Management Intercultural
Volumul XIX, Nr. 38 (1 / 2017)

Botond BENEDEK,
Mihai-Constantin AVORNICULUI
Faculty of Economics and Business Administration, Babeş–Bolyai University of Cluj-Napoca

DESIGNING AN EVENT EXTRACTION SYSTEM

Case Study

Keywords
Event Extraction System,
Designing Methods,
UML

JEL Classification
M15

Abstract

In the Internet world, the amount of information available reaches very high quotas. In order to find specific information, some tools were created that automatically scroll through the existing web pages and update their databases with the latest information on the Internet. In order to systematize the search and achieve a result in a concrete form, another step is needed for processing the information returned by the search engine and generating the response in a more organized form. Centralizing events of a certain type is useful first of all for creating a news service. Through this system we are pursuing a knowledge - events from the Internet documents - extraction system. The system will recognize events of a certain type (weather, sports, politics, text data mining, etc.) depending on how it will be trained (the concept it has in the dictionary). These events can be provided to the user, or it can also extract the context in which the event occurred, to indicate the initial form in which the event was embedded.

INTRODUCERE

Dimensiunea Internetului şi conţinutul său nestructurat şi dinamic, precum şi natura sa multilingvă, fac din extragerea cunoştinţelor utile o problemă de cercetare provocătoare. Web-ul generează o cantitate mare de date în alte formate, care conţin informaţii valoroase. Astfel ar fi informaţia log-urilor serverelor Web despre şabloanele accesate de utilizatori care poate fi folosită pentru personalizarea informaţiei sau pentru îmbunătăţirea design-ului paginii Web.
Pentru a sistematiza căutarea pe Internet şi a obţine un rezultat într-o formă concretă este utilă o etapă în care se prelucrează informaţiile întoarse de către motorul de căutare şi se generează răspunsul într-o formă mai organizată (Avornicului, 2010).
Centralizarea evenimentelor de un anumit tip este utilă în primul rând pentru realizarea unor servicii de ştiri. Aceste servicii trebuie să ofere informaţii actualizate, dacă se poate chiar în timp real, despre un anume tip de evenimente (Han & Kamber, 2006), (Lin, et al., 2008).
Să luăm de exemplu evenimentele sportive. Un utilizator al acestui serviciu de ştiri poate să dorească să afle ce jocuri sportive se desfăşoară într-o anumita regiune de pe glob (Oraş, Ţară, etc.) dintr-o perioadă anume: în acel moment, cu o zi în urmă, sau săptămâna viitoare, etc. Toate acestea informaţii trebuie obţinute din informaţiile deja centralizate. Se pot obţine astfel date despre un anumit eveniment din mai multe surse. Deasemenea sursele (siturile web din care se extrag informaţiile) se pot completa unele pe altele în conţinutul de informaţii despre anumite evenimente.
Acest sistem urmărește extragerea de evenimente din documentele HTML. Sistemul va recunoaşte evenimentele de un anumit tip în funcţie de modul în care va fi antrenat.
Acest sistem urmăreşte sa fie de un real ajutor in cazul căutării informaţiei din mai multe pagini de web, când se doreşte o centralizare a evenimentelor de un anumit tip.
De exemplu, dacă se doreşte aflarea informaţiilor despre vreme din mai multe pagini de web (yahoo, google, situl CNN, etc) pentru o anumită zonă şi o anumită perioadă atunci sistemul se poate dovedi util (Avornicului, 2014).
Pentru fiecare tip de eveniment (vreme, sport, etc.) există câte un dicţionar de concepte. Construirea unui dicţionar de concepte este o sarcină care necesită multă muncă. Încercăm să realizăm un sistem care sa fie extensibil, astfel încât să permită, la adăugarea unui nou dicţionar, extragerea de evenimente specifice acestuia. La ora actuală prezintă un domeniu de interes construirea dicţionarelor de concepte în mod automat, utilizând algoritmi de învăţare.

