de aplicații Web apar la etapele de analiză, determinare a cerințelor față de aplicație, testare, evaluare și mentenanță.

În lumea softului aplicativ (destinat utilizatorilor simpli, spre deosebire de softul de sistem utilizat de programatori, ingineri) dezvoltarea are loc pe orizontală și verticală. Dezvoltarea pe orizontală înseamnă integrare cu aplicații din alte domenii, tendință de soluționare a unui spectru larg de

probleme, asigurarea compatibilității cu aplicații similare și automatizarea procesului de migrare de la sisteme similare. Dezvoltarea pe verticală reprezintă specializarea pe un segment îngust al pieței cu scopul monopolizării unei nișe bine definite. Dinamica dezvoltării verticale este mai redusă fiindcă necesită investiții și cercetări serioase pentru a aduce îmbunătățiri considerabile.

Managementul conținutului pune accentul pe integrare, deaceia majoritatea CMS comerciale se dezvoltă pe orizontală, și tind să acopere o parte cât mai mare a pieței pentru a-și mări veniturile. Companiile care nu reușesc (sau nu au scopul) să facă față politicilor agresive de marketing ale concurenților se limitează la mentenanța soluțiilor sale, lucrează asupra stabilității, asigură suport clienților existenți.

Dacă cu 5 ani în urmă funcția CMS era facilitarea publicării de conținut pe site-uri statice, astăzi volumul enorm de informație, diversitatea surselor din care aceasta provine și caracterul dinamic al site-urilor demonstrează incompetivitatea arhitecturilor și concepțiilor sistemelor de ieri. Pentru a face sistemul mai atractiv producătorii se ocupau de înfrumusețarea sistemelor sale (interfețe grafice prezentabile). Un studiu [4] efectuat de Institutul pentru Arhitectura Informației (AifIA - <http://aifia.org>) arată că 61% din companiile ce utilizează softuri de management al conținutului se bazează pe actualizarea manuală al site-urilor sale.

Complexitatea și caracterul unic al proiectelor web cotidiente au determinat apariția platformelor de management al conținutului. Acestea sînt sisteme destinate construirii CMS prin adaptarea de module existente, care oferă limbaje speciale destinate proceselor de lucru specifice proiectelor web. De regulă aceste platforme sînt scumpe, dar totodată elaborarea și implementarea unui CMS cu ajutorul acestor platforme costă și mai mult. Decalajul de funcționalitate și preț dintre CMS-urile obișnuite și platformele CMS au determinat multe companii să dezvolte CMS-uri proprii, să nu plătească bani grei companiilor mari și totodată să nu devină dependenți de sistemele mediocre cu probleme de scalabilitate.

Problema platformelor de management al conținutului o constituie resursele considerabile pentru ajustarea funcționalității existente la specificul fiecărui proiect în parte după care urmează mentenanța sistemului care iarăși necesită cheltuieli considerabile personalului calificat din compania producătoare.

Datorită interesului academic față de CMS, viitoarele sisteme vor avea la bază modele clare de elaborare a structurii site-ului [27], limbaje speciale de interpelare a conținutului și specificare a legăturilor [26], paradigme de organizare și stocare a informației pentru generarea eficientă a paginilor dinamice. Rezultatele obținute în domeniul modelării conceptuale a site-urilor web vor fi folosite pentru elaborarea unor medii de dezvoltare rapidă RAD (*engl. Rapid Application Development*) de tipul C Bulider, Delphi, care se vor baza pe limbaje de programare existente ca Perl, PHP sau Java, dar în același timp vor oferi carcasa viitoarei aplicații web și vor impune standarde de elaborare care vor asigura scalabilitatea.

1.7. Formularea problemei

Analizînd mai multe CMS-uri, ajungem la concluzia că eficiența și utilitatea unei soluții CMS depinde de facilitățile oferite pentru trecerea de la o etapă la alta în ciclul de viață al conținutului. Separarea, deși logică, a etapelor de colectare, management și publicare a conținutului (§1.3) a favorizat apariția CMS-urilor mari: platforme de dezvoltare CMS și sisteme atotcuprinzătoare (§1.4), care necesită resurse considerabile pentru implementarea unui site Web (personal calificat, timp, costul ridicat al procurării și întreținerii). Ca rezultat, există o cerere crescîndă de tehnologii

ieftine de implementare a site-urilor de talie medie fiindcă utilizarea CMS-urilor mari nu va fi compensată din punct de vedere economic de beneficiile aduse de site.

Cele trei etape ale ciclului de viață al conținutului:

1. colectare;
2. management;
3. publicare.

includ respectiv următoarele sarcini:

1. Elaborarea schemei bazei de date pentru a stoca eficient conținutul. Spre deosebire de modelul rețea și modelul ierarhic, modelul relațional propus de matematicianul Dr. E. F. Codd în iunie 1970 s-a dovedit a fi mai simplu și mai eficient încât astăzi, modelul relațional este cel mai răspândit. Atât în modelul ierarhic cât și în modelul de tip rețea bazele de date dezvoltate sunt specifice aplicației. Dacă se dezvoltă o nouă aplicație trebuie să se dezvolte și o nouă bază de date. Acest lucru este dezavantajos deoarece păstrarea consistenței bazei de date în diferite aplicații este foarte dificil.
2. Elaborarea interfeței cu utilizatorul. Majoritatea sistemelor, inclusiv CMS, folosesc modelul ierarhic pentru interfețele de gestiune a datelor de către utilizator. În calitate de exemple menționăm structura ierarhică a sistemelor de fișiere (*engl. filesystem*), arborele din aplicația Windows Explorer, caracterul arborescent al formatului XML. Organizarea creierului uman este ierarhică. Informația categorizată, clasificată, organizată, prezentată sub formă de ierarhie este mai ușor percepută.
3. Extragerea conținutului din baza de date pentru generarea site-ului. Structura site-urilor web are la bază modelul rețea. Deseori aceeași informație este folosită la generarea mai multor pagini (căi alternative de navigare). Extragerea eficientă a informației din baza de date pentru construirea unei pagini depinde de modelul de date. Structura paginilor este mai „relaxată”, în sens de restricțiile impuse, decât structurile de date. Flexibilitatea în procesul de elaborare a structurii paginilor în particular și al site-ului în general deseori generează probleme de consistență și de mentenanță. Libertatea de exprimare pe care o are proiectantul site-ului trebuie încadrată într-un model de site web care ar facilita navigarea rapidă a unui volum mare și structurat de informație.

Utilizarea a trei modele diferite (relațional, ierarhic, rețea) la fiecare etapă a managementului conținutului necesită eforturi de conversie: sunt necesari pași intermediari pentru trecerea de la un model la altul. Aceste etape intermediare sunt caracterizate prin implementarea unor funcționalități adăugătoare care, deși ascunse de utilizatorul sistemului, asigură „traducerea” dintre modele. Se pune problema de modificare/ajustare a modelelor celor trei etape încât cheltuielile de realizare a pașilor intermediari să fie minime.

Este necesară studierea CMS-urilor existente destinate site-urilor bine structurate construite din baze de date: studierea arhitecturii, depistarea deficiențelor. Cunoștințele acumulate vor fi folosite la elaborarea:

1. Unui model de date flexibil în baza cerințelor și specificului impuse de aplicațiile Web;
2. Unei interfețe generale de management al conținutului pentru modelul de date propus;
3. Unui model de site Web pentru generarea eficientă a paginilor conform modelului de date.

Capitolul 2. Un model de Sistem de Management al Conținutului

2.1. Introducere

Sisteme specializate de management al conținutului au fost elaborate de autor începând cu anul 2000. Pe atunci ele automatizau doar o parte din mentenanța site-urilor: publicarea noutăților pe site, integrarea chestionarelor în paginile site-ului, managementul conținutului site-ului, dar fără o integrare cu partea de prezentare (de exemplu, construirea paginilor din șabloane HTML).

În baza experienței acumulate s-a constatat că procesul cel mai îndelungat și anevoios este crearea interfeței sistemului cu utilizatorul [38]. Majoritatea sistemelor de baze de date oferă instrumente elementare de manipulare și interpelare a datelor, insuficiente și incomode pentru utilizatori. Astfel, fiecare proiect web necesita elaborarea unei interfețe specializate care consuma 60% - 70% din timp. Mentenanța aplicațiilor (interfețelor) fiecărui proiect, lipsa unor standarde și modele de creare și utilizare a acestora ne-au determinat să căutăm/adaptăm/elaborăm un sistem general de management al conținutului în care vor fi integrate componentele ce pot fi utilizate în mai multe proiecte. Sistemele existente pe atunci fie că erau de tip închis și nu presupuneau modificări esențiale din partea altor programatori, fie că erau foarte elementare și nu puteau servi drept bază pentru dezvoltarea unei soluții complexe. Elaborarea modelului de CMS a însemnat realizarea celor trei obiective expuse în §1.7. Rezultatele cercetărilor erau permanent implementate în sistemul de management al conținutului numit NeoSite – un CMS care este rezultatul a patru ani de dezvoltare și restructurare continuă. În acest capitol vom desfășura soluțiile propuse pentru problemele formulate în §1.7 în contextul sistemului NeoSite, care reprezintă implementarea rezultatelor tezei.

2.2. Etapele de elaborare a site-urilor web

Procesul iterativ de elaborare al site-urilor web este dat în fig. 3. Dezvoltarea site-ului constă din câteva cicluri de formulare a problemei și de proiectare. Fiecare iterație produce un prototip sau o versiune parțială. Modelul iterativ este adecvat aplicațiilor web deoarece un aspect esențial al acestor aplicații este timpul limitat și modificările frecvente efectuate pentru satisfacerea cerințelor înaintate.

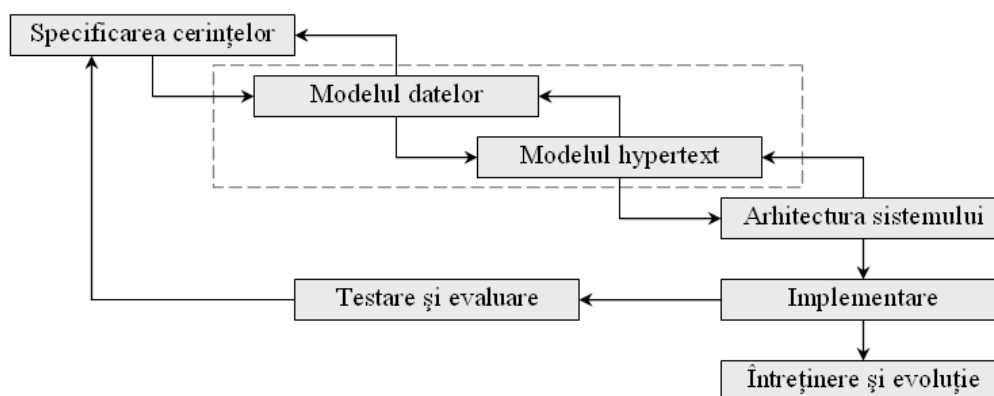


Figura 3. Procesul iterativ de dezvoltare a site-urilor web

Specificarea cerințelor este procesul în care analistul colectează și formalizează informația esențială despre domeniul viitoarei aplicații și a funcționalității așteptate. Rezultatul acestei faze este o specificare precisă adresată proiectanților și clientului pentru validare conform cerințelor înaintate de acest client. Aprobarea specificației de către client înseamnă că cerințele i-au fost satisfăcute și se poate trece la etapele de proiectare.

Proiectarea modelelor de date este o disciplină mai veche, cel mai popular model conceptual fiind modelul Entitate-Relație propus de Chen în 1976 [32]. Însă modelele de date pentru Web se deosebesc de aplicațiile obișnuite. În aplicațiile web prevalează operațiile de citire a datelor la etapa construirii dinamice a paginilor. Elaborarea schemei bazei de date cu satisfacerea formelor normale uneori nu este justificată practic.

Modelarea hipertext are ca scop specificarea interfețelor aplicației web: structura ierarhică a paginilor site-ului, datelor utilizate pentru construirea paginilor, legăturilor dintre pagini. Modelarea datelor și hipertextului sînt strîns legate, totuși, abordările existente [26,27] le examinează separat, motivînd aceasta prin gradul de flexibilitate obținut. Prețul acestei flexibilități este timpul pentru ajustarea unui model la schimbarea celuilalt (de exemplu, compatibilitate, optimizare). În teză se propune elaborarea în paralel al modelului de date și hipertext. În §2.3 este descris un model abstract al structurii paginilor web și enumerate cerințele pe care trebuie să le satisfacă o pagină. O proiecție a modelului hipertext pe planul modelării datelor este modelul relațional-ierarhic descris în §2.4.

Proiectarea arhitecturii constă în definirea elementelor hardware, de rețea și componentelor soft ce stau la baza aplicației ce oferă servicii utilizatorilor săi. Scopul acestei etape este identificarea unei combinații optime ale elementelor menționate pentru satisfacerea cerințelor înaintate din punct de vedere al performanței, securității, scalabilității, costului. În §2.5 sînt date aspectele tehnologice ale sistemului NeoSite.

Implementarea reprezintă producerea de module soft necesare pentru a transforma modelul de date și hipertext într-o aplicație bazată pe arhitectura propusă. În §2.6 este descrisă aplicația client al sistemului NeoSite, conexiunea dintre logica de partea serverului și aplicația client, descris procesul de construire dinamică al site-ului.

Testarea și evaluarea sînt activități de verificare a conformanței aplicației implementate la cerințele înaintate. Menționăm testarea funcționalității, performanței și utilității (*engl. usability, ease of use*). Întreținere și evoluție reprezintă modificările efectuate după lansarea aplicației. Propunerile și cerințele față de o aplicație existentă sînt analizate și efectuate modificări în modelul date sau hipertext.

2.3. Modelul paginii de conținut al site-ului structurat

Organizarea conținutului în paginile web este descrisă în șabloane. După implementarea mai multor site-uri, se conturează o structură comună a șabloanelor din diferite proiecte. Deosebim două tipuri generale de pagini web: pagini de navigare și pagini de conținut. Paginile de navigare sînt construite prin extragerea informației importante din paginile de conținut și prezentarea acesteia într-un mod accesibil utilizatorilor. Explorarea unui volum mare de conținut presupune diferite modalități/căi de navigare. Prin pagini de navigare mai înțelegem și facilități de filtrare și căutare a paginilor de conținut, acest tip de navigare îl vom numi dinamic. Scopul paginilor de navigare este facilitarea procesului de căutare a paginilor de conținut ce satisfac preferințelor utilizatorilor. Pagina de conținut este punctul final la care ajunge utilizatorul după parcurgerea unui șir de pagini de navigare. Cu cît drumul parcurs este mai mic, cu atît site-ul web este mai comod și util vizitatorilor. Dacă site-ul are mai multe tipuri de vizitatori (cu diferite criterii de căutare a informației), trebuie să existe metode alternative de navigare.

Paginile de conținut complexe prezintă probleme de mentenanță, deaceia ele sînt descompuse în pagini mai simple, legate între ele, fig. 4.

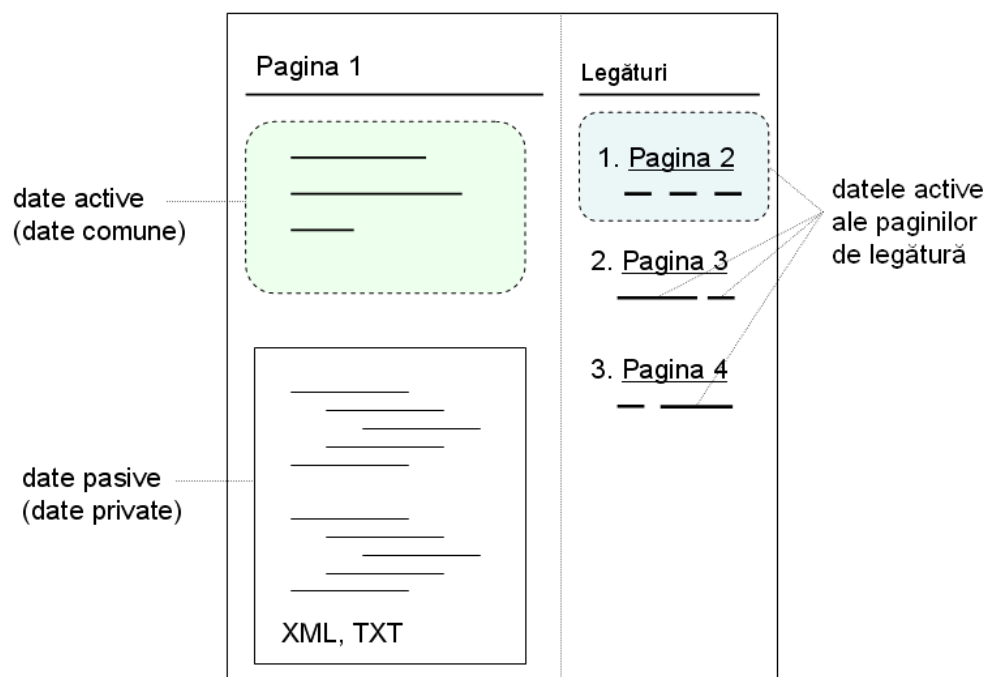


Figura 4. Model abstract al paginii de conținut

Proiectarea structurii paginii de conținut din fig. 4 pe modelul de date descris în §2.4 și invers, modificarea și descompunerea paginilor complexe în baza cerințelor impuse de modelul de date, este un proces continuu în proiectarea site-urilor cu NeoSite. Similaritatea dintre structura paginii și reprezentarea acesteia în modelul de date ușurează considerabil procedura de extragere și publicare a datelor.

La proiectarea paginilor trebuie să se țină cont de următorii factori:

Coerența:

Acest criteriu determină ușurința cu care vizitatorii își formează o viziune (*engl. mental model*) despre domeniul expus pe site în baza documentelor hipertext. Deosebim coerență locală, în contextul unei singure pagini și coerență globală, când vorbim de tot site-ul. Documentele ce folosesc îmbinări de elemente, paragrafe și ilustrații multimedia pentru a delimita conținutul sînt mai coerente (local) decît paginile ce conțin doar text. Coerența globală ține de problema “pierdut în hiperspațiu”: paginile de același tip trebuie să fie asemănătoare ca structură. Un factor important al coerenței globale este dificultatea de navigare: de ex. un site ce nu permite navigarea inversă sau conține legături arbitrare între pagini este mai greu de navigat decît unul cu legături bidirecționale cu un număr redus de legături dar care sînt potrivite contextului. Coerența globală este asigurată prin utilizarea unui număr redus (de obicei unul) de șabloane asociate unei entități.

Suprasolicitarea cognitivă:

Există o limită a volumului de informație pe care o poate procesa mintea umană. O aplicație ce folosește un număr excesiv de fonturi, culori pe fiecare pagină necesită de la utilizator mai mult timp pentru adaptare. Pe lîngă *adaptarea la interfață* există și problema *consecvenței*. Un site web ce prezintă obiectele în mod diferit (amplasate diferit) de la o pagină la alta este mai puțin consecventă.

Coerența și suprasolicitarea cognitivă se referă la interfața aplicației web.

Coeziunea informației într-un document:

Informația din cadrul unei pagini trebuie să fie tot atât de coezivă ca și în lumea reală. De ex. dacă un document conține informații despre cumpărători, atunci el este mai coeziv, deci mai bun, decât un document ce prezintă informații și despre cumpărători și despre vânzări. Prin document înțelegem nu o pagină web, ci unitatea cu care se operează în cadrul unei aplicații, în cadrul CMS în cazul nostru.

Conectarea informației între documente:

Documentele trebuie să fie independente unele de altele, adică să nu conțină informații ce se repetă. De ex. dacă două documente conțin numele și adresa unui consumator, aplicația va fi de o calitate inferioară față de cea în care datele ce se repetă sînt extrase și reprezentate de al treilea document, iar în cele două rămase va fi utilizat identificatorul document nou creat.

La proiectarea site-urilor mari ce prezintă diverse tipuri de conținut trebuie aplicate metode de facilitare a procesului de adaptare la conținut. Noțiunea de *adaptare la conținut* se referă la diversele tipuri de date incorporate în pagini hipertext. În practică însă, majoritatea modelelor și sistemelor implementate se concentrează pe elementele textuale, fără a lua în considerare informația audio și video. Procesul de adaptare poate avea mai multe etape, numărul acestora este direct proporțional cu lungimea medie a segmentelor de text utilizate. Un număr mare de elemente text concise ce descriu obiectele de conținut propriu zise, utilizate pentru a crea diferite scheme de navigare pe site, asigură flexibilitatea aplicației și satisfacerea necesităților utilizatorilor.

Procesul de adaptare la conținutul site-ului este asigurat de utilizarea următoarelor procedee:

Agregare – afișarea unui element ce reprezintă o colecție de elemente mai mici;

Recapitulare – reprezentarea unui volum mare de informație prin unul mai mic, ex: fragmente de text, mostre audio;

Filtrare – afișarea informației relevante;

Eliziune – afișarea unor exemple dintr-o mulțime mare de elemente pentru a-i da posibilitate utilizatorului să înțeleagă ce tip de informație este în această mulțime;

Condiționalizare – una din metodele simple dar foarte eficiente este utilizarea textului condiționat. Această metodă necesită divizarea unui text în unități mai mici, afișate în dependență de anumiți factori (nivelul utilizatorului, stările obiectului de conținut, context, etc.);

Extindere – la deschiderea paginii web, informația ce nu este relevantă utilizatorului este vizibilă parțial. Pe parcursul examinării paginii, utilizatorul are posibilitatea să vizualizeze complet informația opțională și să ascundă porțiunile ce nu-l interesează. Avantajul metodei de extindere/constrîngere a conținutului din paginile web dă posibilitate utilizatorului să se adapteze, iar sistemului să adapteze conținutul paginii.

Odată cu adaptarea la conținutul site-ului, are loc și *adaptarea la nivel de legături*. Procedeele folosite au ca scop reducerea suprasolicitării cognitive cauzate de sistemele complexe și pot fi grupate în următoarele categorii:

– Asigurarea unor puncte de intrare relevante în spațiul informațional. Utilizatorul care pentru prima dată vizitează un site trebuie să-și dea seama despre domeniul abordat pe site. Site-ul trebuie construit în așa mod, încît utilizatorii să poată găsi cît mai rapid informația căutată. Paginile site-ului trebuie să conțină informații succinte despre conținutul paginilor din interior, pentru al scuti pe utilizator de descărcarea unor pagini inutile. Informația importantă trebuie scoasă la suprafața site-ului constituită din pagini apropiate (în sensul numărului de pași efectuați) de pagina de start. Informația greu accesibilă (facilități de navigare insuficiente) va crea impresia unui site incomplet.