MODELUL CONCEPTUAL

Pentru a putea realiza ceea ce ne propunem, sistemul (componentele) trebuie să se ocupe, în general, de trei aspecte (Avornicului, 2009):

- aducerea documentelor HTML de pe Internet şi salvarea lor într-o bază de date pentru a putea fi prelucrate ulterior (a)
- prelucrarea documentelor şi obţinerea informaţiilor necesare (b)
- oferirea utilizatorilor un mod de acces asupra informaţiilor colectate (c)

a) Pentru a se putea realiza acest aspect sistemul trebuie să primească o serie de adrese web pe care trebuie să le parcurgă. Această listă de adrese poate să fie dependentă de tipul de evenimente pe care dorim să le identificăm. Aplicaţia va parcurge recursiv această listă de adrese şi va salva în baza de date locală toate documentele întâlnite.
b) Prelucrarea documentelor şi extragerea informaţiilor necesare are la bază un dicţionar de concepte care descriu tipurile de evenimente. Acest dicţionar de concepte trebuie să fie flexibil pentru a putea identifica diferite variaţii de tiparuri de evenimente ce pot apărea în documente. Toate evenimentele identificate vor fi apoi stocate într-o bază de date pentru o referire ulterioară.
c) În final trebuie oferit utilizatorilor accesul la informaţiile extrase. De exemplu, pentru evenimente din sport, se poate prezenta utilizatorilor o listă cu evenimentele sportive ordonate după data la care vor avea sau au avut loc. Se va permite deasemenea şi o căutare a evenimentelor după diferite criterii. Astfel se va putea face un acces rapid la informaţia dorită, fără a fi necesară parcurgerea evenimentelor zi cu zi. Figura 1 prezintă relevant aspectele sistemului.
Utilizatorii vor trebui să acceseze aplicaţia prin intermediul Internetului. Astfel, dacă utilizatorii sunt umani, se vor oferi documente HTML cu informaţia dorită, iar dacă utilizatorii sunt programe atunci informaţia va putea fi transmisă sub o formă generică, document XML de exemplu, pentru a putea fi prelucrată cu uşurinţă (Markov & Larose, 2007).

CERINȚE FUNCȚIONALE

Fiind vorba de un serviciu, accesul la baza de date cu evenimentele colectate va putea fi disponibil oricărui utilizator prin intermediul Internetului. Bineînţeles că se poate restrânge această arie la utilizatorii înregistraţi să acceseze serviciul respectiv, de exemplu. Deci virtual numărul utilizatorilor care vor folosi acest serviciu este nelimitat (Kovács, 2012). Trebuie să se permită

deasemenea accesul simultan al mai multor utilizatori la baza de date cu evenimentele extrase.
Având în vedere că toate procesele consumatoare de timp şi putere de calcul se vor desfăşura offline (transparent pentru utilizatorii finali), timpul de răspuns al sistemului la cererile utilizatorilor trebuie să fie redus (de ordinul milisecundelor/secundelor).
Sistemul trebuie să fie capabil să proceseze un volum de date suficient, astfel încât, de exemplu în cazul evenimentelor sportive, să se poată face o actualizare a acestora din oră în oră (Lin, 2007).
Scopul sistemului este să ofere un răspuns cât mai complet la interogările utilizatorilor (Zaki, 2000). Teoretic, acest răspuns este corect si complet, însă în practică acest lucru este imposibil de realizat de un sistem, fără a se cunoaşte date despre formatul cunoştinţelor din text (adică modul de prezentare a informaţiei in pagina HTML). Din acest motiv indicatorii de eficienţă ai sistemului vor varia în funcţie de sursele din care se extrage informaţia. Ceea ce ne propunem este sa păstrăm aceşti indicatori între anumite limite astfel încât aplicaţia noastră să aibă o utilitate reală.
Indicatorii pe care sunt luate în considerare sunt:

- **acoperirea** - reprezintă raportul: numărul de rezultate corecte găsite / numărul de rezultate corecte existente;
- **precizia** – reprezintă raportul: numărul de rezultate corecte găsite / numărul total de rezultate găsite.