– Percepția legăturilor: ascunderea legăturilor spre alte pagini este folosită în cazul utilizatorilor noi al site-ului. Evidențierea legăturilor: utilizarea culorilor, texturii, sclipirii (*engl. blinking*). Cel mai des întrebuințat atribut este culoarea: în dependență de importanța legăturii în contextul unei pagini, pot fi folosite culori aprinse sau obscure.

– Sortare: o mulțime de legături ce duc spre pagini cu același tip de informație poate fi sortată după anumite criterii pentru a direcționa utilizatorul către paginile mai relevante.

– Legături semantice: adnotarea legăturilor cu o informație succintă despre conținutul paginii interioare. Puține site-uri acordă atenție modului în care sînt descrise legăturile. Titlul legăturii trebuie să descrie destinația și scopul, în baza acestei descrieri utilizatorul decide, să viziteze sau nu pagina. Timpul investit în crearea unor legături informative se răsfrînge pozitiv nu doar asupra vizitatorilor site-ului, ci și a popularității site-ului în motoarele de căutare. Site-ul cu o structură coerentă de legături bine descrise se poziționează mai bine în rezultatele căutării după cuvintele din titlul și textul aferent legăturilor.

Deosebim două metode de proiectare a siturilor: centrate pe utilizator (*engl. user centered*) și centrate pe date (*engl. data driven*). În primul caz proiectarea începe de la potențialii vizitatori ai site-ului (audiența site-ului): care sînt clasele de utilizatori și informația pe care o caută accesînd site-ul. În modelarea centrată pe date structura site-ului este concepută prin analizarea informației care va fi afișată pe acest site. Rezultatul este o structură completă, clară și bine interconectată a datelor publicate pe site. Modelul de pagină web propus se bazează pe metoda centrată pe date. Cu toate acestea, granularitatea sporită a datelor asigurată de modelul de date descris în §2.4 permite implementarea căilor alternative de navigare a informației pe site și permite vizitatorilor să găsească rapid informația de care au nevoie. Un exemplu de proiectare a site-urilor centrat pe utilizator este propus în [39].

2.4. Model relațional-ierarhic de gestiune a datelor

Scopul modelării datelor este descrierea formală dar intuitivă, o schemă conceptuală a specificațiilor de utilizare a informației în cadrul unei aplicații. De această schemă se va ține cont la proiectarea funcțiilor care vor opera cu aceste date și implementarea la nivel fizic al structurilor de date. Modelarea datelor este o disciplină consolidată din domeniul tehnologiilor informaționale, un model popular este modelul Entitate-Relație (E-R) [32].

Componentele de bază al acestui model sînt entitățile definite ca containere de date structurate și relațiile, care reprezintă legăturile semantice dintre entități. Entitățile sînt descrise cu ajutorul atributelor, și pot fi organizate în ierarhii care exprimă derivarea unui concept specific din unul general. Relațiile sînt caracterizate de condiții de cardinalitate care impun restricții asupra numărului de relații pe care le poate avea un obiect.

Entități:

Entitatea este conceptul central din modelul E-R. O entitate reprezintă descrierea trăsăturilor comune a unui set de obiecte din lumea reală. Exemple de entități sînt: persoană, teză, instituție. O entitate posedă o populație - setul de obiecte descrise de aceasta. Aceste obiecte sînt numite instanțe ale entității. De ex, populația entității „persoană” este un anumit grup de persoane (potențialul științific de ex.).

Atribute:

Atributele reprezintă proprietățile obiectelor din lumea reală care sînt relevante aplicației. Exemple de atribute sînt numele, adresa, telefonul unei persoane. Atributele sînt asociate cu entitățile, aceasta

înseamnă că toate instanțele unei entități posedă același set de atribute. Cu alte cuvinte, entitatea este un identificator al proprietăților comune unui set de obiecte, și aceste proprietăți sînt exprimate prin atribute.

Relațiile:

Relațiile sînt conexiunile semantice dintre entități, de ex: asocierea dintre o persoană și teza acestuia sau dintre o teză și recenziile ei. Semnificația asocierii este dată de numele relației stabilită de proiectant. De ex. relația dintre persoană și teză poate fi numită „Publicare”. Cea mai simplă formă de relație este cea binară. Utilizarea relațiilor de gradul N în care sînt implicate mai multe entități nu se recomandă [27]. Fiecare relație binară este caracterizată de două roluri, și exprimă funcția executată în cadrul relației. De exemplu, în relația „Publicare” rolurile sînt: „Publicat de” (teză spre persoană) și „A publicat” (persoană spre teză). Așadar, rolul în cadrul unei relații poate fi văzut ca o asociere orientată și conectează entitatea sursă cu entitatea destinație.

Au fost propuse modele conceptuale de proiectare a site-urilor [27] cu ajutorul modelului E-R, dar abordarea din [27] este foarte generală, ca rezultat, structura fizică a bazei de date obținut prin aplicarea modelului conceptual este una clasică, și nu reflectă modul în care de obicei (fig. 4) sînt elaborate paginile web. Proiectarea bazei de date, a interfeței cu utilizatorul, a structurii site-ului sînt văzute ca procese independente. Din punct de vedere conceptual aceasta este corect fiindcă se dorește o flexibilitate și divizare clară de etape în crearea unei aplicații. Însă în practică, flexibilitatea este costisitoare deoarece sînt necesare eforturi de integrare ale acestor nivele. Programatorii sînt cei care stabilesc legătura între baza de date și interfață. Dacă structura bazei de date ar putea fi proiectată direct într-o interfață utilizator care să asigure necesitățile de bază în administrarea site-urilor web, nu ar fi nevoie de programatori de fiecare dată cînd au loc modificări în structura bazei de date. S-a propus și argumentat teoretic [40] aplicarea modelului E-R la proiectarea interfețelor utilizator cu bazele de date. S-au elaborat [41] instrumente de generare automată a interfeței din definiția bazei de date. Dar nu s-au propus metode de proiectare a bazelor de date care să ia în considerație specificul organizării și șabloanele de administrare a datelor în aplicațiile web. Aplicarea riguroasă a formelor normale la proiectarea bazelor de date nu este întotdeauna justificată. Din punct de vedere practic, un model mai flexibil, simplu, și general al structurii bazei de date se dovedește a fi mai eficient.

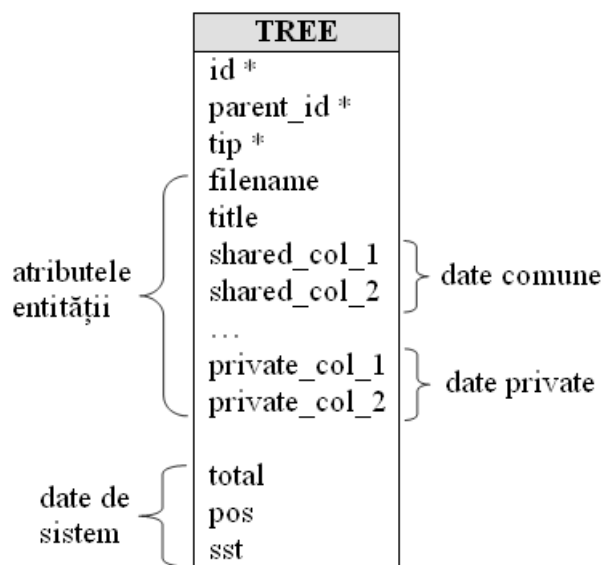
Propunem un model relațional-ierarhic de gestiune a datelor care să asigure:

- flexibilitate: modificări în definițiile entităților și relațiilor dintre acestea nu vor necesita modificări esențiale al structurii bazei de date;
- scalabilitate: nu vor exista probleme de performanță la mărirea volumului de date procesat;
- abstractizare: modelul trebuie să asigure gestiunea datelor pentru orice aplicație web orientată la publicarea informației. Structura tabelelor de bază trebuie să fie similară.

Prima versiune al acestui model este descris în [31].

Organizarea creierului uman este ierarhică. Informația categorizată, clasificată, organizată, prezentată sub formă de ierarhie este mai ușor percepută. Mulțimea instanțelor entităților este organizată într-o ierarhie, la nivel fizic ea este păstrată într-un tabel al bazei de date din fig. 5.

Coloanele *shared_col_1*, *shared_col_2* de obicei sînt de tip *mediumint*, *enum* sau *varchar*, uneori aceste coloane sînt indexate dacă site-ul oferă facilități de filtrare în baza acestor atribute. În dependență de tipul entității, aceste coloane pot conține diferite seturi de atribute. În mod normal, pentru fiecare tip de entitate trebuie creat un tabel aparte în baza de date [27,41] care va conține atributele specifice acestui tip.



id: identificatorul nodului
parent_id: identificatorul nodului tată
tip: identificatorul entității (tipul nodului)
definiția entităților este păstrată în alt tabel
filename: identificatorul nodului folosit în modelul
hipertext pentru a găsi nodul în baza
adresei URL.
title: denumirea nodului vizualizată în NeoSite și
elementul <title> din paginile HTML
shared_col: coloane ce conțin informație folosită la
generarea paginilor de navigare a
site-ului (index pages)
private_col: informație accesibilă din pagina de
bază a nodului.
total: numărul total de noduri descendente
pos: poziția relativă (numărul de ordine) a nodului
în raport cu nodurile
frati

Figura 5. Structura tabelului de conținut

Într-o schemă clasică a bazei de date există mai multe tabele. Relațiile deasemenea sînt păstrate în tabele separate. De obicei numărul tabelor de relații este mai mare decît cel al tabelor de conținut. Crearea unei interfețe abstracte de management al datelor dintr-un număr mare de tabele este complicat. Apar probleme de evoluție al acestei interfețe. De exemplu, există trei tipuri de entități, respectiv trei tabele de conținut. Pentru reprezentarea relațiilor binare va fi nevoie de 6 tabele: cîte două tabele pentru fiecare relație, dacă setul de atribute ce descrie relația este diferit, în dependență de orientarea acesteia. Aplicațiile web sînt extrem de dinamice, sînt frecvente fluctuațiile în numărul de entități și atributele acestora. Apariția unui tip nou de entitate ar însemna crearea de noi tabele, ce se va reflecta asupra interfeței de administrare a conținutului. Acest lucru este justificat pentru sistemele complexe unde o atenție deosebită se acordă securității și integrității datelor (de exemplu, procesarea tranzacțiilor bancare).

Site-urile web au un specific: bazele de date sînt solicitate intens cu operații de citire a informației, și trebuie elaborate modele care să țină cont de acest fapt. Generarea paginii web va utiliza un număr cît mai mic de adresări către baza de date pentru a nu suprasolicita serverul (în același timp site-ul poate fi vizitat de mai mulți utilizatori). Modelul de pagină web din fig. 4 ne sugerează că conținutul paginii este rezultatul asamblării datelor unei entități, datelor comune al entităților din relații, și atributelor ce descriu această relație. Site-urile dinamice cu facilități dezvoltate de navigare, vor avea un număr mare de relații, managementul cărora trebuie asigurat atît la nivel fizic (baze de date), cît și a interacțiunii cu utilizatorul prin intermediul unor interfețe comode.

Utilizarea relațiilor N-are adaugă încă un nivel de complexitate procesului de elaborare a structurii bazei de date și a interfeței cu utilizatorul. Se arată [27] că relațiile N-are pot fi reprezentate prin relații binare, dar nu este explicat impactul asupra interfeței cu utilizatorul. Propunem un model în care relațiile de orice grad sînt reprezentate sub formă de arbore. Toate relațiile sînt păstrate într-un singur tabel al bazei de date, iar interfața de management al relațiilor este comună pentru relațiile de orice grad. În fig. 6 este dată structura tabelului de relații. (În versiunile mai vechi ale sistemului NeoSite [6] erau doar relații binare simple: adică lipsa arborele relațiilor).

Coloanele *to_id_parent*, *from_tip*, *to_tip* sînt redundante: ele sînt prezente în tabelul TREE și au fost introduse din necesități practice pentru a optimiza o serie de interpelări SQL des întrebunțate la generarea site-ului.

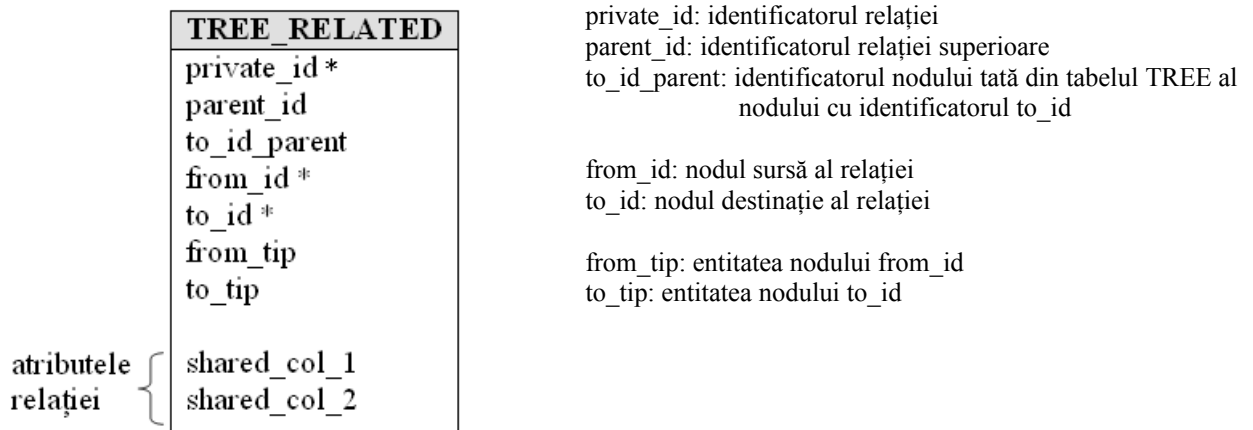


Figura 6. Tabelul de relații

În modelul propus, relațiile de grad N sînt reprezentate prin subrelații alcătuite din relații de grad N-1. Stabilirea unei relații de grad N necesită stabilirea implicită a N-1 relații binare.

Fie că avem o mulțime de elemente între care sînt stabilite relații de diferit grad. În fig. 7 este dată varianta clasică de reprezentare a relațiilor folosind mai multe tabele și varianta cu un singur tabel ierarhic cu redundanță.

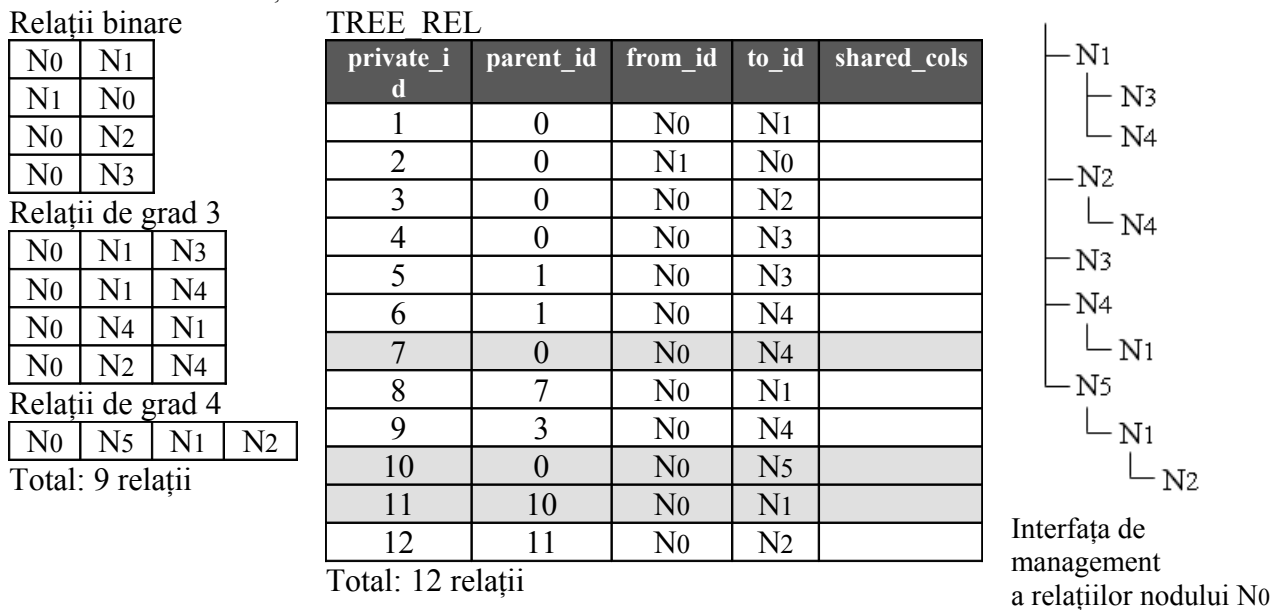


Figura 7. Reprezentarea relațiilor cu ajutorul unui tabel ierarhic.

Înregistrările evidențiate sînt relațiile implicite stabilite. Relațiile N-are sînt reprezentate de N-1 relații binare. Interfața de management al relațiilor este un arbore care reprezintă toate relațiile cu un anumit nod de start, N0 în cazul dat. Relația N1→N0 lipsește din arborele de mai sus, ea va fi vizibilă în fereastra de editare a nodului N1. Interfața de management al relațiilor este descrisă în §2.7. Posibilitatea de management al tuturor relațiilor inițiate de un anumit nod promovează modelul de pagină web descris în §2.3, fig. 4. Practica arată că cel mai des sînt utilizate relațiile binare și terțiare. Arborele de relații în acest caz va avea două nivele.

Utilitatea modelului constă în flexibilitatea cu care pot fi adăugate noi tipuri de relații fără modificarea structurii bazei de date. Acest lucru se obține prin introducerea de noduri ajutătoare în

arborele de conținut: o relație binară poate fi specificată cu ajutorul unei relații de ordin 3 în care primul nod de legătură este un pseudo-nod ce semnifică tipul de relație:

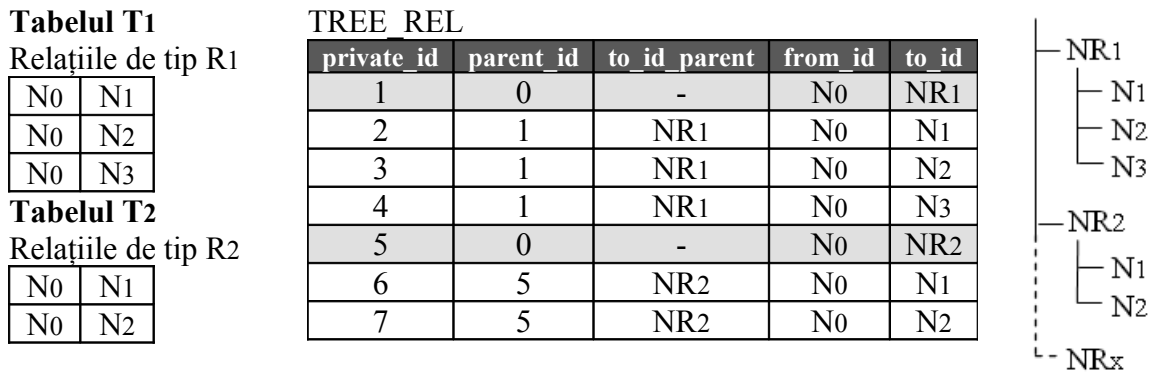


Figura 8. Reprezentarea relațiilor binare cu ajutorul celor terțiare

Nodurile NR1, NR2 sînt adăugate în arborele de conținut (tabelul TREE) ca noduri artificiale, atributele lor sînt folosite la descrierea tipului de relație R1, R2. Coloana indexată *to_id_parent* al tabelului TREE_REL (fig. 6) este identificatorul nodului tată din tabelul TREE al nodului cu identificatorul *to_id* și a fost adăugată pentru a optimiza interpelările SQL de extragere a relațiilor binare de un anumit tip reprezentate prin relații terțiare:

```
SELECT from_id, to_id FROM tree_rel WHERE to_id_parent = NR1
```

Rezultatul acestei interpelări va fi exact tabelul T1.

Setul de atribute ce descrie un anumit tip de relație este păstrat în coloanele *shared_cols* (fig. 7). Dacă fiecărui atribut îi va fi asociată o coloană din tabelul TREE_REL, atunci mulțimea coloanelor *shared_cols* este reuniunea atributelor tuturor relațiilor. De obicei atributele relațiilor sînt folosite la generarea paginilor de conținut și sînt rar utilizate în interpelările SQL. Pentru a evita crearea unui număr mare de coloane, atributele relațiilor sînt descrise cu ajutorul unui document XML stocat într-o coloană de tip BLOB sau TEXT.

Setul de coloane pentru fiecare tip de relație sînt specificate în secțiunea *linkage* din definiția entității descrise în format XML:

```
<entity>
  <memos>
    <f name="db_column1" />
  </memos>
  <fields>
    <f name="db_column2" editor="editor_name"> [editor_params] </f>
    <f name="db_column3" />
  </fields>

  <linkage mode="[tree, list]">
    <r [parent_id="id1" | tip="id1, id2, ..., idN" | parent_tip="id1" ]>
      <fields> ... </fields>
    </r>

    <r [RELATION_RULE]>
      <fields> ... </fields>
    </r>
  </linkage>
</entity>
```

Figura 9. Definiția entității

Secțiunile *memos* și *fields* descriu atributele entității: *memos* conține atributele de tip BLOB și TEXT, *fields* – câmpuri de tip integer, varchar. Secțiunile *entity/linkage/r/fields* au aceeași structură

ca și *entity/fields*. Relațiile pot fi gestionate în formă de listă (simple relații binare), și în formă de arbore. Câmpurile disponibile unei relații se determină după:

parent_id="id1, ..., idN"	identificatorul din tabelul TREE al nodului părinte
tip="id1, ..., idN"	entitatea nodului de legătură
parent_tip="id1"	entitatea nodului părinte

Managementul conținutului și relațiilor este efectuat prin interfețe generale construite automat din definițiile entităților (§2.7). Despre generarea interfețelor descrise în format XML am relatat și în [7].

2.5. Modelul hipertext al site-ului

Noțiunea de hipertext în viziunea lui W.Weiland și B.Shneiderman “denotă o tehnică pentru organizarea informației textuale printr-o metodă complexă neliniară în vederea facilitării explorării rapide a unei cantități mari de cunoștințe”.