În ceea ce priveşte specificaţiile interfeţei cu utilizatorul, ceea ce se poate spune la momentul actual este că se permite utilizatorului introducerea de interogări în diverse formate şi, în acelaşi timp, controlarea modului de afişare a rezultatelor (Sommerville, 2010). Aspectul exact al interfeţei cu utilizatorul va fi stabilit in faza de testare cu scopul de a determina un mod de lucru cât mai ergonomic (Han & Kamber, 2006).

ASPECTE LEGATE DE PROIECTARE

Analiza şi proiectarea unui sistem informatic reprezintă una dintre cele mai importante etape ale realizării unui sistem. Dacă un sistem nu este bine gândit de la bun început, este foarte posibil ca, la implementarea sa sau la exploatarea sa, să constatăm că datele pe care ne aşteptăm să le furnizeze respectivul sistem nu pot fi obţinute, prelucrările durează mai mult decât ne-am fi aşteptat, situaţiile conţin date eronate care nu ne permit să luăm o decizie, sau că, sistemul, în loc să simplifice activitatea mai mult o îngreunează. Puţini sunt aceia care realizează că de fapt, toate aceste neajunsuri nu sunt datorate nici programatorilor şi nici implementatorilor sistemului informatic ci chiar celor care au „gândit" respectivul sistem. De multe ori, etapa de analiză şi modelare este „sărită" considerându-se ca fiind neproductivă şi consumatoare de timp. Într-adevăr, dacă privim activitatea de analiză şi proiectare având drept finalitate o serie de documente cu scheme şi descrieri cu privire la cam cum ar arăta sistemul în realitate, putem considera că atât timpul alocat cât si banii investiţi sunt nişte pierderi inutile. Dacă însă luăm în calcul faptul că după etapa de analiză şi proiectare echipele de dezvoltatori au un plan de abordare a problemei, multe dintre module sunt demarate în baza unei analize de optimizare a termenului de finalizare iar posibilitatea de a relua anumite aspecte este mult diminuată vom începe să acordăm atenţia cuvenita acestei prime etape (Jing & McKeown, 1999). Ceea ce trebuie remarcat este faptul că, adevărata returnare a investiţiei în analiză şi proiectare nu are loc în momentul dezvoltării ci în ultimele etape: mentenanta si exploatarea. Un proces de analiză şi proiectare derulat după toate rigorile metodologice conferă o temelie robustă a ceea ce urmează să fie făcut şi asigură continuitate sistemului în momentul în care acesta va trebui extins sau îmbunătăţit. Fără o documentaţie atent întocmita, respectivul sistem va avea serios periclitate acele etape care au un impact decisiv asupra clientului. Dacă va apărea o problemă, aceasta nu va putea fi soluţionată în mod optim fără o cunoaştere în detaliu a arhitecturii sistemului. Se va încerca rezolvarea unui aspect semnalat de către utilizatori dar modificările introduse pentru rezolvarea acelui aspect se vor propaga în lanţ şi vor pune echipa în faţa unor situaţii uneori imposibile (Avornicului, 2010).
Trebuie mentionat că, proiectarea unui sistem este influenţată de o serie de factori. De multe ori, proiectarea unui sistem depinde de experienta proiectantului sau a echipei de proiectanţi. Există chiar un moto care afirmă că, cel de-al doilea sistem conceput de un proiectant pentru un anumit domeniu este întotdeauna mai bun decât primul. Afirmatia are un suport real dat fiind faptul că, echipa de proiectanţi deja cunoaşte anumite aspecte şi evită greşelile făcute anterior. Un alt aspect ce trebuie luat în considerare este acela că, anumite metodologii de proiectare se pretează mai bine unui anumit tip de problemă, echipa de proiectanţi reuşind să obţină rezultate bune utilizând mult mai puţine resurse. Deşi ridicate la un nivel de formalizare suficient de mare, anumite metodologii sunt preferate altora din motive legate strict de derularea unui anumit proiect. De exemplu, este de dorit ca o echipă de proiectanţi să recurgă la utilizarea UML (Avornicului, 2014). Totuşi, dată fiind experienţa pe care componenţii respectivei echipe o au în utilizarea metodelor sistemice si intervalul de timp extrem de redus în care trebuie parcursă etapa de proiectare, este posibil ca aceştia să recurgă la o metodă cunoscută lor. Echipa respectivă ar fi trebuit să facă un important salt calitativ dublat de un effort sustinut de