Modelul hipertext specifică structura și navigarea site-ului web. Structura descrie paginile și unitățile de conținut din care acestea sînt generate. Navigarea site-ului are loc prin intermediul legăturilor. Legăturile pot fi stabilite între unități de conținut din aceeași pagină, din pagini diferite, sau în general dintre pagini. Textul din jurul legăturilor este numit context de navigare. Dacă acest text descrie conținutul paginii la care indică legătura, atunci aceste legături sînt numite contextuale.

Sistemele CMS nu dispun de modele hipertext flexibile care să-l ajute pe proiectant la automatizarea procesului de elaborare a structurii site-ului și legăturilor dintre pagini. Astfel de modele există [26,27], dar nu sînt utilizate în practică la proiectarea de CMS-uri fiindcă sînt aplicabile doar la site-uri generale. O excepție este aplicația WebRatio (www.webratio.com), care este un instrument CASE de proiectare a site-urilor elaborat în baza rezultatelor publicate în [27]. Sistemul permite generarea automată a codului pentru aplicațiile web din modele conceptuale. WebRatio generează o carcasă care trebuie finisată pînă la o aplicație finală de către programatori.

Sistemul NeoSite se deosebește de CMS existente prin elaborarea în paralel al modelului hipertext și modelului date. Nodurilor cu sub-arbori le corespund pagini de navigare spre conținutul sub-arborelui. Nodurilor frunză (*engl. leaf nodes*) le corespund pagini de conținut. Astfel, nodurile tată al unui nod în modelul hipertext reprezintă consecutivitatea de pagini accesate de utilizator pentru a ajunge la pagina de conținut. Relațiile nodului frunză în pagina HTML (fig. 4) sînt reprezentate de linkuri spre paginile nodurilor de legătură. Obținem că relațiile unui nod aparțin mulțimii de legături contextuale din pagina nodului, nodurile părinți fiind linkurile de navigare verticală în site.

Sistemul NeoSite este un mediu de elaborare a site-urilor web, iar elaborarea de site-uri specializate este imposibilă prin aplicarea unor modele generale, prețul pentru adaptarea la condiții reale a unor principii generale este prea mare.

2.6. Arhitectura sistemului

La elaborarea aplicațiilor web sînt folosite diverse protocoale de comunicare, limbaje, sisteme soft și chiar sisteme de operare diferite, care deseori sînt greu de combinat. Proiectarea arhitecturii este un proces complex din cauza neomogenității tehnologiilor folosite. Obiectivele urmărite de proiectant sînt performanța, scalabilitatea, securitatea. Trebuie luate în considerație și restricțiile de timp, resurse umane disponibile, costul soluției elaborate și a softului utilizat la elaborarea acestuia etc. Sistemul NeoSite a fost proiectat după următoarele criterii:

- costul soluției să fie minim. Toate componentele sistemului, cu excepția aplicației client care a fost elaborată în Delphi, au fost implementate folosind soft gratuit: baza de date MySQL, serverul web Apache, limbajul de programare Perl;
- sistemul trebuie să fie o soluție independentă. Proiectantul (programatorul) trebuie scutit de utilizarea unor instrumente străine, sistemul va oferi un set complet de instrumente;
- sistemul trebuie să fie flexibil. Arhitectura sistemului va permite integrarea de funcționalități noi.

Sistemul este alcătuit din patru componente de bază:

- Aplicația client;
- Baza de date;
- Scripturi perl și module mod_perl;
- Sistemul de șabloane.

Proiectanții, autorii de conținut, administratorii utilizează aplicația client și browserul pentru a lucra cu sistemul. În fig. 10 este dată arhitectura generală a sistemului NeoSite (stînga), și procesul de generare a paginilor web (dreapta).

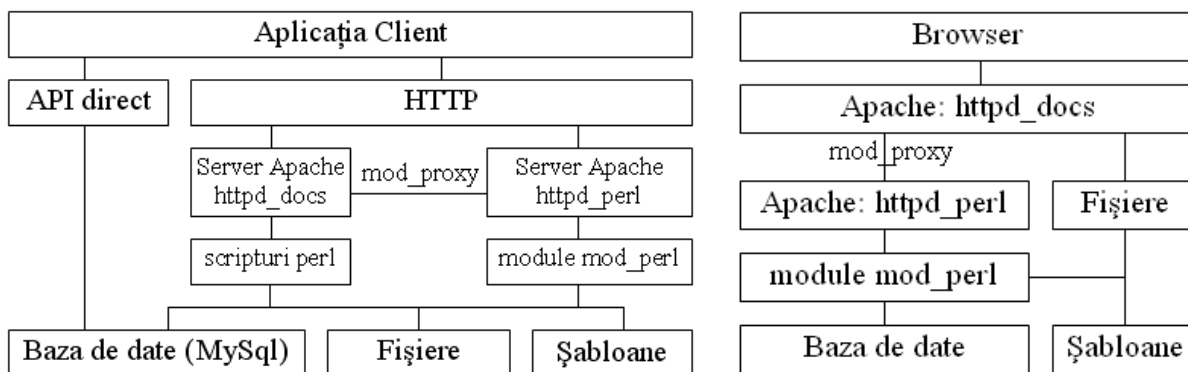


Figura 10. Arhitectura sistemului NeoSite

Cu ajutorul aplicației client este gestionat conținutul din baza de date după modelul descris în §2.4. Logica aplicației web este implementată de partea serverului în scripturi. Aplicația client prin intermediul protocolului HTTP invocă serviciile serverului. Paginile site-ului sînt generate prin aplicarea unui set de șabloane peste conținutul din baza de date furnizat de scripturi și module Apache.

Companiile producătoare de CMS doresc o acoperire cît mai mare a pieței. Sînt dezvoltate produse care să fie utilizate și administrate de utilizatori simpli fără cunoștințe de programare. Spre deosebire de sistemele cu cod deschis, arhitectura sistemelor comerciale nu este vizibilă, în sensul că nu presupune modificări din partea clientului. La prima etapă cerințele simple ale consumatorului care nu are experiență în evaluarea unui CMS sînt acoperite de posibilitățile de bază ale sistemului. Curînd clientul conștientizează potențialul și impactul pozitiv pe care îl aduce un CMS și dorește integrarea și automatizarea de alte procese în cadrul soluției existente. Arhitectura rigidă specifică de obicei produselor de larg consum impune consumatorul să treacă la alt sistem mai complex sau să elaboreze un sistem propriu care să-i satisfacă necesitățile.

O trăsătură specifică al CMS-urilor este modul de formare al adreselor URL pentru accesarea conținutului. Adrese de tipul

http://www.asm.md/noutcur.shtml?AA_SL_Session=aa86622066af96f345ef6d0cc30f7ed9&x=1923
http://www.mamboserver.com/index.php?option=com_content&task=view&id=35&Itemid=116

sînt incomode pentru vizitatorii site-ului care doresc să-și noteze adresa paginii, totodată ele nu oferă contextul necesar. Cu alte cuvinte adresa paginii nu este sugestivă ca cineva să-și deie seama de conținutul ei fără s-o acceseze. În afară de oameni site-urile sînt accesate și de motoarele de căutare care acordă o importanță deosebită cuvintelor din adresele paginilor la evaluarea relevanței paginii în baza cuvintelor cheie din interpelarea către sistemele de căutare.

Sistemul NeoSite propune o schemă ierarhică de construire al adreselor paginilor care reflectă modelul de date propus în §2.4. În fig.5 coloana *filename* al tabelului TREE este identificatorul nodului folosit în modelul hipertext. Astfel, adresa URL al nodului de nivelul 3 va fi:

/p1/p2/filename/

unde p1, p2 sînt identificatorii *filename* ai nodurilor părinți. Pentru a găsi un nod de nivelul N sînt necesare N interpelări SQL. Coloana *filename* este indexată în tabelul TREE.

Existența căilor alternative de navigare a site-ului înseamnă că numărul paginilor unui site dinamic este mai mare decît numărul nodurilor din arborele de conținut. Spre exemplu la data de 19 februarie 2006 tabelul TREE al site-ului www.cnaa.acad.md avea 3650 de noduri, iar numărul de pagini indexate de motorul Google era de 30800 (căutați în google după „site:cnaa.acad.md”). Fiecărei pagini i se pune în corespondență un șablon HTML. Acest proces necesită implicarea programatorului.

Sistemul NeoSite utilizează o combinație de două servere Apache notate în fig.10 prin *httpd_docs* și *httpd_perl* care comunică prin intermediul modulului *mod_proxy*. Toate adresările către un site web le primește primul server *httpd_docs*. Cu ajutorul modulului *mod_rewrite* acest server determină dacă este accesată o pagină dinamică sau statică. Conținutul static (imagini, documente word, pdf) este deservit de *httpd_docs*. Accesul către paginile dinamice sînt direcționate de *httpd_docs* către serverul *httpd_perl*. *httpd_perl* este un server apache compilat cu modulul *mod_perl*, el generează pagini html pe care le întoarce prin *mod_proxy* primului server ca acesta în final s-o transmită browserului. În fig.11 este dată configurația succintă al unui site pentru serverele Apache.

httpd docs	httpd perl
<pre><VirtualHost 81.180.65.122> ServerName cnaa.acad.md RewriteEngine On RewriteRule \.(gif jpg pdf doc)\$ - [last] RewriteRule ^/(.*) http://cnaa.acad.md:8080/\$1 [P] ProxyPassReverse / http://cnaa.acad.md/ </VirtualHost></pre>	<pre><VirtualHost 81.180.65.122> ServerName cnaa.acad.md <Location /> PerlModule CNAA::Main SetHandler perl-script PerlHandler CNAA::Main </Location> </VirtualHost></pre>

Figura 11. Configurarea serverelor apache pentru sistemul NeoSite.

2.7. Aplicația client al sistemului

Interfețele Web de interacțiune cu sistemele de administrare a conținutului sînt cele mai populare astăzi. Avantajele lor sînt: comoditate și simplitate în utilizare, excluderea necesității de instalare a unor softuri de partea beneficiarului. Totuși, browserele sînt extrem de limitate în planul elementelor de interfață oferite în comparație cu aplicațiile obișnuite. Interfețele web nu permit managementul unor volume mari de informație. Aplicațiile web sub formă de java appleturi n-au avut succes, iar tehnologia AJAX (<http://en.wikipedia.org/wiki/AJAX>), ce apăruse odată cu introducerea de către Microsoft a obiectului XMLHttpRequest în 2002 pentru elaborarea aplicațiilor web interactive, necesită eforturi considerabile pentru crearea unei aplicații similare după posibilități cu cea de tip desktop.

În plus, o interfață trebuie să funcționeze normal și în condiții nefavorabile (legătura proastă cu Internetul, transferul rapid și procesarea complexă a datelor). Obiectiv greu de atins pentru interfețele Web. Sistemul presupune o dinamică ridicată de redactare, adăugare și interconectare a informației. Aceste activități necesită interfețe mai rapide și democratice de interacțiune. De aceea s-a optat pentru o aplicație client de administrare.

Aplicația client a fost implementată în mediul Borland Delphi. Flexibilitatea și fiabilitatea sînt asigurate de arhitectura modulară a programului. Modulele se pot realiza în paralel și relativ independent de către mai mulți programatori ce duce la creșterea eficienței de programare. Un modul îndeplinește un set de sarcini pentru realizarea cărora poate apela la alte module. Modulul este o bibliotecă DLL (*engl. Dynamic Loading Library*) încărcată/descărcată de programul principal în funcție de necesități. Toată funcționalitatea este concentrată în module, programul principal are funcția de moderator al modulelor și răspunde de reglementarea situațiilor excepționale inițiate de module. În anexa 6 este ilustrat modulul *PluginManager* care vizualizează toate modulele încărcate în sistemul NeoSite și oferă posibilitatea de vizualizare desfășurată a proprietăților modulelor. În interfața sistemului prezentată în fig. 12 sînt marcate elementele de bază, discutate mai jos. Mai multe imagini ale sistemului le găsiți pe pagina oficială al sistemului NeoSite: www.cms.neonet.md.

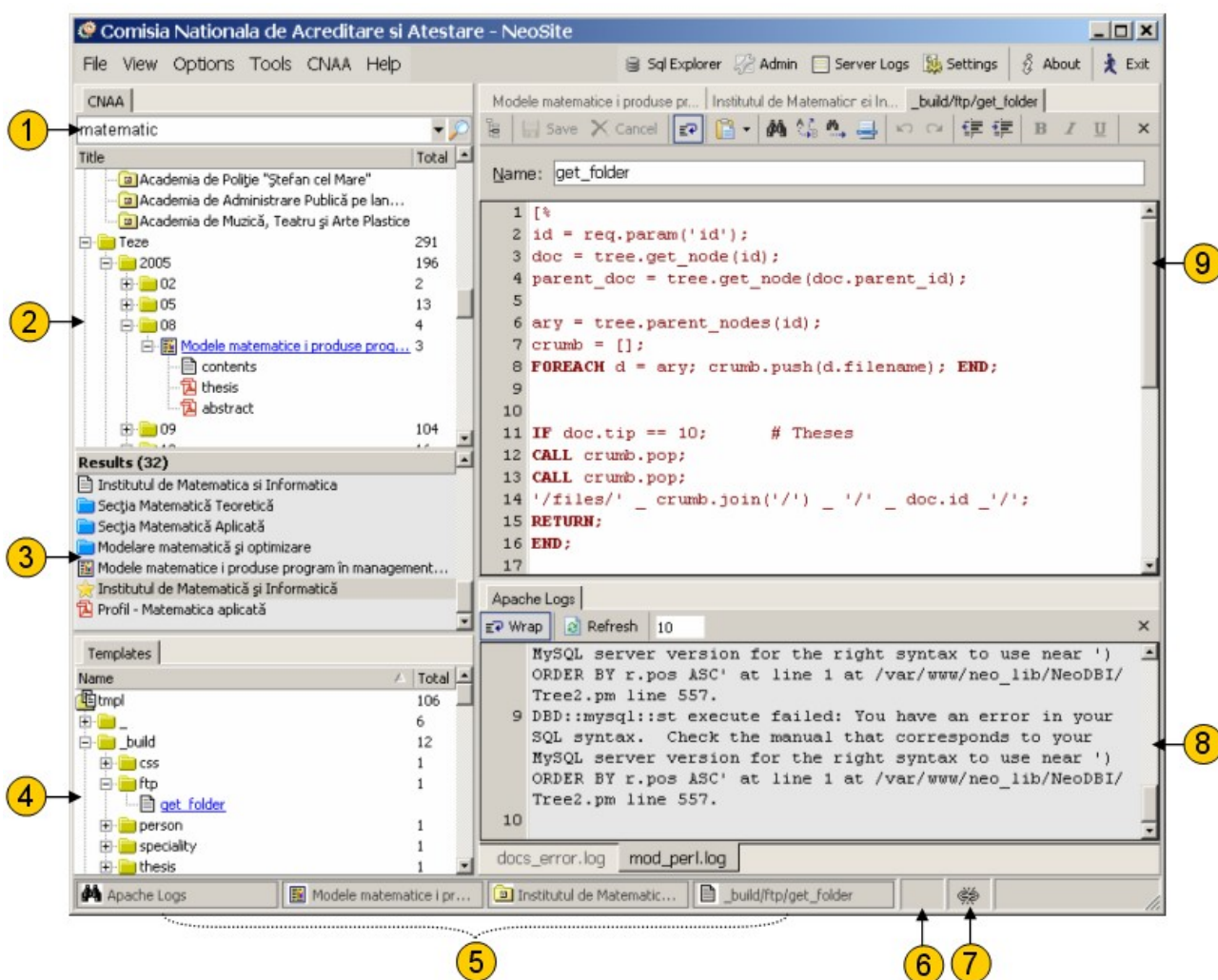


Figura 12. Interfața sistemului NeoSite.

Obiectivul principal la etapa de proiectare a fost flexibilitatea. Configurația modulelor abstracte din care este compus sistemul este specificată prin documente XML păstrate în baza de date. Pentru fiecare proiect web este creată configurația sa specifică. Funcționalitatea (logica) aplicației este implementată de partea serverului, iar aplicația client asigură comunicarea: invocarea serviciilor de partea serverului.

Modulele de bază sînt:

- *SiteTree*, manipularea (vizualizare, redactare) arborelui de conținut;
- *Srv*, oferă servicii și resurse altor module: asigurarea conexiunii cu baza de date, colecții de pictograme grafice, etc.;
- *EdGeneric*, managementul unui nod de conținut din arborele principal, acest modul este utilizat de modulul *SiteTree*;
- *EdSql*, managementul unui set de noduri (interpelări SQL des folosite pot fi păstrate în noduri speciale de conținut. Redactarea unui astfel de nod de fapt înseamnă executarea interpelării SQL. Fereastra modulului *EdSql* este ilustrată în Anexa 1);
- *CGI*, răspunde de comunicarea prin protocolul HTTP cu serverul web. Este utilizat de alte module (În Anexa 2 este ilustrată fereastra modulului CGI în care este afișat rezultatul executării scriptului de construire a șabloanelor site-ului);
- *SQL*, executarea interpelărilor SQL. Este folosit de către persoanele ce cunosc limbajul SQL sau alte module pentru vizualizarea rezultatelor unor interpelări construite automat. (În Anexa 3 este ilustrat rezultatul unei interpelări SQL cu posibilitatea grupării după cîmpuri);
- *TmplTree*, managementul șabloanelor proiectului. Este similar modulului *SiteTree*;
- *EdTmpl*, fereastra de redactare a șabloanelor HTML;
- *FTP*, răspunde de transmiterea/descărcarea fișierelor de pe serverul FTP al site-ului web;
- *Admin*, managementul configurației modulelor: de exemplu, structura meniului principal al programului și definiția acțiunilor respective. (În Anexa 4 este ilustrată fereastra modulului *Admin* în care sunt afișați parametrii modulului *SiteTree*).

Modulul SiteTree:

În tabelele bazei de date TREE și TREE_RELATED după modelul de date descris în §2.4 este stocat conținutul site-ului în formă de arbore (punctul 2 din fig. 12) și a relațiilor dintre nodurile acestuia. Modulul *SiteTree* răspunde de managementul conținutului site-ului, relațiile sînt gestionate de către modulul *EdGeneric*. Modulul este reprezentat de punctele 1,2,3 din fig.12. Căutarea nodurilor (punct 1, fig.12) în arborele de conținut după anumite criterii (denumirea nodului, conținut, sau condiție SQL) și afișarea nodurilor găsite (punct 3, fig.12). Există posibilitatea de redactare simultană a cîtorva noduri (punct 5, fig.12). La redactarea unui nod, *SiteTree* încarcă o instanță a modulului *EdGeneric* cu parametrii: denumirea tabelului de conținut și identificatorul nodului respectiv. În anexa 5 este ilustrată secțiunea de redactare a cîmpurilor textuale ale nodului de conținut: cîmpurile în care sînt păstrate documente XML de exemplu.

Conținutul site-ului este stocat într-un singur tabel TREE. Datele pentru construcția paginii html sînt obținute din mai multe interpelări SQL. Procesul poate fi optimizat prin introducerea de tabele ajutătoare pentru stocarea informației des utilizate din tabelul TREE sau rezultatul unor calcule care sînt des folosite la generarea paginilor și trebuie efectuate doar la modificarea informației din tabelul TREE și nu de fiecare dată la generarea paginilor.

Numărul de noduri din tabelul TREE poate fi mare. *SiteTree* extrage din tabel doar nodurile cerute de utilizator: în momentul cînd modulul *SiteTree* este încărcat de aplicația principală, sînt extrase nodurile primului nivel. La un click pe semnul [+] sau dublu click pe nod vor fi extrase și prezentate

doar nodurile copii ale nodului în cauză. În acest mod performanța modulului nu depinde de numărul de noduri al arborelui. La încheierea sesiunii de lucru, *SiteTree* înregistrează nodul activ și nodurile „deschise” de utilizator pentru ca la următoarea sesiune să restabilească imaginea anterioară al arborelui.

În anexa 8 este ilustrat meniul cu entități al proiectului. Anexa 9 reprezintă fereastra de management al entităților.

Modulul EdGeneric:

Interfața de management al nodului de conținut este generată în baza definiției entității (fig. 9). Atributelor entității (secțiunea */entity/fields/*) le sînt asociate o serie de editoare vizuale în dependență de tipul coloanei tabelului bazei de date. Fereastra de management este compusă din trei părți (§3.3.4, fig. 21): *memos*, *data*, *related* și înseamnă respectiv cîmpuri text, cîmpuri simple (*varchar*, *integer*) și relațiile nodului.

De obicei la crearea nodurilor sau modificarea conținutului acestora trebuie efectuate anumite acțiuni: sincronizarea cu alte tabele ale bazei de date, înștiințarea autorului nodului despre modificarea acestuia etc. În sistemele bazelor de date această necesitate a fost acoperită prin introducerea triggerilor. În cazul unui sistem abstract aceste proceduri trebuie implementate în afara sistemului. Posibilitatea de integrare în sistemul NeoSite a funcționalității specifice unui proiect web este realizată prin introducerea modelului de evenimente. Ideea a fost preluată din mediile de programare rapidă (Delphi, C Builder). La efectuarea anumitor operații de către utilizator(sistem) sînt generate o serie de evenimente de care pot fi atașate proceduri implementate în exterior. În definiția entității din fig. 9 a fost introdusă secțiunea *events*:

```
<events>
  <event_name>
    <e type="[sql, http, ...]" /> <params> ... </params> </e>
  </event_name>
  <OnCreate>
    <e type="sql">INSERT INTO site_related (from_id, to_id, from_tip, to_tip)
      VALUES ($id, 73, 10, 1)
    </e>
    <e type="http">
      <params>
        <url>$builder_script/thesis/on_save/</url>
        <prms>id=$id</prms>
      </params>
    </e>
  </OnCreate>
</events>
```

Figura 13. Evenimentele de sistem asociate unei entități

În modulul *EdGeneric* sînt generate următoarele evenimente:

OnCreate - la crearea unui nod, *OnBeforeSave* - înainte de salvarea modificărilor în baza de date, *OnAfterSave*, *OnRelateNode* - la stabilirea relațiilor dintre noduri.

În exemplul din fig. 13, la crearea unui nod sînt efectuate două acțiuni: o comandă SQL, care stabilește o relație dintre nodul nou creat și nodul cu identificatorul 73, și o comandă de tip HTTP care semnifică invocarea unui script de pe un server web cu adresa *url* și parametrii *prms*.

EdGeneric nu “știe” ce înseamnă SQL sau HTTP și folosește serviciile oferite de modulele ajutatoare *Srv* și *CGI*.

Pe lângă evenimentele de sistem (*OnCreate*, *OnSave*) sînt și evenimente definite de utilizator. În definiția entității este descris meniul ferestrei de redactare a nodurilor (fig. 15) care este creat de *EdGeneric* la momentul încărcării sale.

```

<menu>
  <sm caption="Workflow" ico="71">
    <li caption="Notificare" event="EmailEditor" ico="33" />
    <li caption="Sustinere" event="ActivateThesis" />
    <separator />
    <li caption="Arhivare" event="ArchiveThesis" ico="12" />
  </sm>
  <sm caption="Comentarii" ico="20">
    <li caption="Aprobate" event="EditComments" />
  </sm>
</menu>

```

Figura 14. Evenimentele entității definite de utilizator

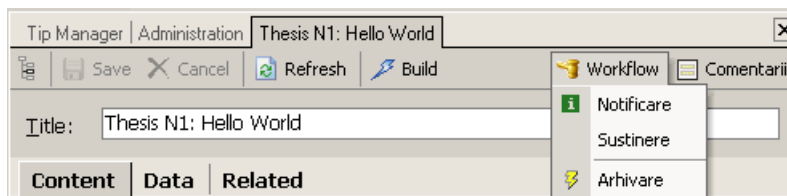


Figura 15. Meniul ferestrei de redactare generat din definiția XML a entității

În definiția entității este indicat denumirea modulului care este încărcat pentru managementul nodului. *EdGeneric* acoperă necesitățile de bază, dar putem folosi și alte module cu condiția că ele au aceeași interfață de comunicare cu modulul *SiteTree*.

Soluții pentru generarea interfețelor de management al conținutului din bazele de date există [40,41], definirea entităților în format XML din [42] este asemănătoare cu cea din NeoSite cu excepția modelului de evenimente. În [27] întâlnim un model de evenimente: în baza unui model conceptual este generat scheletul codului sursă al aplicației web în care programatorul implementează evenimentele necesare.

Modulul EdSql:

Una din problemele CMS-urilor actuale este lipsa metodelor alternative de navigare a conținutului site-ului. Majoritatea CMS-urilor oferă facilități de căutare a informației după anumite criterii. Utilizatorul trebuie scutit de formularea aceleiași cereri de fiecare dată cind are nevoie de careva informație. *EdSql* este modulul pentru navigarea și redactarea colecției de noduri obținut printr-o interpelare SQL.

Modulele *TmplTree* (punct 4, fig. 11) și *EdTmpl* (punct 9, fig. 11) asigură managementul șabloanelor HTML și sînt similare modulelor *SiteTree* și *EdGeneric*. Modulele SQL și FTP realizează funcțiile întâlnite în softurile standarde de lucru cu bazele de date și serverul FTP. Ele au fost elaborate pentru integrarea cu alte module și scutirea programatorului de utilizarea unor programe separate. Există module pentru crearea copiilor de rezervă a bazei de date și restabilirea acesteia. Pentru programatori foarte util este modulul *LogMon* (punct 8, fig. 11) de monitorizare a modificărilor recente a fișierelor de pe server. *LogMon* utilizează modulul CGI care execută un script de pe server avînd ca parametru numele fișierului monitorizat. Rezultatul sînt ultimele N rînduri ale fișierului dat. Modulul este folosit pentru monitorizarea erorilor generate de server. Arhitectura sistemului permite adăugarea de noi module, deaceea numărul modulelor standarte din pachetul NeoSite este în creștere.

2.8. Procesul de generare a site-ului

CMS-ul generează pagini folosind un set de șabloane. Construirea paginii dinamice constă din trei etape:

1. determinarea șablonului după adresa URL a paginii și parametrilor transmiși;
2. extragerea informației din baza de date necesare pentru generarea paginii;
3. procesarea șablonului.

În sistemele simple programatorul nu poate face schimbări în primele două etape, iar posibilitățile limitate ale sistemului de șabloane nu permite includerea de module noi pentru extragerea și procesarea informației. Sistemele complexe sînt orientate pentru utilizarea de către un grup de programatori, deaceea se impune o “distanță” și chiar independență dintre aceste etape. Ambele abordări sînt corecte în dependență de problema pusă. Sistemele simple sînt bune pentru elaborarea rapidă, ad-hoc a site-urilor web dotate cu funcționalități bogate dar standarde. Acest tip de site-uri nu pot evoluționa în sensul integrării de funcționalități noi. Site-urile dezvoltate cu CMS complexe pot evoluționa, însă ele din start trebuie să fie complicate pentru a justifica cheltuielile mari pentru elaborarea structurii site-ului, programarea de module specializate. CMS-uri pentru elaborarea de site-uri simple dar care pot deveni complexe în viitor nu sînt rentabile comercial. Realizările din mediul academic [26,27] sînt exotice pentru aplicații practice fiindcă presupun un anumit nivel de cunoștințe și impun deprinderea unor procedee „incompatibile” cu modul existent de elaborare a site-urilor. Puține din aceste realizări [27] reușesc să fie utilizate în produse comerciale: www.webratio.com.

În sistemul NeoSite cele trei etape sînt controlate și de programator, și de utilizator (persoana care lucrează cu șabloanele). Arhitectura componentei server se aseamănă cu cea al aplicației client, procesul de generare al paginilor este ilustrat în fig.16:

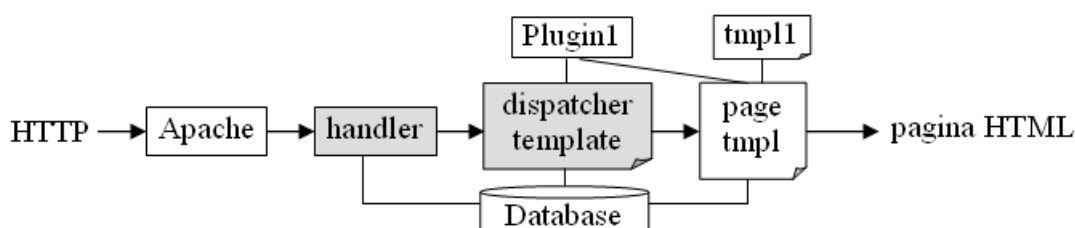


Figura 16. Procesul de generare al paginilor.

Handler este un modul perl integrat cu serverul Apache. Handlerul este indicat în configurația serverului `httpd_perl` din fig.11: `PerlHandler CNAA::Main`. Dacă ne conducem de modelul MVC (engl. *Model-View-Controller*), logica aplicației trebuie implementată separat de partea de prezentare, deaceea determinarea șablonului este efectuată de *handler*, care procesează șablonul *page_tmpl*. Acesta este modelul clasic utilizat de CMS. Utilizatorul nu are posibilitate să schimbe modul în care sînt construite adresele URL ale paginilor. În NeoSite șablonul poate fi determinat și de către utilizator: *handler-ul* după ce stabilește conexiunea cu baza de date transmite controlul unui șablon artificial *dispatcher_template* care stabilește șablonul de bază din care este generată pagina. Logica aplicației web este implementată în module separate care sînt încărcate din șabloane (*plugin1* din fig. 16). Șabloanele mari pot fi divizate în unele mai mici care ulterior pot fi reutilizate în cadrul altor șabloane.

Modulul *handler* și șablonul *dispatcher* sînt similare programului principal al aplicației client: ele au funcția de coordonator și nu conțin funcționalitate.

Sistemul de șabloane utilizat este Template-Toolkit (www.tt2.org) care oferă un limbaj de descriere a șabloanelor și un set de module perl pentru compilarea și procesarea lor. Unele sisteme: Strudel [26], Zope [35] de exemplu, propun limbaje proprii pentru descrierea șabloanelor. Sistemul de șabloane Zope este foarte asemănător cu Template-Toolkit, însă cel din urmă e mai compact, mai bine documentat [43], mai acceptat (este folosit de mai multe CMS).

La proiectarea șablonelor problema principală este modul în care informația necesară este extrasă din baza de date. În sistemele simple datele necesare sînt extrase de program și transmise șablonului pentru procesare. La generarea unei pagini complexe nu se cunoaște din start informația necesară: de exemplu, conținutul paginii depinde de profilul utilizatorului care o accesează. În acest caz, interpelările SQL sînt descrise nemijlocit în interiorul șablonelor ce contravine principiului MVC și crează probleme de mentenanță. Sistemul de șabloane oferă posibilitatea de transferare a funcționalității în module, însă lipsa modelelor de date impune crearea/modificarea frecventă al acestor module.

Modelul simplu de date implementat în NeoSite a permis elaborarea unui modul de extragere a datelor care acoperă 90% din necesități. Modulul *Tree.pm* oferă proceduri de extragere/modificare a arborelui de conținut. Procedurile cele mai frecvent utilizate sînt:

<code>find_node('/p1/p2/filename/')</code>	întoarce nodul cu identificatorul filename părinții caruia sînt p1 și p2
<code>search_nodes(where_sql)</code>	nodurile care satisfac condiția SQL indicată
<code>parent_nodes(id)</code>	nodurile părinți
<code>first_child(id, where_sql)</code>	primul nod copil al nodului 'id' care satisface condiția 'where_sql'
<code>last_child(id, where_sql)</code>	similar first_child
<code>next_sibling(id, where_sql)</code>	nodul frate vecin (care au același părint și de același nivel)
<code>child_nodes(id, where_sql)</code>	nodurile copii ale nodului 'id'
<code>descendant_nodes(id, where_sql)</code>	procedură recursivă de extragere a subarborelui nodului 'id'.
<code>related_nodes(id, where_sql, direction = [to, from])</code>	relațiile nodului 'id'
<code>relate_node(from_id, to_id)</code>	stabilirea relației: from_id: nod sursă to_id: nod destinație

În șablonul *dispatcher* cu ajutorul funcției *find_node* este stabilit nodul și șablonul *page_tmpl* care trebuie procesat. *find_node* întoarce nodul de nivel maxim găsit. Presupunem că nodul '/p1/p2/filename/' este nod frunză și este accesată pagina cu adresa '/p1/p2/filename/n1/n2'. În acest caz controlul se transmite șablonului nodului '/p1/p2/filename/' cu parametrul 'n1/n2'. Astfel, șablonul folosit la generarea paginilor poate fi determinat după procesarea mai multor șabloane de coordonare. Apare posibilitatea integrării într-un site existent al unor sub-site-uri: de exemplu, adresa 'n1/n2' poate fi identificatorul unui nod din alt arbore de conținut.

În modelul de site descris în §2.3 s-au identificat două tipuri de pagini: de navigare și de conținut. În șabloanele paginilor de navigare se folosesc funcțiile *child_nodes*: pentru a construi legături către paginile interioare ale site-ului și *search_nodes*: pentru căi alternative de navigare. În exemplul de mai jos este încărcat modulul *Tree* (*dbh* este conexiunea către baza de date), după adresa indicată în browser (*REQUEST_URI*) este determinat nodul arborelui (*doc*) și construite o serie de legături: fig. 17.

Majoritatea legăturilor din interiorul paginii de conținut sînt relațiile nodului respectiv în care acesta figurează ca nod sursă. Pentru navigarea bidirecțională pot fi extrase relațiile în care nodul paginii este nod destinație.

```

[% USE tree = Tree2({DBH => dbh, TABLE => 'site',
                    COLUMNS => 'id, filename, title'}) %]
[% doc = tree.find_node(REQUEST_URI) %]
<ul> <!-- navigare verticala: linkuri catre pagini interioare -->
[% FOREACH d = tree.child_nodes(doc.id, {where_sql => 'approved=1'})%]
<li><a href="[% d.filename %]"/>[% d.title %]</a>
[% END %]
</ul>

<!-- navigarea pe orizontala: noduri de acelasi nivel -->
[% d1 = tree.prev_sibling(doc.id); d2 = tree.next_sibling(doc.id) %]
Capitolul precedent: <a href="[% d1.filename %]"/>[% d1.title %]</a>
Capitolul urmator: <a href="[% d2.filename %]"/>[% d2.title %]</a>

```

Figura 17. Model șablon al paginii de navigare

2.8.1. Implementarea modelului hipertext

În §2.5 s-a menționat că sistemul NeoSite se deosebește de alte CMS prin elaborarea în paralel al modelului hipertext și modelului date iar elaborarea de site-uri specializate este imposibilă prin aplicarea unor modele generale.

În [27] se propune asamblarea paginilor din elemente primitive. Această abordare este foarte productivă când se dorește elaborarea unui site în termeni restrânși. Scopul acestor site-uri industriale este de a oferi vizitatorilor posibilitatea de a găsi informația necesară utilizând instrumente standarde. Pentru ajustarea elementelor de bază ale sistemului este nevoie de intervenții în codul sursă. Pentru un site ce conține multe elemente specifice ajustarea de obicei este mai costisitoare decât implementarea de la zero. Utilizarea limbajului CSS pentru ajustarea vizuală a elementelor facilitează separarea prezentării de funcționalitate. Din păcate ajustarea funcțională sau combinarea de elemente poate fi efectuată prin intervenții în codul sursă.

O singură pagină din cadrul site-ului nu reflectă integral conținutul acesteia. Importante sînt legăturile către alte pagini similare. Conținutul paginii poate fi divizat în două tipuri: conținutul nodului propriu zis și informație complementară. Din cauza separării modelului date de modelul hipertext CMS-urile nu pot evidenția conținutul și legăturile care pot fi utilizate la construirea acestei părți. Aceasta implică utilizarea unui număr mare de SQL interpelări neomogene pentru generarea paginii. Modificări în modelul date necesită ajustarea sau chiar rescrierea interpelărilor scrise de programatori dar nu generate semiautomat de sistem.

Dependența modelelor date și hipertext în NeoSite simplifică procedura de selectare a legăturilor care trebuie incluse într-o pagină. În §2.5 se spune că relațiile unui nod aparțin mulțimii de legături contextuale din pagina nodului. Cu cît nivelul nodurilor de legătură din arborele de relații al nodului este mai mic, cu atît legătura către aceste noduri este mai importantă. Relațiile nodului sînt extrase folosind o singură funcție din modulul *Tree.pm*. Cu ajutorul funcției *related_nodes* putem lucra cu relații de orice nivel. Stocarea tuturor relațiilor într-un singur tabel ierarhic, definiția (structura) căruia nu se schimbă la introducerea de noi tipuri de relații permite crearea unor șabloane stabile și reutilizabile.

2.8.2. Servicii web

Informația ce apare pe site poate fi folosită și de alte sisteme: motoare de căutare, agenți pentru monitorizarea paginilor și înștiințarea utilizatorului despre modificările care au avut loc. După definiția W3C (*engl. World Wide Web Consortium*) serviciile web sînt sisteme software proiectate pentru asigurarea interacțiunii a două calculatoare într-o rețea. În prealabil este stabilită o interfață de comunicare, de obicei descrisă cu ajutorul limbajului XML. Arhitectura CMS-ului trebuie să permită integrarea de servicii web din afara sistemului pentru asigurarea flexibilității întregului

sistem. Totodată CMS-ul are rolul de mediator între serviciile web și utilizatori. Posibilitatea de determinare a șablonului în dependență de adresa URL al paginii accesate a permis sistemului NeoSite să facă față cerințelor de mai sus.

Necesitatea integrării de servicii web a fost condiționată de transferul funcționalității din aplicația client al sistemului NeoSite pe server. Flexibilitatea unui sistem prea general necesită timp pentru adaptare la o problemă concretă. Funcționalitatea aplicației client NeoSite este modelată de documente XML în care sistemul este ajustat pentru fiecare proiect în parte. Imposibilitatea ajustării unei funcționalități abstracte a determinat implementarea procedurilor specifice de partea serverului, și trebuia elaborat un mecanism de comunicare între aplicație și server:

- procedurile au fost “învelite” în module încărcate din șabloane;
- Au fost rezervate o serie de adrese URL șabloanele cărora transmit modulelor parametrii indicați în URL;
- Rezultatul întors de module este procesat în șablon și întors aplicației client.

Rezervarea adreselor URL este realizată în șablonul *dispatcher_template* sau chiar în modulul apache *handler*: toate adresele de forma “/system/*” sau “/build/*” sînt procesate de același șablon, care încarcă un modul și execută procedura cu numele ce urmează după “/system/” în URL. Uneori posibilitățile sistemului de șabloane este suficient pentru a construi răspunsul.

Un exemplu de utilizare a schemei propuse este determinarea mapei în care vor fi păstrate fișierele unui nod de conținut. Cînd utilizatorul dorește să atașeze un fișier de nodul cu identificatorul *ID*, aplicația client prin intermediul modulului CGI „acesează” pagina

http://server/system/ftp_upload_dir/?item=ID pentru care este folosit un șablon de forma:

```
id = req.param('item');
doc = tree.get_node(id);
ary = tree.parent_nodes(id);
crumb = [];
FOREACH d = ary; crumb.push(d.filename); END;
IF doc.tip == 10;
  '/files/' _ crumb.join('/') _ '/' _ doc.id _ '/';
END;
```

Rezultatul procesării acestui șablon este denumirea mapei de pe server în care prin protocolul FTP va fi transmis un fișier.

2.9. Sistemul de management al conținutului NeoSite

Sistemul NeoSite este rezultatul a patru ani de dezvoltare și restructurare continuă. Implementarea mai multor proiecte complexe reprezintă aplicarea practică a conceptelor și modelelor de elaborare a site-urilor web propuse în teză.

NeoSite este o aplicație multi-utilizator client/server de management al conținutului unui site web. Sistemul promovează o buna organizare a informației și automatizează mentenanța site-urilor. Este destinat pentru elaborarea site-urilor organizațiilor mici și mijlocii. Utilizatorii de bază ale sistemului vor fi furnizorii de conținut (*engl. content providers*), autori și editori de conținut. Dacă informația este bine structurată și are un grad avansat de granularitate cunoașterea limbajului HTML este opțională.

Sistemul nu este o soluție integră și independentă de implementare a site-urilor. Sistemul reprezintă o carcasă peste care este construită o aplicație unică în baza necesităților specifice fiecărui site.

Motto-ul sistemului: Sistem flexibil pe bază de module ce oferă interfețe simple și comode pentru administrarea eficientă și efectivă a informației.

Particularități semnificative ale NeoSite:

- administrarea conținutului pe bază de unități mici, interconectate de informație (noduri de conținut);
- nodul este definit prin descrierea proprietăților sale și a relațiilor cu alte tipuri de noduri;
- grad avansat de interconectare între noduri;
- interfețe eficiente de management al conținutului;
- facilitează activitatea editorială (șabloane, scripturi);
- nu necesită editoare sau vizualizatoare speciale;
- separare de conținut, design și funcționalitate.

Obiectivele stabilite pot fi realizate numai printr-o concepție scalabilă, flexibilă, modulară. La elaborarea unui sistem care să poată fi cât mai ușor de întreținut și dezvoltat toate etapele sînt importante: de la structurarea informației pînă la implementarea unor soluții de generare dinamică a paginilor. Cu cît concepția inițială este mai riguroasă și, în același timp, mai flexibilă, cu atît mai simplă va fi întreținerea și dezvoltarea ulterioară.

Modelarea strictă a lumii reale va duce la un sistem care reflectă realitatea curentă, dar nu neaparat și pe cea viitoare. Abstracțiunile identificate în timpul proiectării sînt esențiale în obținerea unui sistem flexibil.

Activitatea majorității softurilor, în principiu, poate fi împărțită în trei etape: obținerea datelor inițiale (intrările sistemului), procesarea lor și prezentarea rezultatului final (ieșirile sistemului). Aceste etape coincid cu componentele arhitecturii (paradigmei) Model/View/Controller (MVC): modelul poate fi văzut ca o structură logică de date, oferite inițial de intrările sistemului. Controlerul reprezintă etapa de procesare în care modelul este manipulat sau modificat. Ieșirile sistemului (View) sînt vizualizările modelului (datele) într-un anumit format (de ex. pagini HTML).

Paradigma MVC ia naștere în sistemul Smalltalk-80 pentru a promova o abordare stratificată de dezvoltare a interfețelor grafice cu utilizatorul. Astăzi arhitectura MVC este utilizată inclusiv și pentru dezvoltarea structurată a sistemelor mari, în special al aplicațiilor Web [42].

Paradigma MVC este unul din conceptele principale ale arhitecturii NeoSite ce promovează diferențierea clară între date, prezentare și logică. Devine posibilă separarea în timp și spațiu a proceselor de gestiune a informației, proiectării șablonelor și implementării scripturilor.

Componentele sistemului NeoSite sînt baza de date, interfața utilizator, instrumente de vizualizare, redactare și căutare a informației, șabloane, scripturi de prezentare a informației în baza șablonelor. Construcția unui sit web constă din:

- selectarea, integrarea și administrarea datelor ce urmează a fi publicate pe site;
- definirea structurii site-ului: tipurile de pagini, datele folosite în fiecare pagină și legăturile dintre acestea;
- elaborarea aspectului grafic al site-ului (design).

Caracteristici tehnice:

Toate componentele sistemului, în afară de aplicația client, au fost implementate folosind soft gratuit. Serverul baze de date a fost ales MySQL: conectivitatea, viteza și securitatea au făcut serverul MySQL un standard de facto pentru soluții web și aplicații de talie medie.

Pentru aplicațiile de partea serverului s-a folosit limbajul Perl: este gratuit și posedă numeroase mijloace de documentare online. Este folosit în special pentru dezvoltarea rapidă de aplicații de

administrare a sistemului de operare și destinate Web-ului, reprezentînd un mediu ideal pentru conceperea scripturilor CGI.

Serverul web care rulează scripturile scrise în Perl este Apache, ale cărui facilități și performanțe l-au făcut cel mai popular server web de pe internet: potrivit statisticilor Netcraft (www.netcraft.com/survey) Apache ocupă 55% din piață. Mai mult, Apache beneficiază de un interpretator Perl intern, numit mod_perl, care sporește productivitatea. Aplicația client a fost implementată în mediul Borland Delphi și rulează pe orice sistem Windows (9x, NT, 2000, XP).

2.10. Concluzii

Administrarea efektivă a site-urilor web poate fi realizată folosind instrumente relativ simple și resurse limitate. În acest capitol a fost descris un model de sistem de management al conținutului destinat site-urilor structurate.

Succesul implementării unui CMS depinde de acceptarea sa de către autori, creatori, manageri și administratori. Cerințele cotidiene de integrare cu alte aplicații înseamnă tipuri noi de utilizatori. Pentru a fi acceptat, un sistem trebuie să dispună de o interfață comodă care ar spori productivitatea pentru toți utilizatorii săi. Acest punct de vedere este susținut de un studiu efectuat în 1994 numit "Raport despre haos" [44], în care se conchide că factorul cel mai important ce duce la implementarea nereușită a sistemelor este lipsa interesului (implicare redusă) din partea utilizatorilor. Proiectanții sistemelor acordă cele mai multe resurse aspectului tehnologic al sistemului și deseori nu i-au în considerație viitorii utilizatori. Cu alte cuvinte, ei sînt preocupați de *cum lucrează sistemul*, ci nu *pentru cine și pentru ce acest sistem se proiectează*.

În modelul propus de CMS datele sînt stocate și manipulate cu ajutorul unui arbore. Dacă volumul de informație este mare (arborele are multe noduri), o singură modalitate de gestiune a datelor devine insuficientă. Sistemul oferă instrumente complementare de vizualizare a informației. Cu alte cuvinte, există mai multe perspective asupra datelor. Ca urmare se obține o imagine mai amplă a conținutului site-ului.

Modelul de date propus, în care relațiile sînt reprezentate sub formă de arbore, a permis elaborarea unei interfețe generale de management al conținutului. Interfața este generată din fișiere de configurare XML, deaceia este posibil de creat și un „interpretator” pentru generarea interfeței web de administrare.

Arhitectura plug&play al aplicației client și a programelor de partea serverului permite extinderea sistemului și dotarea acestuia cu module specifice.

Sistemul NeoSite, elaborat după modelul descris, nu a fost conceput ca un produs aparte, dar succesul sistemului ne-a determinat să ne gîndim la posibilitatea poziționării sistemului NeoSite ca o platformă de dezvoltare a site-urilor. Ca un soft să devină produs trebuie scrise manuale pentru utilizatori, petrecute campanii publicitare, studiată piața și determinată nișa. Pentru a preveni utilizarea nesacționată și răspîndirea copiilor neautorizate a fost implementat un sistem de licențe. În baza parametrilor hard a calculatorului, a numelui și denumirii companiei care beneficiază de sistem este generată o cheie valabilă doar pentru un singur calculator. Acesta este unul din primii pași făcuți în direcția obținerii unui pachet soft. În anexa 7 este ilustrată fereastra în care trebuie indicată cheia de înregistrare.

Sistemul a fost conceput să fie administrat de oameni (utilizatori), pentru care acesta reprezintă un instrument de publicare a informației. Pe viitor, vor apărea și alți furnizori de informație: CMS-uri, agenți inteligenți, programe de lucru cu poșta electronică. Sistemul va putea fi administrat atât direct, de utilizatori, cât și indirect, de alte programe prin intermediul unui protocol de comunicare.

Este complicat de elaborat un sistem general care să fie ușor adaptat necesităților reale. Sistemul NeoSite s-a dezvoltat continuu pe parcursul a patru ani. El a înglobat doar acea funcționalitate care este utilizată aproape în orice proiect web. Funcționalitatea specifică era generalizată sau trecută de partea serverului.

Scopul unui sistem informatic este:

1. realizarea eficientă a sarcinilor înaintate de beneficiari;
2. acceptarea sa de către utilizatori.

Implementarea reușită a site-urilor cu ajutorul NeoSite a arătat că modelul care stă la baza sistemului face față obiectivelor propuse.

CMS-ul este un concept și nu un produs. Concept ce întruchipează un set de procese care vor sta la baza generației următoare de site-uri unde autorii de conținut vor avea mai multe privilegii și responsabilități decât proiectanții și programatorii.

Capitolul 3. Aplicațiile sistemului NeoSite

Din momentul apariției primei versiuni a sistemului în 2002, NeoSite s-a dezvoltat permanent. Configurația tuturor proiectelor realizate era ajustată pentru a fi compatibilă cu versiunile noi ale sistemului. În acest capitol vom descri tipul de probleme rezolvate de NeoSite, poziționarea pe piața CMS. Proiectele notorii realizate sînt descrise în §3.2. Detaliat v-a fi descris procesul de elaborare și implementare al site-ului Consiliului Național de Acreditare și Atestare, ca fiind proiectul unde s-au aplicat integral funcționalitățile oferite de sistem.

3.1. Domeniul de aplicabilitate

Sistemul nu reprezintă o soluție completă pentru anumite clase de probleme. NeoSite automatizează unele procese ale managementului conținutului și permite integrarea de funcționalități noi. Astfel, sistemul este o platformă (mediu de dezvoltare) al site-urilor. Sistemul este conceput pentru elaborarea de site-uri bine structurate cu un volum mediu/mare de informație (arborele de conținut să conțină 500-1000 de noduri și mai mult). Cînd informație este multă și diversificată, numărul de entități (tipuri de noduri ale arborelului de conținut) este mai mare, crește numărul și gradul relațiilor stabilite între noduri. Reprezentarea obiectelor complexe prin unele elementare și a relațiilor dintre acestea permite implementarea unor modalități alternative de navigare a conținutului. Acesta este unul din scopurile sistemului: posibilitatea de navigare prin mai multe căi.

Arhitectura distribuită permite utilizarea unor părți componente ale sistemului pentru alte scopuri în afară de elaborarea de site-uri web. Generarea dinamică a interfeței aplicației client în general și a interfeței de gestiune a conținutului bazei de date în particular transformă aplicația client într-un instrument de gestiune a bazelor de date.

3.2. Portofolio

Cu ajutorul sistemului NeoSite au fost elaborate și sînt întreținute în prezent următoarele proiecte:

www.press.mednet.md – proiectul centrului MedNet: presa despre sănătate în Moldova. Este disponibilă arhiva presei din august 2000. NeoSite este folosit de operatorii centrului MedNet pentru publicarea noutăților. Pentru căutarea în conținutul integral al noutăților este utilizat un motor propriu de căutare [45].

www.stats.mednet.md - 250 de indicatori ai sănătății publice în Moldova. Datele sînt vizualizate în formă de tabel, hartă sau grafic. NeoSite a fost folosit pentru managementul șabloanelor site-ului și importarea datelor din fișiere de format special cu prelucrarea lor ulterioară.

Aceste site-uri fac parte din portalul medical www.mednet.md care a luat premiul de gradul II în edițiile anilor 2001, 2002 al concursului WebTop (www.webtop.md).

rusnet.nl - Rusnet Partners este o firmă de consultanță pentru companiile multinaționale și mari preponderent din zona Benelux. Rusnet se specializează în oferirea de servicii acestor companii pentru dezvoltarea activităților lor comerciale cu Rusia și țările europene din CSI.

Partea cea mai dinamică a site-ului o constituie noutățile: zilnic sînt publicate peste 10 noutăți în 3 limbi. Utilizarea motorului de căutare pasiv [45] (fără robot web) asigură prospețimea și relevanța rezultatelor căutării.

www.monument.md - Patrimoniul istoric și arhitectural al Republicii Moldova: cuprinde cele mai importante monumente de arheologie, istorie, arhitectură și artă plastică monumentală. Site-ul cuprinde 409 monumente. Navigarea conform categoriei este asigurată de structura arborelului de conținut (relații implicite între noduri). Autorul monumentului este specificat ca relație dintre două noduri, astfel putem naviga și conform autorilor. Localitatea în care se află monumentele la fel sînt

reprezentate ca relații. Căutarea și filtrarea monumentelor este asigurată de motorul de căutare propus în [45]. Site-ul a obținut premiul de gradul II la concursul WebTop 2003, 2005.

www.e-democracy.md - Site-ul Asociației pentru Democrație Participativă ADEPT.

Un site voluminos în două versiuni: engleză și română, la 2 martie 2006 site-ul avea 33,900 de pagini indexate de Google. Site-ul este static, relații între noduri nu se stabilesc, un nod al arborelui reprezintă o pagină a site-ului. Site-ul a luat premiul I trei ani consecutiv: WebTop 2001, 2002, 2003.

www.tehnomarket.md - Catalogul produselor salonului TehnoMarket. Un site dinamic în care relațiile dintre noduri joacă un rol important pentru sistemul de filtrare a produselor (ex. <http://www.tehnomarket.md/catalog/acoustic/floor/>) și compararea caracteristicilor a două produse (ex. http://www.tehnomarket.md/catalog/acoustic/floor/elac_105/?cmp=587).

www.patria.md - Rețeaua de cinematografe Patria. După complexitate site-ul este similar site-ului www.tehnomarket.md. Pentru indicarea zilelor și orelor când rulează un film au fost introduse noduri virtuale în arborele de conținut: „Premiera - Centru”, „Afisha - Centru”, „Afisha - Rîșcani” etc. Între aceste noduri și filme sînt stabilite relații, în atributele relațiilor este introdus orarul filmelor pentru un anumit cinematograf. Actorii, genul, studioul care a produs filmul sînt specificate ca relații ce sînt folosite și pentru sistemul de filtrare: <http://www.patria.md/movies/>. Premiul I la WebTop 2004.

www.alegeri2005.md - Informație privind campania electorală, inclusiv date despre concurenții electorali. Rezultatele detaliate ale alegerilor parlamentare: participarea electoratului, rezultatele votării pe țară, rezultatele după concurent, după localitate. Site-ul a fost elaborat pe etape, stabilirea șablonului pentru generarea unei pagini este divizat în mai multe *dispatcher_template* (§2.8.) Site-ul are trei versiuni: română, rusă, engleză generate dintr-un singur arbore de conținut. La 2 martie 2006 Google avea indexate 28,500 pagini ale site-ului. Unul din primele site-uri unde s-a folosit reprezentarea ierarhică ci nu liniară a relațiilor dintre noduri. Premiul II la WebTop 2005.

www.bizpro.md - BizPro este un proiect de dezvoltare a întreprinderilor, finanțat de Agenția USAID. Site dinamic întreținut integral de către personalul din cadrul BizPro cu ajutorul sistemului NeoSite.

www.moldovawinehouse.com – complex modern din 4 săli specializate de degustare a vinului. Un exemplu de site mic elaborat cu NeoSite. Relații dintre noduri nu sînt.

3.3. Studiu de caz: site-ul Consiliului Național de Acreditare și Atestare

Consiliul Național pentru Acreditare și Atestare (în continuare – Consiliul Național: CNAA), cu statut de persoană juridică, este instituția administrației publice centrale în domeniul evaluării și acreditării organizațiilor din sfera științei și inovării, precum și al atestării cadrelor științifice și științifico-didactice de înaltă calificare.

CNAA deține informație necesară și utilă pentru comunitatea științifică. Un site web care ar reflecta activitatea CNAA va accelera diseminarea informației. Site-ul existent al CNAA nu reflecta integral informația disponibilă și nu era comod în navigare. Vom descrie procesul de elaborare al site-ului CNAA în modul următor:

- versiunile anterioare ale site-ului CNAA și problemele existente;
- premisele creării unui site nou (publicarea pe site al tezelor);

- analiza și structurarea informației;
- elaborarea modelului de date și a structurii site-ului;
- implementare.

3.3.1. Site-urile anterioare ale CSA/CNAA

Prima versiune al site-ului CNAA a fost elaborată fără aplicarea metodelor propuse de arhitectura informației. Site-ul a fost creat de persoane care nu s-au gândit la viitorii utilizatori ai site-ului și la modul în care trebuia prezentată informația. Site-ul nu avea o structură bine definită și era reînnoit prin simpla publicare a unor documente word salvate în format html. Această versiune este accesibilă în Internet Archive Wayback Machine (http://web.archive.org/web/*/http://csa.acad.md). Cel mai vast și incomod în navigare compartiment al site-ului era „Doctorate”, care conținea nomenclatorul cu specialități științifice, instituțiile cu activitate de doctorat, consiliile științifice specializate. Există doar o singură modalitate de navigare: după domeniul științei. Pe o singură pagină în formă de tabel erau afișate specialitățile din acest domeniu, denumirile instituțiilor și consiliilor științifice pentru fiecare specialitate. Specialitățile erau legături către pagina pașaportului și programei examenului de doctorat. Alte pagini interioare ale compartimentului Doctorate nu erau (pagina instituției de ex.). Problema majoră a site-ului era lipsa căilor alternative de navigare.

Conținutul site-ului a fost introdus în NeoSite. Au fost definite patru entități de bază ale arborelui de conținut: domeniu al științei, instituție, consiliu, specialitate. Nomenclatorul este reprezentat de un subarboare de nivel 3 alcătuit din noduri de tip „domeniu” (nivelul 1 sau 2 pentru subdomeniu) și „specialitate” (nivelul 2 și 3). Nodurile copii ale instituțiilor sînt consiliile din cadrul acestor instituții. Specialitatea consiliului este reprezentată de o relație între nodul „specialitate” și nodul „consiliu”. Președintele și secretarul consiliului sînt specificați într-un document XML al nodului „specialitate”. Specialitățile instituției sînt specificate de relații.

În pagina instituției sînt date legături la specialitățile acesteia și desfășurate consiliile științifice. A apărut posibilitatea de navigare a consiliilor după domeniul științei. În pagina consiliilor unui domeniu (<http://csa.acad.md/css/fm/>) specialitățile sînt grupate după instituție și sortate după grad (D, DH). Lista instituțiilor și consiliilor unei specialități este generată din relațiile instituție-specialitate, consiliu-specialitate. Granularitatea conținutului permite crearea diferitor pagini din același conținut: <http://csa.acad.md/nm/whole.shtml> - toate specialitățile grupate după domeniul științei pe o singură pagină.

După elaborarea site-ului a fost adăugat un compartiment nou: Teze. Pe site era publicat textul integral în format pdf al tezelor de doctor/doctor habilitat. Acest compartiment nu era preconizat la etapa elaborării structurii site-ului, deoarece tezele erau grupate doar după domeniul științei și afișate manual pe site. Tezele nu erau integrate cu alte compartimente ale site-ului, de exemplu, nu era posibil de vizualizat tezele elaborate în cadrul unei instituții sau la o specialitate dată. Tezele au devenit un compartiment dinamic și important al site-ului, era necesară o schimbare în structura site-ului pentru a da mai multă „vizibilitate” acestui compartiment.

3.3.2. Teze digitale

Tezele reprezintă o sursă globală de informație potențialul căreia nu este folosit din cauza disponibilității reduse. Publicarea tezelor în internet este o metodă eficientă de diseminare a rezultatelor cercetărilor. În cadrul unui seminar din septembrie 1999 organizat de UNESCO și consacrat tezelor digitale (*engl. ETD - Electronic Theses and Dissertations*) s-a constatat că problemele majore în accesarea rezultatelor investigațiilor științifice sînt:

- Accesul limitat în organizațiile tradiționale (biblioteci, facultatea universității);
- Prețul pentru varianta tipărită;

- Metode de păstrare și arhivare neadecvate a materialelor în format tradițional (hârtie).

Tehnologiile TI trebuie folosite pentru facilitarea accesului la informația cu caracter științific, în particular la rezultatele publicate în teze.

Aceste probleme încearcă să le abordeze Biblioteca Digitală a Tezelor și Disertațiilor (*engl. NDLTD - Networked Digital Library of Theses and Dissertations*) [46] – organizație internațională dedicată promovării adoptării, creării, utilizării, diseminării și păstrării versiunilor digitale ale tezelor. Scopul acestei organizații constă în: perfecționarea studiilor postuniversitare prin crearea librăriilor digitale accesibile tuturor; sporirea disponibilității cercetărilor și arhivarea lor electronică. Un proiect de perspectivă este Uniunea Cataloagelor NDLTD [47] – repozitoriul materialelor educaționale postuniversitare oferite de instituții membre ale proiectului din toată lumea.

În [48] se menționează că unul din beneficiile tezelor digitale este reducerea investigațiilor similare repetate din cauză că oamenii nu sînt la curent cu rezultatele publicate în teze de alți cercetători. Avînd la dispoziție o colecție globală de teze studenții vor putea căuta lucrări pe tematică similară, să le examineze și să învețe din aceste materiale.

Dacă pînă acum teza era citită de un număr restrîns de oameni, odată cu apariția tezelor digitale oricine o va putea accesa. Autorii și tezele lor vor deveni mai vizibili, ei se vor strădui să lucreze mai calitativ, facultatea va impune cerințe mai stricte față de stilul în care este scrisă o teză și o prezentare mai clară a rezultatelor obținute.

Desigur sînt persoane cu viziuni sceptice în privința publicării online al tezelor și aduc ca argument plagiatul: din moment ce o teză digitală este disponibilă oricui, ea poate fi copiată. Paradoxul publicării tezelor online, mai ales dacă ele sînt indexate integral de un motor de căutare, este posibilitatea identificării textelor asemănătoare. Ca rezultat, publicarea online nu doar sporește vizibilitatea rezultatelor, dar și oferă o metodă eficientă de protecție a proprietății intelectuale și împiedică plagiatul [49].

Cu părere de rău, unii conducători de doctorat sînt împotriva tezelor digitale. Mizînd pe faptul că teza va fi citită de un cerc restrîns, conducătorul științific se poate atîrna mai puțin serios față de obligațiunile sale în procesul de pregătire a doctorandului.

Diseminarea rezultatelor cu ajutorul tezelor digitale este o modalitate rapidă și ieftină pentru cercetătorii din țările în curs de dezvoltare să fie auziți în lume și să contribuie la progresul științific.

Proiectele dedicate tezelor digitale investigate se concentrează pe procesarea automată și distribuită a tezelor, indexarea integrală a tezelor, conversia de formate. Mai puțină atenție este acordată funcționalității, utilității, arhitecturii informației. Posibilitatea de căutare în textul integral al tezelor nu este suficient. În teză este indicat conducătorul științific, consiliul științific specializat, membrii consiliului, referenții oficiali, data susținerii, instituția în care a fost elaborată teza etc. Această informație poate fi păstrată separat, o teză fiind compusă din mai multe elemente și a relațiilor dintre acestea. Aceasta va permite căi de navigare alternative a colecției de teze, se va putea răspunde la următoarele întrebări: cîți doctoranzi a pregătit un conducător științific și în ce ani, cîte teze au fost elaborate în cadrul unei instituții, care sînt specialitățile populare (după numărul de teze elaborate). etc. Conceput ca o bibliotecă electronică, sistemul va fi util în procesul de luare a deciziilor.

Înainte de a elabora structura noului site al CNAA s-au studiat sistemele existente de publicare a tezelor digitale. ETD-db [50] este cel mai răspândit pachet soft pentru tezele digitale datorită suportului NDLTD. Însă din martie 2004 dezvoltarea versiunii oficiale a sistemului s-a stopat. DSpace [51] a fost elaborat de Hewlett-Packard și MIT. DSpace primește, indexează, arhivează și redistribuie rezultatele cercetărilor unei organizații în format digital. Este un produs mai matur și oferă mai multă funcționalitate decât ETD-db. Folosind aceste pachete soft, multe universități au lansat propriile sale biblioteci de teze digitale. Există și proiecte la nivel de țară care colectează tezele din universități: Australian Digital Theses Program (ADT) [<http://adt.caul.edu.au/>], Theses Canada Portal [<http://www.collectionscanada.ca/thesescanada/index-e.html>], Cybertheses [www.cybertheses.org], Swedish Digital Scientific Archive (DiVA) [<http://www.diva-portal.org/>]. Considerăm că Cybertheses este proiectul reprezentativ din cele menționate. Cu ajutorul Cybertheses putem vizualiza teza (fiecare pagină în parte) pe site fără s-o descărcăm pe calculator: navigarea în interiorul tezei este realizată cu ajutorul cuprinsului tezei extras în regim automat din varianta pdf.

3.3.3. Analiza și structurarea informației

Site-ul CNAA a fost elaborat conform etapelor descrise în fig. 3 din §2.2. Specificarea cerințelor față de un site este o procedură complicată fiindcă reprezintă etapa unde sînt implicați oameni din diferite domenii. Din discuțiile avute cu specialiștii CNAA au fost identificate tipurile de vizitatori ai site-ului și informația de care au nevoie. Au fost stabilite următoarele compartimente ale site-ului:

- Structura CNAA;
- Comisia de Atestare;
- Comisia de Acreditare;
- Noutăți;
- Dispoziții, Instrucțiuni;
- Acte Normative;
- Nomenclatorul cu specialități științifice;
- Instituții cu activitate de doctorat;
- Consilii științifice specializate;
- Teze.

În structura site-ului a fost integrat și compartimentul Potențialul Științific care trebuia să cuprindă: conducători de doctorat; deținători de grad; doctoranzi; competitori. Din lipsă de informație acest compartiment la moment nu este activ pe site.

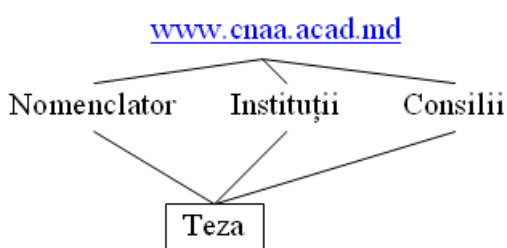
Cel mai dinamic compartiment al site-ului este Teze. Pe an sînt susținute în jur de 200 de teze, în 2005 de ex. au fost susținute 171 teze. Adăugarea continuă a tezelor poate crea probleme de navigare. Dimensiunea exagerată a paginilor și numărul mare de legături va necesita mai mult timp și efort pentru a găsi informația dorită. Coerența globală (§2.5) este asigurată prin generarea paginilor de același tip din șabloane predefinite. Teza este alcătuită conform unor standarde și are o structură bine definită. Conform acestei structuri a fost elaborat șablonul html al tezei și identificate compartimentele site-ului ce au legături cu compartimentul Teze.

Structura tezei:

- Titlul;
- Autorul;
- Gradul: [D, DH];
- Domeniul științei*;
- Specialitatea*;

- Data susținerii;
- Conducătorul științific;
- Referenții oficiali;
- Instituția*;
- Consiliul științific*;
- Statutul tezei: [examinare CSS, examinare CNAA, aprobat/respins];
- Cuvinte cheie;
- Adnotarea;
- Cuprinsul;
- Autoreferatul în format PDF;
- Teza integrală în format PDF.

Elementele marcate cu (*) sînt prezente pe site în compartimentele: Nomenclator, Instituții, Consilii. Aceasta permite introducerea căilor alternative de navigare a tezelor:



Un sistem de filtrare după acești parametri (instituție, specialitate, consiliu) poate compensa sistemele de navigare dar este lipsit de context. În plus, filtrarea este utilizată cînd trebuie specificați doi sau mai mulți parametri. Integrarea în pagina instituției a tezelor elaborate poate fi interpretată ca filtrarea implicită după un parametru. Acest procedeu îl numim *filtrare contextuală*. Utilizatorul beneficiază de contextul paginii pentru a trece în alte compartimente ale site-ului: în pagina unei specialități sînt date instituțiile acestei specialități, consiliile în care sînt susținute teze la această specialitate, lista tezelor în examinare și numărul tezelor din arhivă grupate după anul susținerii.

Tezele sînt transmise CNAA după ce au fost prezentate consiliului științific și stabilită data susținerii. Inițial teza are statutul „În examinare la CSS”, după susținere în cadrul CSS și pînă la ședința Comisiei de Atestare a CNAA unde pretendentului i se conferă titlul științific teza are statutul „În examinare la CNAA”. După ce teza este aprobată de CNAA ea obține statutul „susținută” și trece în arhiva tezelor. Tezele în examinare au o vizibilitate sporită și sînt scoase la suprafața site-ului fiindcă ele sînt mai des întrebate de vizitatori. Există o delimitare clară între tezele în examinare și cele susținute. La 4 martie 2006 în examinare se aflau 70 teze, inclusiv 61 în examinare la CNAA. Monitorizarea unei liste din 70 de elemente este obositoare, deaceia au fost stabilite criterii de filtrare și ordonare a tezelor în examinare. Gruparea după domeniul științei și ordonarea după data susținerii au fost suficiente pentru micșorarea numărului de teze afișate pe o pagină.

Înainte ca utilizatorul să descarce autoreferatul sau teza în format PDF, el are posibilitatea să citească adnotarea, cuvintele cheie și cuprinsul tezei și să decidă dacă teza respectivă prezintă interes pentru dînsul. Descrierea detaliată a tezelor reduce numărul de descărcări inutile a fișierelor mari. În pagina tezei este afișat numărul total al tezelor elaborate la aceeași specialitate și date legături către tezele în examinare. Acest filtru contextual este un indicator al competitivității/popularității specialității.

„Acte Normative” este un compartiment static (modificări rare), regulamentele sînt afișate pe site integral, divizate pe capitole. Clarificarea unei întrebări necesită consultarea mai multor acte normative voluminoase și este un proces de durată. Afișarea capitolelor relevante în dependență de compartimentul site-ului facilitează consultarea legislației. De exemplu, în pagina tezei, în dependență de statut (examinare, susținută, respinsă), sînt afișate legături la capitolele și formularele de care are nevoie doctorandul.

Site-ul CNAA este un site mare: 3 * 3,510 (un nod conține varianta rom, rus, eng) noduri al arborelui de conținut, 35,600 pagini indexate de Google la data de 4 martie 2006 și 1.3Gb în fișiere pdf. Navigarea pe site este complementată de un motor de căutare. Rezultatele căutării cu aplicarea morfologiei limbilor română, engleză și rusă sînt net superioare celor oferite de motoarele de căutare precum Goolge, MSN sau Yahoo. Iar căutarea în textele integrale ale tezelor fac posibilă identificarea tezelor similare ce conțin porțiuni identice de text.

3.3.4. Elaborarea modelului de date și a structurii site-ului. Implementare

În §2.5. s-a menționat că elaborarea modelului hipertext și modelului date se face concomitent. Structura arborelui de conținut este similară modului de formare ale adreselor paginilor: pagina cu adresa /f1/f2/f3/ reprezintă nodul de nivel 3 cu filename='f3' și nodurile părinți au valoarea cîmpului filename 'f1' și 'f2' (§2.6).

Pentru adresele paginilor cărora nu le corespund noduri din arbore sînt create șabloane *dispatcher* separate (§2.8).

Vom descrie entitățile și relațiile lor care participă la construirea compartimentelor dinamice ale site-ului: nomenclatorul, instituții, consilii, teze. Vom stabili adresele URL de acces și structura paginilor. Partea dreaptă a paginilor va fi descrisă după introducerea tuturor entităților.

Pagina de start al site-ului CNAA este ilustrată în anexa 10.

Nomenclatorul cu specialități:

Nomenclatorul din 14 septembrie 2004 conține 351 specialități. În arborele de conținut ele sînt categorizate după domeniul/subdomeniul științei. Paginile domeniilor științei au adresa /nomenclature/[domeniu]/.

Ex: http://www.cnaa.acad.md/nomenclature/physics_mathematics/. Această pagină este ilustrată în anexa 13.

Doar 4 din 22 de domenii conțin subdomenii, deaceia pagini cu adresa /nomenclature/[domeniu]/[sub_domeniu]/ nu există. Pentru paginile acestor domenii a fost aplicat principiul de extindere/constrîngere (§2.3): inițial, toate specialitățile subdomeniilor erau ascunse. Utilizatorul poate desfășura subdomeniile care îl interesează pentru a vizualiza specialitățile. Sistemul memorizează subdomeniile deschise în cookies-urile browserului și la următoarea accesare a paginii le deschide automat.

Din pagina domeniului accesăm paginile specialităților ce au adresa: /nomenclature/[domeniu]/[cod_specialitate]/. Conținutul specialității este alcătuit din pașaport și programa examenului de doctorat. Specialitățile adiacente și complementare din pașaportul specialității sînt reprezentate de relații:

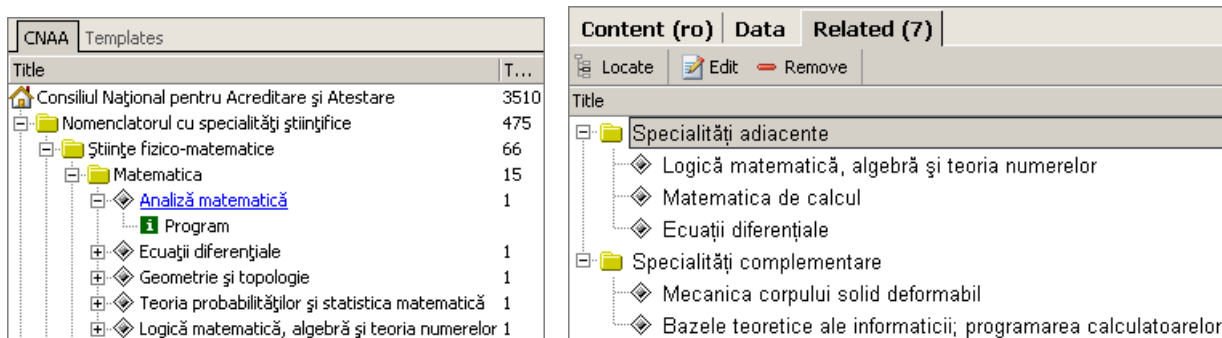


Figura 18. Arborele cu domenii/specialități (stînga). Relațiile specialității (dreapta).

Nodurile „Specialități adiacente” și „Specialități complementare” sînt noduri virtuale în arborele de conținut (§2.4, fig. 8).

Instituții cu activitate de doctorat:

Paginile instituțiilor au adresa /institutions/[instituție]/. Pagina unei instituții este ilustrată în anexa 12. Cele 51 de instituții sînt grupate după ministere. Pentru CNAA sînt relevante consiliile și specialitățile instituției. În fig. 19 specialitățile instituției sînt indicate prin relații cu nomenclatorul. Structura instituției (facultăți, catedre) a fost introdusă pentru a stabili relații cu următoarele compartimente:

- Teze: relația facultate-teză denotă că teza a fost elaborată la facultatea indicată și permite generarea de rapoarte statistice mai detaliate;
- Potențialul Științific: faptul că o persoană activează la o facultate/catedră/laborator este specificat de relația persoană-facultate.

Informația despre structura instituțiilor și potențialul științific n-a fost colectată integral deoarece ea nu este disponibilă pe site.

Pentru materialele prezentate spre acreditare și avizele comisiei de evaluare a fost introdusă categoria „Documente”. Pentru fișiere transmise prin FTP din aplicația client există o entitate (tip al nodului) specială numită „file”. Nodul de tip file conține numele deplin al fișierului transmis și dimensiunea în bațiți al acestuia, dacă fișierul este o imagine, nodul mai conține și dimensiunile ei. Structura mapelor în care sînt transmise fișierele este determinată de șabloane speciale (§2.8.2).

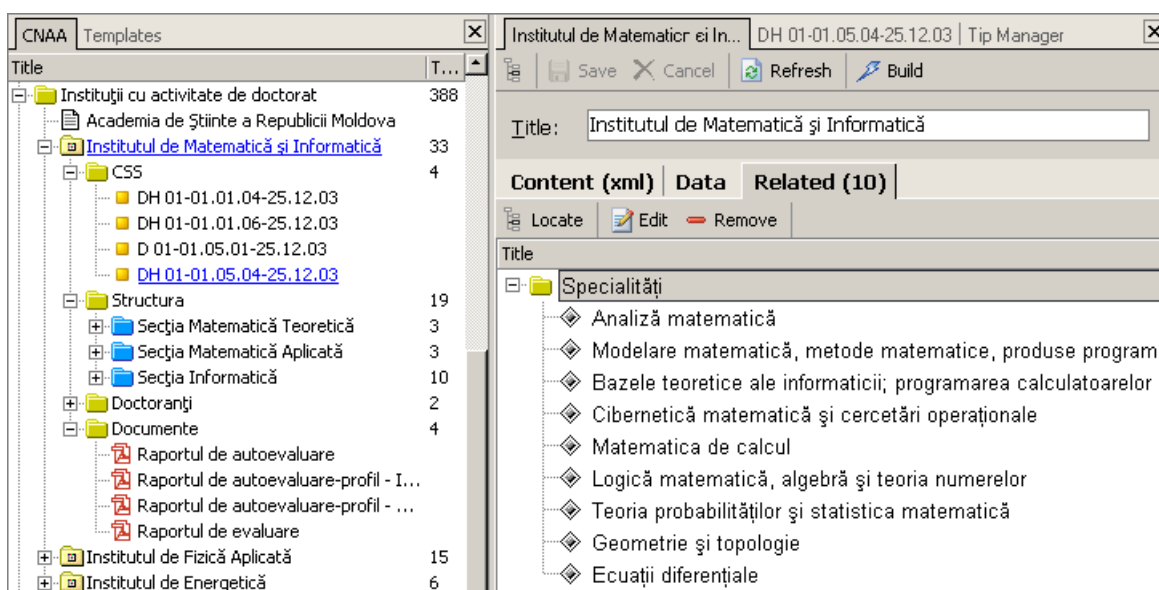


Figura 19. Managementul instituției și a relațiilor sale

Consilii științifice specializate:

Vizitatorul poate căuta un consiliu după mai multe criterii:

- Instituție;
- Domeniul științei;
- Specialitate.

De modul în care consiliile sînt introduse în arborele de conținut depinde:

- eficiența întreținerii ulterioare: cît de comod (de cîte operații este nevoie) editorul gestionează consiliile în cadrul aplicației client;
- existența unei corespondențe dintre adresa URL a consiliului și „adresa” în arborele de conținut;
- Complexitatea introducerii căilor noi de navigare.

Din Fig. 19 se vede că în arbore consiliile sînt noduri copii a instituției. Adresa URL al consiliului este /institutions/[instituție]/css/[consiliu]/. Președintele și secretarul științific sînt date de relația persoană-consiliu. Specialitatea este indicată ca o relație 1:1 (Fig. 20). Din această relație este determinat domeniul consiliului și implementate paginile de navigare a consiliilor după domeniu: /css/[domeniu]/. În această pagină consiliile sînt grupate după instituție.

Content	Data	Related (2)
filename	DH01010504251203	
specialitate	◆ Modelare matematică, metode matematice, [te]	
tip	DH	
phone	D	
email	DH	

Content	Data	Related (2)
Locate Edit Remove		
Title		
Constantin Gaidric		
Svetlana Cojocar		
info	presedinte	

Figura 20. Fereastra de editare a consiliului.

Tezele de doctorat:

În §3.3.3 s-a menționat că Teze este compartimentul cel mai dinamic al site-ului căruia i se va acorda vizibilitatea maximă: integrarea contextuală în alte compartimente ale site-ului. Teza în NeoSite este reprezentată prin relații cu alte noduri pentru a reduce introducerea manuală a datelor și pentru a oferi posibilitatea de căutare/filtrare după mai mulți parametri. În fig. 21 dreapta observăm cum cu ajutorul relațiilor sînt indicați conducătorul și consultantul științific, referenții oficiali, membrii consiliului specializat. Autorul tezei este specificat ca și specialitatea: cu ajutorul relațiilor (operației drag and drop al nodului din arbore în fereastra de redactare a tezei). Deoarece în sistem încă nu au fost introduși doctoranzii, autorul tezei este indicat manual într-un document XML în care sînt specificate și alte informații despre teză (cuvinte cheie, titlul tezei în rom, rus, eng...). În format XML este introdusă informația afișată doar pe pagina tezei (date pasive, §2.3) pentru a reduce numărul de procesări ale fișierelor XML. Adresa URL al tezei sugerată de structura arborelui (fig. 21, stînga) ar fi /theses/[anul]/[luna]/[denumire_teză]/. Adresele URL trebuie să fie concise, memorabile și standarte pentru același tip de pagini. Este imposibil să redai denumirea tezei în două trei cuvinte, ca urmare în teze coloana „filename” nu este folosită iar adresa URL este /thesis/[teza_id]/. teza_id este identificatorul nodului din arbore. Pagina unei teze este ilustrată în anexa 11.

Acestea au fost entitățile de bază ale proiectului CNAA, navigarea orizontală (între două compartimente) pe site este realizată cu ajutorul relațiilor dintre noduri. Conform modelului din fig. 4 §2.3, sistemul de navigare și alte legături utile sînt amplasate în partea dreaptă a paginii. Respectarea acestui șablon asigură coerența globală a site-ului și facilitează adaptarea vizitatorilor la structura site-ului. Acest fapt este accentuat și de dimensiunea redusă a fontului utilizat în partea dreaptă a paginii.

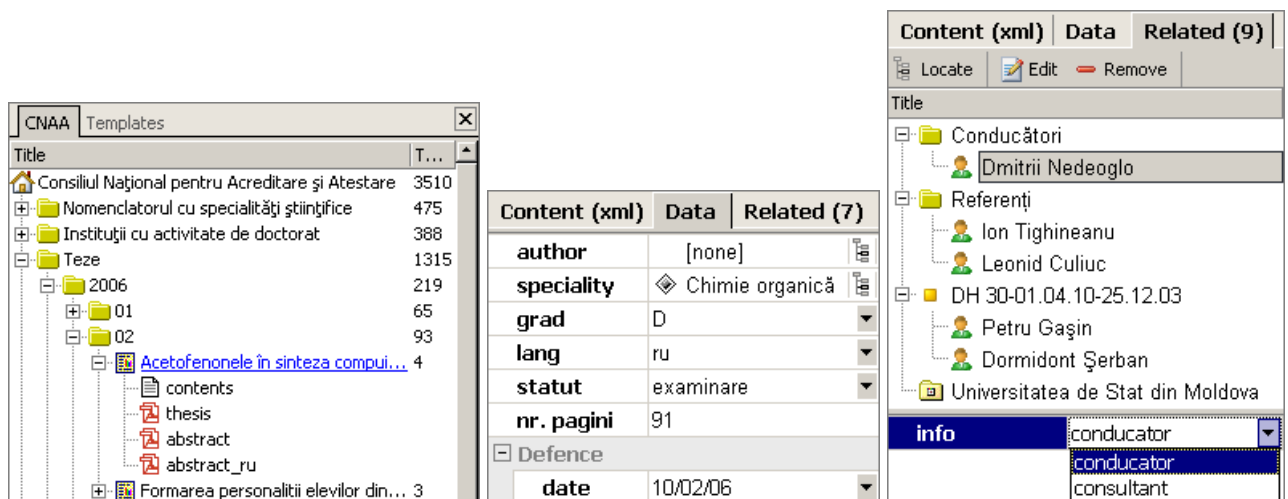


Figura 21. Teza în arborele de conținut; proprietățile tezei; relațiile tezei.

Navigarea tezelor se face direct și indirect:

- navigare directă: vizitatorul se află în categoria Teze;
- navigare indirectă: tezele sînt afișate în cadrul altor compartimente ale site-ului.

Navigare directă:

De obicei intrările repetate pe site sînt făcute începînd cu prima pagină (home page). Vizibilitatea maximă este oferită tezelor care curînd vor fi susținute: primele 5 teze după data susținerii din cele în examinare la CSS sînt afișate pe pagina de start. În tabelul de mai jos sînt date paginile compartimentului teze.

	URL	Descriere
1	/theses/	5 teze în examinare, 5 teze recent susținute. Linkuri la 2,3,4,6
2	/theses/examination/	toate tezele în examinare la CSS. Linkuri la 4, 6
3	/theses/examination/cnaa/	toate tezele în examinare la CNAA. Linkuri la 4, 6
4	/theses/[anul]/	linkuri la 5, 7
5	/theses/[anul]/[luna]/	tezele sustinute in luna indicata.
6	/theses/[domeniu]/	tezele în examinare din domeniul dat. Linkuri la 7.
7	/theses/[domeniu]/[anul]/	teze din domeniul dat susținute în anul indicat.

Navigare indirectă:

În pagina instituției sînt afișate tezele în examinare și linkurile:

/theses/[institutie]/ - toate tezele în examinare.

/theses/[institutie]/[anul]/ - arhiva tezelor elaborate în această instituție.

În paginile consiliilor și specialităților sînt afișate tezele în examinare și linkuri către sistemul de

filtrare a tezelor: /theses/filter/?param1=value1¶m2=value2... Parametrii filtrului:

spec=[specialitate_id]

yr=[anul_susținerii]

css=[consiliu_id]

g=[gradul: D,DH]

Filtrul tezelor nu are o interfață prin care s-ar putea specifica cîtiva parametri și vizualiza rezultatele. Adresele URL ale tuturor combinațiilor posibile de parametri pentru care rezultatul filtrării nu este vid sînt indicate în paginile site-ului. Specificarea implicită a parametrilor în dependență de context (pagina accesată) este mai eficientă decît varianta explicită cînd:

- Valorile posibile primite de un parametru este mare: este incomod să operezi cu lista compusă din 351 de specialități;
- Sînt frecvente combinațiile de parametri cu rezultat vid. Cînd utilizatorul de cîteva ori specifică o serie de parametri și rezultatul este vid el este frustrat: există specialități la care nu au fost elaborate teze în anumiți ani. În pagina specialității sînt indicați doar anii în care au fost elaborate teze pentru specialitatea dată. În dreptul anilor sînt indicate și numărul de teze elaborate. Utilizatorul fără să apese linkul va ști cîte teze vor fi afișate.

3.3.5. Sistemul de căutare

Navigarea/filtrarea pe site este complementată de sistemul de căutare. Majoritatea site-urilor fie au un sistem de căutare elementar care nu asigură relevanța rezultatelor, fie în genere nu oferă această funcționalitate. Unele CMS-uri oferă motoare de căutare proprii. S-a încercat [45] dezvoltarea unui motor de căutare pentru NeoSite care a fost utilizat în toate proiectele pînă la CNAA. Lipsa de resurse (timp, experiență, rentabilitate) ne-au determinat să căutăm un motor de căutare performant care poate fi ajustat la specificul fiecărui proiect. După testarea a trei motoare de căutare: ASPseek (<http://www.aspseek.org/>), mnoGoSearch (<http://www.mnogosearch.org/>) și DataparkSearch (<http://www.dataparksearch.org/>), a fost ales DataparkSearch – un sistem gratuit cu cod deschis (open source) scris în limbajul C++ care rulează pe sistemele Unix. Datapark posedă două calități importante:

1. integrarea parserelor externe pentru transformarea fișierelor pdf, doc, xls etc. în format text pentru indexare;
2. Utilizarea morfologiei limbii la căutare.

Datapark folosește dicționarele și regulile de inflexiune ale sistemului ispell pentru a genera toate formele cuvintelor. Pentru limba română, spre deosebire de engleză și rusă, astfel de dicționare nu au fost elaborate. Mihai Budiu (<http://www.cs.cmu.edu/~mihaib>) a creat pentru ispell un dicționar mic și un set de reguli de inflexiune pe departe complet al limbii române.

Grupul de procesare al limbajului natural din cadrul Institutului de Matematică și Informatică (<http://www.math.md/>) a elaborat pachetul RomSP – un spell checker al limbii române pentru MS Word. Dicționarul de cuvinte avea în jur de 70,000 de cuvinte, din care un program de inflexiune genera toate formele cuvintelor: peste 1,000,000. Regulile de inflexiune nu puteau fi ușor rescrise în formatul ispell. Din baza de date a formelor cuvintelor a fost creat un fișier de forma:

```
word word1
...
word wordN
```

unde word1...wordN erau formele cuvîntului word. În configurația sistemului Datapark era indicat că acest fișier conține sinonime! În acest mod a s-a obținut un motor de căutare care „înțelegea” morfologia limbii române. Rezultatul căutării după cuvîntul „model”

(<http://www.cnaa.acad.md/cgi-bin/cnaa/search.cgi?t=1&q=model>) sînt paginile ce conțin cuvintele: model: 84, modele: 144, modelelor: 59, modelele: 50, modelului: 46, modelul: 162, models: 1

84, 144, 59 ... este numărul de apariții al cuvîntului respectiv în paginile găsite. „models” a fost inclus fiindcă cuvintele din interpelare sînt căutate în dicționarele celor trei limbi.

Motorul de căutare „vede” altfel site-ul. Paginile nu sînt indexate integral. În șabloanele site-ului sînt definite secțiuni care vor fi excluse la indexare. Una din secțiuni este partea dreaptă a paginii. În sursa html a paginilor aceste secțiuni sînt demarcate prin construcțiile

```
<!--noindex--> ... </--noindex-->
```

Titlul paginii conține titlurile prescurtate ale nodurilor părinți, pentru motorul de căutare titlul paginii este doar titlul nodului. În configurația motorului de căutare sînt definite zone, căutarea poate fi efectuată doar în cadrul unei zone anumite. Există două zone principale: paginile html ale

site-ului și fișierele doc, pdf. Zona paginilor html este compusă din 3 compartimente: versiunile site-ului în română, rusă, engleză. Paginile html sînt organizate în categorii: instituții, teze, consilii, specialități, acte normative. Iconița în dreptul titlului paginii găsite din rezultatele căutării denotă categoria din care face parte.

Informația despre categoria paginii este folosită la generarea rezultatelor căutării: pentru teze sînt indicate și cuvintele cheie:



1.  [Teoria și metodologia evaluării în învățămîntul universitar](#)
... pe 25 martie 2005 și aprobată de CNAA evaluare, calitate, **model**, axă, cadru didactic, învățămînt universitar, măsurare, gestiune, stabilirii fundamentelor teoretice ale evaluării, elaborării **modelului** de evaluare a cadrului dida...
[Cuvinte Cheie](#) evaluare, calitate, **model**, axă, cadru didactic, învățămînt universitar, măsurare, gestiune, produs, procedură, proces, teleologie, structuralism, cibernetică, sistemism, docimologie, psihometrie, edumetrie, scop, obiect, destinat, surse de infor
2.  [01.05.04 – Modelare matematică, metode matematice, produse program](#)
01.05.04 – Modelare matematică, metode matematice, produse program ... 01.05.04 - Modelare matematică, metode matematice, produse program Pașaportul ... specialității Formula specialității: st...
3.  [Caracteristici fizico-chimice și procese în sistemele apă-materii în suspensie-sedimentele nistrului și prutului](#)
... sedimente acvatice, structura, proprietăți de suprafață, **modele** fizico-chimice. – 0.98 Mb / în română CZU suprafață s-a efectuat prin verificarea adecvanței unui șir de **modele** fizico-chimice. ... S-au analizat **modele** ...
[Cuvinte Cheie](#)

Figura 22. Rezultatele căutării pe site-ul www.cnaa.acad.md

Pagina integrală a rezultatelor căutării este ilustrată în anexa 14.

3.3.6. Statistica site-ului. Evaluare

Una din metodele de evaluare a site-ului (structură, navigare, funcționalitate) este sistemul de statistică a vizitelor. Modul în care utilizatorii intră și navighează site-ul, paginile des vizitate etc. ajută la identificarea sarcinilor pe care utilizatorii doresc să le realizeze. Statistica site-ului CNAA este accesibilă tuturor pe adresa <http://www.cnaa.acad.md/stats/>.

În fig. 23 observăm că cea mai populară pagină a site-ului este pagina de start, atît după numărul de vizualizări (Viewed: 2525 în mediu pe zi) cît și după punctele de intrare pe site (Entry: 1449). Punctul de intrare este prima pagina accesată de utilizator în timpul unei sesiuni de lucru pe site. Era de așteptat ca pagina de start a site-ului să fie cea mai des vizitată, fiindcă ea reprezintă un punct de trecere spre categoriile site-ului. În procesul lucrului utilizatorul intersectează de mai multe ori această pagină. Faptul că pagina de start este cel mai popular punct de intrare înseamnă că majoritatea vizitatorilor accesează site-ul prin introducerea directă a adresei site-ului în browser sau accesîndul din linkurile preferate (favorites): deci există un grup de utilizatori care accesează site-ul regulat.

După numărul de vizualizări prima pagină a categoriei Teze este pe al doilea loc. Așadar, cel mai popular compartiment al site-ului CNAA sînt tezele. În coloana „Exit” este indicat numărul de vizitatori (în mediu pe zi) pentru care pagina respectivă este ultima pagină vizitată în timpul unei sesiuni de lucru.

Pagina /theses/ este cea mai nepopulară în această privință (cu 205 de vizitatori), fiindcă ea este punctul de acces la tezele publicate pe site. Probabil că în jur de 100 vizitatori (205 – 98 = 107) intră regulat pe site și accesează compartimentul pentru a afla tezele care curînd vor fi susținute și cele recent susținute.

După teze, cele mai accesate sînt specialitățile științifice. Numărul de intrări și ieșiri de pe site prin paginile specialităților este mare. Împreună cu analiza interpelărilor către motoarele de căutare Google, Yandex, Yahoo, MSN după care vizitatorii au intrat pe site-ul CNAA putem constata că mulți vizitatori intră pe site din motoarele de căutare direct în paginile specialităților în speranța să găsească materiale pe tematica expusă în pașaportul specialității și programul examenului de doctorat. Doar 11% din ei o să continue să navigheze site-ul. Cu toate că este o popularitate

artificială, în acest mod se pot determina specialitățile întrebate. După locația geografică a vizitatorilor (fig. 24) ne dăm seama că majoritatea traficului generat către paginile nomenclatorului vine din România. Pașapoartele specialităților sînt într-o singură limbă: română.

Pages-URL (Top 20) - Full list - Entry - Exit					
66885 different pages-url	Viewed	Average size	Entry	Exit	
/	2525	9.22 KB	1449	581	
/theses/	1472	15.40 KB	58	205	
/nomenclature/law/120008/	1347	283.66 KB	1021	919	
/nomenclature/economy/080001/	1263	45.26 KB	1022	869	
/nomenclature/philology/100101/	1251	43.69 KB	986	915	
/nomenclature/economy/080010/	1219	85.21 KB	992	880	
/nomenclature/philology/100106/	1118	45.08 KB	921	816	
/nomenclature/economy/080005/	985	63.98 KB	780	724	
/nomenclature/economy/080012/	976	35.51 KB	832	688	
/nomenclature/philology/100108/	953	126.40 KB	717	699	
/nomenclature/pedagogy/130001/	940	51.77 KB	721	635	
/nomenclature/politics/230001/	911	112.53 KB	677	601	
/nomenclature/geography/110011/	799	34.77 KB	653	571	
/nomenclature/law/120010/	795	92.22 KB	616	568	
/nomenclature/sociology/220004/	784	35.08 KB	677	600	
/nomenclature/medicine/140002/	700	31.00 KB	585	507	
/nomenclature/psychology/190007/	672	50.41 KB	543	460	
/nomenclature/biology/030007/	603	55.78 KB	447	369	
/nomenclature/medicine/140022/	564	76.67 KB	468	427	
/nomenclature/art-culture/170004/	560	38.33 KB	493	435	

Figura 23. Cele mai populare pagini CNAA din februarie 2006.

Countries					
Countries	Pages	Hits	Bandwidth		
Moldova	md	91269	92323	3.94 GB	
Romania	ro	71369	71827	4.82 GB	
Russian Federation	ru	5256	5383	332.13 MB	
Unknown	unknown	1922	2139	64.05 MB	
United States	us	1847	1847	134.97 MB	
Ukraine	ua	1043	1043	101.57 MB	
Germany	de	532	532	49.53 MB	
Canada	ca	503	503	19.00 MB	
France	fr	384	384	31.13 MB	
Italy	it	377	377	19.45 MB	
Great Britain	gb	322	322	21.13 MB	

Figura 24. Traficul generat după locația geografică a vizitatorilor.

3.4. Concluzii

Succesul implementării unui CMS depinde în mare măsură de analiza și structurarea prealabilă a informației, identificării tipurilor de vizitatori. Beneficiarul trebuie să fie cointerestat și să participe activ la întrunirile cu proiectanții pentru a oferi o imagine cât mai clară a proceselor și informației din cadrul organizației. După lansarea site-ului cu CMS-ul lucrează beneficiarul, deaceia este important ca la etapa inițială de elaborare a structurii site-ului, stabilirea tipurilor de conținut, relațiilor, etc. proiectanții să-și formeze un model cât mai precis.

Elaborarea site-urilor este un proces iterativ, cerințe noi și modificări pot apărea la orice etapă de implementare. Totuși, scheletul site-ului este definit la prima etapă de elaborare a conceptului pentru care se alocă în jur de 30% din timpul total alocat.

Simplitatea integrării de noi tipuri de conținut și stabilire de noi relații în NeoSite este rapid asimilată de personalul beneficiarului care lucrează cu sistemul, încât ei singuri pot integra conținut

nou pe site. Acest lucru este îmbucurător, dar poate contribui negativ la arhitectura informațională a site-ului și crea probleme de navigare. Orice schimbare în structura site-ului este controlată și aprobată de proiectanți.

Site-ul CNAA este primul site în care căutarea ia în considerație morfologia limbii române. Lipsa regulilor de inflexiune pentru pachetul ispell ne-a impus să folosim un dicționar ce conținea toate formele de cuvinte generat dintr-un dicționar de 70,000 de cuvinte. Sistemul de căutare, cu morfologia limbilor engleză și rusă ocupă în memorie 20Mb, la adăugarea fișierului cu „sinonime” procesul ocupă deja 176Mb. Cu toate că nu are legătură cu domeniul CMS, elaborarea setului de reguli de inflexiune pentru ispell ar fi un lucru util.

Publicarea tezelor în internet trebuie să sporească calitatea cercetărilor. Elaborarea site-ului CNAA este un pas spre integrarea cercetărilor efectuate în Moldova în spațiul informațional global. Unul din scopurile site-ului este sporirea vizibilității și valorificarea rezultatelor științifice din Moldova la nivel internațional.

Concluzii și recomandări

Noțiunea de management al conținutului reprezintă soluția actuală pentru problema mai veche de management al comunicării umane. Managementul conținutului este un concept cu mai multe fațete, este o activitate care trebuie să fie centrată pe utilizator: să ia în considerație problemele și necesitățile lor, să aplice tehnologiile adecvate pentru a ajuta oamenii în lucrul lor.

Managementul conținutului are trei faze distincte: 1) colectarea informației: managementul bunurilor digitale (*engl. digital asset management*) 2) crearea de conținut: structurarea, categorizarea informației 3) publicarea conținutului în formate utile pentru audiența site-ului.

Cunoștințele, informația și conținutul sunt resursele principale ale economiei moderne. De modul în care noi le creăm și gestionăm, depinde succesul nostru ca indivizi și/sau organizații.

Astăzi asistăm la o schimbare de accente. Industria Tehnologiilor Informaționale (TI) pune accentul pe tehnologii și nu pe informație. Cuvântul „informație” figura în expresia TI în calitate de substantiv: informația e ceva care trebuie procesat, distribuit și stocat. În contextul site-urilor web, accentul se pune pe produsele soft de management al conținutului și alte aplicații aferente, dar nu pe managementul conținutului ca atare. Aceasta explică eșecul mai multor proiecte de implementare a soluțiilor CMS. Experiența anterioară și concurența acerbă a favorizat consolidarea pieței de soluții CMS și standardizarea proceselor managementului conținutului. Este conștientizat faptul că conținutul este mai important decât tehnologiile pentru managementul acestuia.

Modelul de date propus în teză permite integrarea de noi tipuri de conținut și reprezentarea conținutului complex prin mai multe noduri de conținut și a relațiilor ierarhice. Separarea de conținut și funcționalitate a permis administrarea site-ului de persoane slab inițiate TI. Administratorii site-ului (persoanele care gestionează conținutul) se ocupă de introducerea conținutului, care în mod automat este publicat pe site în baza modelului site-ului elaborat la etapa de proiectare.

Modelul paginii de conținut propus în §2.3 și procedeul de filtrare contextuală sunt aplicabile la orice site complex ce oferă informație structurată. Filtrarea contextuală asigură legătura dintre compartimentele site-ului și oferă un context de navigare pentru utilizator.

Modelul de evenimente (§2.7) și serviciile web de partea serverului (§2.8.2) asigură flexibilitatea modelului CMS: Interfața este separată de logica aplicației.

CMS-ul NeoSite, implementat după modelul propus, este recomandat studiourilor Web pentru elaborarea site-urilor complexe. Conținutul pe site va fi plasat de către beneficiar prin intermediul aplicației client al sistemului. NeoSite poate fi folosit și ca instrument adăugător în cadrul unor proiecte mai mari unde se cere o interfață comodă și flexibilă de management al conținutului: informația poate fi folosită în cadrul altor aplicații (site-ul web este una din posibilitățile de publicare a conținutului).

Rezultatele principale ale lucrării sînt următoarele:

- A fost elaborat un model de date relațional-ierarhic (descriș în §2.4) în care relațiile de grad N sînt reprezentate prin N-1 relații de grad 2. Stabilirea de noi tipuri de relații între noduri nu necesită noi tabele în baza de date și modificări esențiale în definiția tabelor existente. Acest lucru asigură flexibilitatea modelului;

- A fost elaborat și implementat un sistem de generare a interfeței de management al conținutului. Interfața generică propusă poate fi folosită atât în CMS-uri, cât și în orice sistem în care predomină operații (adăugare, modificare) cu date;
- A fost introdusă noțiunea de context al relației. Semnificația relației este dată implicit de nodurile superioare, acest lucru a permis generarea de interfețe simple de management al conținutului;
- S-a propus elaborarea în paralel al modelului date și hipertext al site-ului. S-a accentuat dependența dintre aceste modele. Aceasta a redus considerabil efortul pentru stabilirea corespondenței dintre adresa URL a paginii accesate și nodului din modelul date corespunzător. Cu alte cuvinte, este promovată similaritatea dintre arborele de conținut și structura ierarhică a adreselor URL al paginilor;
- A fost elaborat un model de pagină web în care se accentuează importanța relațiilor dintre nodurile de date. S-a constatat că paginile complicate de conținut pot fi eficient construite din datele nodurilor de legătură și a nodurilor copii al nodului de bază asociat adresei paginii Web (§2.3). Acest fapt a permis implementarea de site-uri complexe ce au la bază un model de date simplu;
- A fost propusă metoda de filtrare contextuală pentru interconectarea paginilor și implementarea căilor alternative(indirecte) de navigare pe site. Aplicarea acestei metode accentuează legăturile paginii cu alte compartimente ale site-ului și contribuie la conștientizarea poziției conținutului paginii în spațiul informațional al site-ului. Ca urmare, devine mai clară structura site-ului și se reduce timpul necesar vizitatorului pentru adaptare;
- În baza modelelor propuse a fost implementat CMS-ul NeoSite care conține funcționalitatea de bază necesară elaborării și întreținerii site-urilor web, dar care oferă posibilitatea dezvoltării ulterioare pînă la o soluție pentru o problemă specifică. Succesul implementării mai multor site-uri cu ajutorul CMS-ului elaborat subliniază importanța modelelor conceptuale care stau la baza unor astfel de sisteme;
- Viabilitatea sistemului NeoSite, ce are la bază rezultatele expuse, este demonstrată prin implementarea cu succes a peste 15 site-uri web de diferit grad de complexitate, care au obținut premii de gradul I și II în cadrul concursului național WebTop.

Bibliografia

1. Mihai Drăgănescu, „Societatea informațională și a cunoașterii. Vectorii societății cunoașterii”, studiu pentru Proiectul SI-SC (Societatea Informațională - Societatea Cunoașterii) al Academiei Române, București, 9 iulie 2001. Pe Internet: http://www.academiaromana.ro/pro_pri/pag_com01socinf_tem.htm
2. Rothfuss G., Ried C. „Content Management with XML”, Springer-Verlag, Berlin, Germany, 2001, ISBN 3540665943.
3. Paul Browning, Mike Lowndes. “Content Management Systems”, JISC TechWatch Report, September 2001
4. The Problems with CMS, The Asilomar Institute for Information Architecture (AIFIA), http://aifia.org/pg/the_problems_with_cms.php
5. “Web Content Management: Covering the Essentials, Avoiding Overspending”, Jupiter Research 2003, <http://www.jupiterresearch.com>
6. O. Burlaca, “NeoSite: A Simple Content Management System”, Computer Science Journal of Moldova, vol. 12, no.1(34), 2004, pp. 107-126.
7. O. Burlaca, “Generic Interfaces for Managing Web Data”, Computer Science Journal of Moldova, vol. 13, no.1(37), 2005
8. What is Content Management, The Gilbane Report, Vol. 8, No. 8, October 2000. www.gilbane.com
9. Louis Rosenfeld. „Content Management and Information Architecture”, CMS Watch, 2003 <http://www.cmswatch.com/>
10. The Content Management Community of Practice. <http://www.cmprofessionals.org/>
11. CMS Wiki: a knowledge base for Content Management Professionals. <http://www.cmswiki.com/tiki-index.php>
12. Louis Rosenfeld, Peter Morville. „Information Architecture – Designing Large-Scale Web Sites” 2nd edition, O’Reilly 2002
13. Salminen, A. „Methodology for Document Analysis.” Encyclopedia of Library and Information Science. 2000. Pag. 299-320.
14. Sprague, R.H. „Electronic Document Management: Challenges and Opportunities for Information Systems Manager”. MIS Quarterly 19 (1), 1995, pag. 29-49.
15. Mcintosh, M. „Content Management Using the Rational Unified Process”. Rational Software White Paper. 2000
16. Boiko, B. „Content Management Bible”. 2002, New York, U.S.A: Hungry Minds, Inc.
17. Gerry McGovern, Rob Norton. „Content Critical: Gaining Competitive Advantage through High-Quality Web Content”, Financial Times Prentice Hall, 2001
18. JoAnn T. Hackos. „Content Management for Dynamic Web Delivery”, Wiley, 2002
19. Bill Trippe. „Bridging the Real Silos in Content Management with Flexible Content Objects”, The Gilbane Report, 2003
20. Content Management directory on DMOZ: http://dmoz.org/Computers/Software/Internet/Site_Management/Content_Management/
21. „Open Source Content Management Systems: A Parallel Universe”, The Gilbane Report, Vol. 9, No. 4, 2001.
22. Michael Seifert, “Choosing a Content Management System - An objective market overview”, Sitecore, January 2004.
23. Seth Gottlieb, „Content Management Problems and Open Source Solutions”, Optaros white papers and reports, January 2006. www.optaros.com
24. “The Classification & Evaluation Of Content Management Systems”, The Gilbane Report, vol. 11, No. 2, March 2003.

25. James Robertson, "How to evaluate a content management system", Knowledge Management Column, January 2002.
http://www.steptwo.com.au/papers/kmc_evaluate/
26. M. Fernandez, D. Florescu, J. Kang, A. Levy, D. Suci, "Catching the Boat with Strudel: Experiences with a Web-Site Management System", SIGMOD Conference, pag. 414-425, 1998, <http://citeseer.ist.psu.edu/fernandez98catching.html>
27. Stefano Ceri, Piero Fraternali, Aldo Bongio, "Designing Data-Intensive Web Applications", 2003, Morgan Kaufmann Series in Data Management Systems, ISBN: 1-55860-843-5
28. Eric Lecolinet, "XXL: A Dual Approach for Building User Interfaces", ACM Symposium on User Interface Software and Technology 1996. <http://www.inf.enst.fr/~elc/XXL/index.html>
29. B. Surjanto, N. Ritter, H. Loeser, "XML Content Management based on Object-Relational Database Technology", Web Information Systems Engineering 2000.
30. Carlos Castillo, "A Framework for the design and implementation on web sites", IADIS WWW/Internet, 2002, <http://citeseer.nj.nec.com/castillo02framework.html>
31. Oleg Burlaca, "An object/relational approach to content management systems", BiT+ 2003, Chişinău, p. 167.
32. P.P. Chen. "The Entity-Relationship Model - Toward a Unified View of Data". ACM TODS, 1(1), March 1976, pp. 9-36.
33. T. Schattkowsky, M. Lohmann, „Rapid Development of Modular Dynamic Web Sites using UML”, In Proc. of 5th International Conference on UML, 2002.
34. P. Borgermans, T. Wood, P. Forsyth, B. Pirt, B. Dieding, "Learning eZ publish 3: Building Content Management Solutions", Packt Publishing Limited, May 2004. ISBN: 1904811019
35. M. Bernstein, S. Robertson, „Zope Bible”, Wiley, April 2002, ISBN: 0764548573
36. M. Brockmann, S. Lühnsdorf, M. Pratt ş.a., „Zope Web Application Construction Kit”, Sams Publishing, October 2001, ISBN: 0672321335
37. A. McDonald, R. Welland, "Web Engineering in Practice", Proceedings of the Fourth Workshop on Web Engineering, May 2001, <http://citeseer.ist.psu.edu/mcdonald01web.html>
38. C. Gaidric, O. Burlaca, „Decision support system user interface: design and implementation” BiT+ 2001, April 2002, Chişinău. p. 66.
39. O.M.F. De Troyer, C.J. Leune, "WSDM: A User Centered Design Method for Web Sites", Computer Networks and ISDN systems, Proceedings of the 7th International WWW Conference, Elsevier, pp. 85 - 94, 1998
40. Simon Rollinson, Stuart Roberts, "An Entity-Relationship Approach to Modelling Database User Interfaces", University of Leeds, 1999
41. Sebastiano Vigna, "ERW: Entities and Relationships on the Web", 11th International WWW Conference, USA 2002
42. Alan Knight, Naci Dai. "Objects and the Web." IEEE Software, March/April 2002.
43. Darren Chamberlain, David Cross, Andy Wardley, „Perl Template Toolkit”, O'Reilly, December 2003, ISBN: 0-596-00476-1
44. „The CHAOS Report”, Standish Group International, 1994. www.standishgroup.com
45. Oleg Burlaca, "The hidden web. Specialized search engines”, Bit+ 2002, Chişinău, p. 92.
46. Networked Digital Library of Theses and Dissertations (NDLTD), <http://www.ndltd.org>
47. NDLTD Union Catalog, <http://www.vtls.com/ndltd/>
48. The Guide for Electronic Theses and Dissertations, <http://etdguide.org/etdguide.pdf>
49. O. Burlaca, Constantin Gaidric, Svetlana Cojocaru, "A Content Management System for Electronic Theses", IV International Conference on Microelectronics and Computer Science, Chisinau, 2005, pp. 514-509.
50. VT ETD-db, <http://scholar.lib.vt.edu/ETD-db/>
51. DSpace, <http://www.dspace.org/>

ADNOTARE

la teza de doctor „Sistem de management al conținutului pentru Web”
autorul: Oleg Burlaca

Trecerea la societatea informațională implică sporirea volumului de informații și accelerarea comunicării, problema managementului eficient al conținutului devine tot mai importantă. Sistemele de management al conținutului (CMS) facilitează elaborarea și mentenanța site-urilor Web complexe.

În teză este propus și implementat un model nou de CMS bazat pe următoarele rezultate:

- Modelul de date relațional-ierarhic în care relațiile sînt reprezentate sub formă de arbore. Relațiile de grad N sînt reprezentate prin N-1 relații binare;
- O interfață generică de management al conținutului;
- O tehnică de elaborare în paralel al modelului date și modelului hipertext al site-ului;
- Un model de pagină web eficient generată din modelul de date propus care asigură interconectarea logică a compartimentelor site-urilor structurate și complexe;
- Metoda de filtrare contextuală pentru interconectarea paginilor și implementarea căilor alternative de navigare pe site.

În baza modelelor propuse a fost implementat CMS-ul NeoSite care conține funcționalitatea de bază necesară elaborării și întreținerii site-urilor web, dar care oferă și posibilitatea dezvoltării ulterioare pînă la o soluție pentru o problemă specifică. Succesul implementării mai multor site-uri cu ajutorul CMS-ului elaborat subliniază importanța modelelor conceptuale care stau la baza unor astfel de sisteme.

Cuvinte Cheie

Sistem de management al conținutului, CMS, model de date, arhitectura sistemului, elaborarea site-urilor web.

ABSTRACT

on the Ph.D. thesis „A Web Content Management System”

Author: Oleg Burlaca

The shift towards the information society means large volumes of information and accelerated communication. The problem of content management is becoming important. Content management systems (CMS) facilitate complex web site development and maintenance.

The thesis describes a new model of CMS based on the following results:

- A relational-hierarchical data model where relationships are hierarchically managed. N-ary relationships are described by N-1 binary relationships;
- The generic user interface for content management;
- An approach for parallel development of the data model and the hypertext model of a website;
- A webpage model that can be efficiently generated from the underlying data model. The model promotes the logic interconnection of site sections in a complex website;
- The contextual filtering method for page interconnection and implementation of alternative ways for website navigation.

Based on the proposed models, the NeoSite CMS was implemented. It contains only the basic functionality needed for the development and maintenance of websites. At the same time, the system allows future development and integration of new features. Successful implementation of websites using NeoSite underpins the importance of such conceptual models.

Keywords

Content management system, CMS, data model, system architecture, web development.

АННОТАЦИЯ

на докторскую диссертацию
”Система управления содержанием для Web”
Автор: Олег Бурлака

Переход к информационному обществу предполагает увеличение объемов информации и ускорение процессов общения. Проблемы связанные с управлением содержанием становятся все актуальней. Системы управления содержанием (CMS) упрощают создание и обслуживание комплексных Web сайтов.

В диссертации предложена и реализована новая модель CMS которая основана на следующих результатах:

- Реляционно-иерархическая модель данных в которой связи представлены древовидной структурой. Связи N-ого уровня представлены N-1 бинарными связями;
- Общий настраиваемый интерфейс для управления содержанием;
- Технология параллельной разработки модели данных и гипертекстной модели сайта;
- Модель Web страницы которая позволяет эффективно генерировать страницы на основе модели данных. Предложенная модель обеспечивает логическую взаимосвязь между разделами сложного сайта;
- Контекстный метод фильтрации для связывания страниц и внедрения новых путей навигации по сайту.

На базе предложенных моделей была разработана система NeoSite CMS которая содержит базовую функциональность для разработки и поддержки Web сайтов. Система позволяет дальнейшее развитие и интеграцию новой функциональности. Успех разработки многочисленных сайтов на базе NeoSite подчеркивает значимость концептуальных моделей которые лежат в основе таких систем.

Ключевые слова

Система управления содержанием, CMS, модель данных, архитектура систем, разработка вэб сайтов.

ANEXA 1

The screenshot shows the NeoSite interface for the 'Comisia Nationala de Acreditare si Atestare'. The main content area displays a table of items with the following data:

tip	filename	title	title_ru
◆	010101	Analiză matematică	Математический анализ
◆	010102	Ecuatii diferențiale	Дифференциальные уравнения
◆	010104	Geometrie și topologie	Геометрия и топология
◆	010105	Teoria probabilităților și sta...	Теория вероятностей и матема...
◆	010106	Logică matematică, algebră...	Математическая логика, алгеб...
◆	010107	Matematica de calcul	Вычислительная математика
◆	010108	Teoria algoritmilor și mate...	Теория алгоритмов и дискретн...
◆	010109	Cibernetică matematică și c...	Математическая кибернетика ...
◆	010301	Astrometrie și mecanică ast...	Астрометрия и небесная механ...
◆	010302	Astrofizică și radioastronomie	Астрофизика и радиоастроном...
◆	010401	Aparate și metode ale expe...	Приборы и методы физического...
◆	010402	Fizică teoretică și matematică	Теоретическая и математичес...
◆	010403	Radiofizică	Радиофизика
◆	010404	Electronică fizică	Физическая электроника
◆	010405	Optică, fizica laserilor	Оптика, лазерная физика

The interface includes a sidebar with a tree view of folders like 'templ', 'css', 'custom', 'dispositions', etc., and a bottom status bar showing '1 of 351' items.

ANEXA 2

The screenshot shows the NeoSite interface with a log window open. The log contains the following information:

```

/tmp/iod/item/#
/tmp/iod/item/structure

2 templates built
12:24 HTTP/1.1 200 OK
14:20 http://81.180.65.122:8080/neo-service/ttV3?db=cnaa&action=build&opt=modified
Building Templates:

0 templates built
14:21 HTTP/1.1 200 OK
  
```

Below the log, a 'Queue' section shows a single entry:

Source	Url	Params	Plugin	Command
Build modified templa...	http://81.180.65.122:8080/...	db=c...	spiTmpl	

The interface also shows the same sidebar as in ANEXA 1 and a status bar at the bottom indicating 'Connecting to 81.180.65.122'.

ANEXA 3

The screenshot shows the NeoSite application interface. The 'Sql Explorer' window is active, displaying a query and its results. The query is:

```
1 select id, parent_id, title from site where tip=4 order by parent_id
```

The results are shown in a table with columns 'id' and 'title'. The data is grouped by 'parent_id'.

parent_id	id	title
parent_id : 898		
parent_id : 1195	11	Bazele teoretice ale informaticii; programarea calculatoarelor
	11	Organizarea ei programarea proceselor de calcul
	11	Tehnologii informacionale
	11	Modelare matematic, metode matematice, produse program
	12	Sisteme informacionale
parent_id : 1205		
	12	Securitate tehnologic ei informacionalr
	12	Armamentul ei tehnica militar
	12	Bazele securitroii nacioionale
	12	Economie militar
	12	Perdanonie si psihologie militar

The left sidebar shows a tree view of the site structure, including categories like 'Științe fizico-matematice', 'Științe chimice', etc.

ANEXA 4

The screenshot shows the NeoSite application interface. The 'Administration' window is active, displaying the configuration for the 'site' menu item. The 'Name' field is set to 'site' and the 'Ico' field is set to '12'. The configuration code is as follows:

```
1 <action>
2 <plugin>Site.npl</plugin>
3 <plugin_debug>C:/NeoNet/NeoSite/Site/Site.npl</plugin
4 <deepname>SiteTree</deepname>
5 <instance>run_or_new</instance>
6 <command>EnsureVisible</command>
7 <dock>LeftTop</dock>
8
9 <params>
10 <caption>CNAA</caption>
11 <table>site</table>
12 <lang>ro</lang>
13 <local>
14 <script>http://cnaa2/_build</script>
15 <builder>/gui</builder>
16 </local>
17 <remote>
18 <script>http://www.cnaa.acad.md/_build</scrip
```

The left sidebar shows a tree view of the site structure, including categories like 'menu', 'tools', 'actions', etc.

ANEXA 5

The screenshot shows the NeoSite application interface. On the left, a tree view displays the site structure under 'CNAAA' and 'Templates'. The main window shows a document editor with the following content:

```

1 <thesis czu="37.014.3 (043.2) ">
2 <ro>
3 <title>Fundamente teoretice și metodologice ale reformei
manageriale în învățământ</title>
4 <keywords>management educațional, managementul
competențelor, schimbare în educație, dezvoltare
organizațională, cultură organizațională, managementul
schimbării, matrice pe domenii de management, competențe
manageriale, stil de conducere, management de curriculum,
standarde de formare a managerilor școlari</keywords>
5 </ro>
6 <ru>
7 <title>Основы теории и методологии реформы менеджмента в
образовании</title>
8 <keywords>менеджмент образования, менеджмент компетенции,
изменение в образовании, развитие организации,
организационная культура, менеджмент изменений, матрица
  
```

The interface includes a menu bar (File, View, Options, Tools, CNAAA, Help), a toolbar with 'Save', 'Cancel', 'Refresh', and 'Build' buttons, and a status bar at the bottom.

ANEXA 6

The screenshot shows two windows from the NeoSite application. The 'Plugin Manager' window on the left lists installed plugins:

Name	Ver	Author
spiClient	3.5.7	Oleg Burlaca
spiTaskBar	2.0	Oleg Burlaca
spiCGI	1.20	Oleg Burlaca
spiFTP	0.99	Oleg Burlaca
spiSite	1.71	Oleg Burlaca
spiTmpl	2.1	Oleg Burlaca
spiedNeo	1.00	Oleg Burlaca
spiEdTmpl	3.0	Oleg Burlaca
spiEdTmpl_2	3.0	Oleg Burlaca
spiEdTmpl_1	3.0	Oleg Burlaca
spiNPLManager	2.00	Oleg Burlaca

The 'spiedNeo' configuration window on the right shows the 'Commands' tab with the following settings for the 'SaveFocusedControl' command:

- Name: SaveFocusedControl
- Allow Null In All Params: Y
- Check Param Count: Y
- Check Param Type: Y
- Hint: Called by nplTaskBar before switching to a new TaskWnd
- Params: (Empty table)

The background shows a file explorer with folders like 'docs', 'etd', 'generic', 'iod', 'item', and 'news', and a code editor with HTML/XML snippets.

ANEXA 7

NeoSite - Registration

Name:
Oleg Burlaca

Company:
IMI

Serial Key:
[] - [] - [] - [] - []

Warning: Previously entered serial key is invalid

Reasons: the hardware of your computer has changed
(for ex. you have installed a new processor or network card.)

Note: the serial key is valid for a single computer

In order to obtain a serial key, send NeoSite.nkd file
to the following email address:
oburlaca@neonet.md

Steps for obtaining the NeoSite.nkd file:
1. Enter your Name and Company
2. Press the 'Make NeoSite.nkd' button

Make NeoSite.nkd

Register Cancel

ANEXA 8

Comisia Nationala de Acreditare si Atestare - NeoSite

File View Options Tools CNAA Help

Sql Explorer Admin Server Logs Settings About Exit

CNAA

Fundamente teoretice i metodol... Funcțiile ei atribuțiile autor... Tip Manager

Title

Consilii științifice specializate

Acte norma

Codul c

doc_

pdf_

Dom

Noti

Politi

Fund

Acad

Insti

Acre

Ates

Prot

Asig

Asig

Stab

Stab

Rela

Disp

doc_

pdf_

Regulan

Regulan

Regulan

Regulan

New

Edit F2

Delete Del

Refresh F5

Add Files

Move Up

Move Down

Move In

Move Out

Customize

Find Ctrl+F

State

Fix ordering

Build

Proprieties Alt+Enter

Folder Ctrl+Ins

Page Ins

Institution

Consiliu

Speciality

Person Ctrl+B

Thesis Ctrl+T

Lucrare

Act normativ

Link

Dispozitie / Instructiune

Sedinta

News

Publicatie de profil Ctrl+E

Speciality Program

Manage... Shift+Ctrl+M

Fundamente teoretice i m... Funcțiile și atribuțiile autor... Tip manager

ANEXA 9

The screenshot shows the NeoSite web application interface. On the left is a file explorer showing a tree structure of folders and files. The main area displays the details for a 'Thesis' item, including a table with columns 'Ico', 'Name', 'Active', and 'Id'. Below the table is a code editor showing XML/HTML code for the thesis item, including tags like <tip>, <has_related>, <template>, <is_dir>, <shortcut>, <linkage mode="tree">, <rules>, <r parent_id="1464">, <fields>, <f name="info" editor="ComboBox">, <style>Fixed</style>, <items>, , , </items>, </f>, </fields>, and </tip>.

ANEXA 10

The screenshot shows the website for the National Council for Accreditation and Attestation (CNAA). The header includes the CNAA logo and navigation links for 'română', 'русский', and 'english'. The main content area is divided into several sections: 'CNAA' (with sub-links for Nomenclatura, Comisia de atestare, Comisia de acreditare, Dispoziții, instrucțiuni, and Acte normative), 'PREGĂTIREA CADRELOR' (with sub-links for Nomenclatura, Instituții, Consilii, and Teze), and 'POTENȚIALUL ȘTIINȚIFIC' (with sub-links for Conducători de doctorat, Deținători de grad, Doctoranzi, and Competitori). Below these are sections for 'Consiliul Național pentru Acreditare și Atestare', 'Acte Normative', 'Teze', and 'Documente relevante'. The footer contains copyright information, contact details, and a logo for 'Super.md'.

ANEXA 11

CNAA / Teze / Examinare /

Metodologia studierii elementelor de geometrie analitică prin intermediul calculatorului

Autor:	Liubov Zastinceanu
Gradul:	doctor în pedagogie
Specialitatea:	13.00.02 - Teoria și metodologia instruirii (pe discipline)
Anul:	2006
Conducători științifici:	Ilie Lupu Valeriu Cabac
Instituția:	Universitatea de Stat din Tiraspol
CSS:	D 36-13.00.02-26.02.04 Universitatea de Stat din Tiraspol

Statut
Teza a fost susținută pe 19 aprilie 2006 în CSS și se află în examinare la CNAA

Cuvinte Cheie
instruire asistată de calculator (I.A.C.), tehnologie informațională, tehnologie educațională, consecințe ale I.A.C., momente oportune de utilizare a I.A.C., programatură educațională, particularități de studiere a geometriei analitice, evaluarea automatizată, teste electronice, medii de instruire

Autoreferat
 - [0.34 Mb](#) / în română

Teza
CZU 514.12 (043.2)
 [1.20 Mb](#) / în română
158 pagini

Referenți Oficiali

- Anatol Gremalschi
- Aurelia Raileanu

Teze
Au fost elaborate 3 teze. (la specialitatea dată)
În examinare [2]:

- [Formarea competențelor de comunicare în limba română a elevilor alințivi în perioada preabecedară](#)
19 mai 2006
- [Tehnologii informaționale în procesul de predare-învățare a informaticii](#)
19 aprilie 2006


Arhiva tezelor:

Acte Normative

- [Lista actelor pentru dosarul de atestare deus la Consiliul Național pentru Acreditare și Atestare pentru conferirea / confirmarea gradului științific de doctor/doctor habilitat](#)
- [Reguli privind aspectul grafic al tezelor de doctor/doctor habilitat și al autoreferatelor/referatelor științifice](#)
- [Publicațiile ce seiau în considerare la prezentarea tezelor de doctor/doctor habilitat](#)
- [Referenții oficiali](#)
- [Sustinerea tezelor de doctorat](#)
- [Statutul juridic al cercetătorului științific](#)
- [Regulamentul privind organizarea și desfășurarea](#)

Adnotare
Cercetarea realizată este dedicată unei probleme practice și teoretice actuale: stabilirea bazelor teoretice și condițiilor metodologice de elaborare și utilizare a programelor de instruire în procesul de studiu al matematicii.
În rezultatul cercetării a fost elaborată metodologia utilizării programelor computaționale de instruire în procesul de studiu al elementelor de geometrie analitică în liceu, un pachet de programe computaționale de autor pentru realizarea acestei metodologii, ghidul de utilizare a acestui program.

ANEXA 12









 **Universitatea de Stat din Moldova (USM)**

Adresa: str. Mateevici, 60, Chișinău, MD-2009
Tel: (373 2) 57-77-90
Fax: (373 2) 57-77-90
Web: www.usm.md

Potențialul Științific
În proces de acumulare a informației...

Structura
Facultăți, catedre, laboratoare [...]

Documente

- [0.41 Mb](#)  Aviz de evaluare
- [0.50 Mb](#)  Aviz de evaluare-profil-Probleme actuale ale informaticii aplicate
- [0.50 Mb](#)  Aviz de evaluare-profil-Matematica Teoretică și Aplicată
- [0.37 Mb](#)  Aviz de evaluare-profil-Tehnologii nonpoluante și materiale noi cu proprietăți utile pentru economia națională
- [0.27 Mb](#)  Aviz de evaluare-profil-Fizica și tehnologia sistemelor, structurilor și dispozitivelor optoelectronice semiconductoare
- [0.50 Mb](#)  Aviz de evaluare-profil-Matematica Teoretică și Aplicată
- [0.39 Mb](#)  Aviz de evaluare-profil-Probleme social-politice ale Republicii Moldova
- [0.20 Mb](#)  Aviz de evaluare-profil-Modernizarea învățământului superior din Republica Moldova în contextul integrării europene

Doctoranzi
În proces de acumulare a informației...

Competitori
În proces de acumulare a informației...

Teze
Au fost elaborate 51 teze, inclusiv 7 de doctor habilitat.
În examinare [18]:

- [Studiu asupra verbelor eventive în limba română](#)
12 mai 2006
- [Fenomenul reduplicării în limba română](#)
12 mai 2006
- [Efectele cinetice determinate de interacțiunea electronilor cu oscilații acustice în heterostructurile cuantice](#)
26 aprilie 2006
- [Conceptul de suveranitate în dreptul internațional contemporan](#)
15 aprilie 2006
- [Contribuții la geneza halstattului tracic timpuriu în spațiul carpato-danubiano-pontic](#)
14 aprilie 2006

Arhiva tezelor:

Specialități Științifice
Lista specialităților grupate după domeniul științei. Specialitățile evidențiate semnifică existența unui consiliu în această instituție.

- [+] **Științe fizico-matematice** [11]
- [+] **Științe chimice** [4]
- [+] **Științe biologice** [11]
- [+] **Științe tehnice** [2]
- [+] **Științe istorice** [3]
- [+] **Științe economice** [4]
- [+] **Științe filosofice** [3]
- [+] **Științe filologice** [9]
- [+] **Științe geografice** [1]
- [+] **Drept și administrație de stat** [5]
- [+] **Științe pedagogice** [2]
- [+] **Științe psihologice** [1]
- [+] **Științe sociologice** [1]
- [+] **Științe politice** [2]

Profiluri Acreditate

- Probleme actuale ale informaticii aplicate
- Matematica Teoretică și Aplicată
- Tehnologii nonpoluante și materiale noi cu proprietăți utile pentru economia națională
- Fizica și tehnologia sistemelor, structurilor și dispozitivelor optoelectronice semiconductoare
- Matematica Teoretică și Aplicată
- Probleme social-politice ale Republicii Moldova
- Modernizarea învățământului superior din Republica Moldova în contextul integrării europene
- Tracologie, probleme de etnogeneză și glotogeneză, relații interetnice și internaționale
- Lingvistică teoretică și aplicativă, istoria și teoria literaturii
- Biodiversitatea, securitatea biologică și utilizarea rațională a resurselor naturale

Consilii Științifice Specializate
În paranteze este indicat numărul tezelor din anul curent examinate în consiliul respectiv

- [D 30-01.01.01-25.12.03](#)
Analiză matematică
- [D 30-01.01.07-25.12.03](#)

ANEXA 13

Nomenclator – Științe fizico-matematice

Căutare

Indicați cifra specialității. ↵ De ex. 01_05_04

Științe fizico-matematice

[–] Matematica

01.01.01	Analiză matematică	4	1	
01.01.02	Ecuații diferențiale	4	1	
01.01.04	Geometrie și topologie	3	1	
01.01.05	Teoria probabilităților și statistica matematică	1		
01.01.06	Logică matematică, algebră și teoria numerelor	4	1	
01.01.07	Matematica de calcul	2	1	
01.01.08	Teoria algoritmilor și matematică discretă			
01.01.09	Cibernetică matematică și cercetări operaționale	2	1	1

[+] Astronomie

[+] Fizica

[–] Informatica

01.05.01	Bazele teoretice ale informaticii; programarea calculatoarelor	3	1	
01.05.02	Organizarea și programarea proceselor de calcul			
01.05.03	Tehnologii informaționale			
01.05.04	Modelare matematică, metode matematice, produse program	2	2	
01.05.05	Sisteme informaționale	1		

Teze

Au fost elaborate 14 teze, inclusiv 1 de doctor habilitat.

În examinare [6] :

- [Efectele cinetice determinate de interacțiunea electronilor cu oscilații acustice în heterostructurile cuantice](#)
26 aprilie 2006
- [Obținerea și studierea proprietăților fotoelectrice a heterostructurilor cu straturi subțiri a semiconductorilor II-VI](#)
5 aprilie 2006
- [Proprietățile optice ale straturilor subțiri și nanostructurilor în baza GaN și ZnO](#)
5 aprilie 2006
- [Procese de recombinare radiativă în disulfizi metalelor de tranziție 2H-Ws2 și 2H-MoS2](#)
15 martie 2006
- [Microstructure and Flux Flow Processes in Superconducting MgB2 Films](#)
1 martie 2006
- [Modele de control optimal discret și soluționarea jocurilor dinamice pe rețele](#)
26 ianuarie 2006

Arhiva tezelor:
 Selectați anul

Instituții

Instituțiile abilitate cu activitate de doctorat la specialități din domeniul dat (în paranteze este indicat numărul specialităților)

 Selectați instituția

ANEXA 14

Căutare: model (129)

Harta sitului

în paginile sitului în fișiere (pdf, doc)

Ați căutat: **model** [...] Rezultate 1-20 din 129

model: 84, modele: 144, modelelor: 59, modelele: 50, modelului: 46, modelul: 162, models: 1

1. [Teoria și metodologia evaluării în învățământul universitar](#)
... pe 25 martie 2005 și aprobată de CNAA evaluare, calitate, **model**, axă, cadru didactic, învățământ universitar, măsurare, gestiune, ... stabilirii fundamentelor teoretice ale evaluării, elaborării **modelului** de evaluare a cadrului dida...
Cuvinte Cheie evaluare, calitate, **model**, axă, cadru didactic, învățământ universitar, măsurare, gestiune, produs, procedură, proces, teleologie, structuralism, cibernetică, sistenism, docimologie, psihometrie, edumetrie, scop, obiect, destinat, surse de infor
2. [01_05_04 – Modelare matematică, metode matematice, produse program](#)
01.05.04 – Modelare matematică, metode matematice, produse program ... 01.05.04 - Modelare matematică, metode matematice, produse program Pașaportul ... specialității Formula specialitatii: st...
3. [Caracteristici fizico-chimice și procese în sistemele apă-materii în suspensie-sedimentele nistrului și prutului](#)
... sedimente acvatice, structura, proprietăți de suprafață, **modele** fizico-chimice. – 0.98 Mb / în română CZU ... suprafață s-a efectuat prin verificarea adecvanței unui șir de **modele** fizico-chimice. ... S-au analizat **modele** ...
Cuvinte Cheie sedimente acvatice, structura, proprietăți de suprafață, **modele** fizico-chimice.
4. [Fundamente teoretice și metodologice ale educației pentru familie](#)
... eficient, problematica lumii contemporane, noile educații, **model** comportamental, mediul familial, cultura familiei, ethosul pedagogic ... democratizarea relațiilor familiale, cultura toleranței, **model** pedagogic, repere conceptuale, competențe ...
Cuvinte Cheie
5. [Proiectarea curriculum-ului de consiliere școlară](#)
... rezultate, dezvoltare integrală, evaluare a curriculum-ului, **modele** de proiectare a curriculum-ului, proiectarea curriculum-ului, ... competențelor personale și sociale. Promovăm un **model** alternativ de proiectare, în baza

Potențialul Științific

- Conducători de doctorat
- Deținători de grad
- Doctoranzi
- Competitori