documentare pentru a putea prelua din plin avantajele noilor concepte care stau la baza UML cum ar fi: polimorfismul, evenimentele, interfaţa, mesajele, componentă comportamentală etc. Pentru aceeaşi problemă sunt oferite soluţii diferite. Diferenţa este făcută pe considerente care ţin de aspecte precum (Avornicului, 2009):

- cât de flexibilă este soluţia adoptată
- cât de rapidă este în prelucrarea datelor
- cât de mulţi utilizatori concurenţi suportă respectiva soluţie
- cât de costisitoare sunt dispozitivele hard necesare rulării unei anumite soluţii
- cât de extensibilă este respectiva soluţie
- care este efortul pe care-l necesită pentru a o adapta unor cerinţe specifice
- cât de mare este redundanţa datelor sau care este gradul de normalizare etc..

De multe ori, o anumită soluţie reflectă personalitatea celui care proiectează, felul în care respectiva persoană sau grup de persoane percepe realitatea. Acest aspect i-a determinat pe mulţi să spună că modelarea ca si programarea este o artă (Chen & Wei, 2002). De exemplu unii proiectanţi, atunci când au modelat conceptul de partener au văzut necesar să menţioneze dacă acesta este furnizor, client sau ambele (Cutting, et al., 1992).

DIAGRAMA USE CASE PENTRU SISTEMUL DE EXTRAGERE DE EVENIMENTE

Acestea au rolul de a reprezenta într-o formă grafică funcţionalităţile pe care trebuie să le îndeplinească sistemul informaric în faza finală. De aceea modelul realizat de ele împreună cu descrierea succintă a fiecărui caz de utilizare determinat se numeşte model al cerinţelor.

Cazurile de utilizare arată comportamentul sistemului sau a unei părţi din sistem şi este o descriere a unui set de secvenţe de acţiuni, incluzând variante pe care un sistem le execută pentru a produce un rezultat observabil pentru un actor (Raffai, 2005), (Sommerville, 2010).

Diagramele acestea sunt formate din două categorii de entităţi (actori şi cazuri de utilizare) şi relaţiile dintre acestea (Figura 2).

Ele implică interacţiunea dintre actori şi sistem, iar actorii pot fi oameni sau sisteme automate.

Comportamentul unui caz de utilizare se poate specifica prin descrierea fluxului de evenimente. Acestea trebuie să indice cum şi când începe şi se încheie cazul de utilizare, când interacţionează cu actorii, ce obiecte sunt schimbate, fluxul de bază şi cel alternativ al comportamentului.

În continuare prezentăm detaliat funţia fiecărui modul (Figura 3):

- **Baza de Date cu Adrese de Pagini**: Păstrează lista paginilor Web pentru un anumit domeniu, pe care se va face căutarea de evenimente
- **Baza de Date de Pagini**: Păstreaza paginile Web, indexate şi categorizate după domenii
- **Baza de Date cu Evenimente**: Păstrează evenimentele identificate, indexate după domenii, timp, locaţii şi alte criterii
- **Baza de Date cu Dictionarul de Concepte**: Păstrează conceptele care vor fi căutate, indexate după domenii
- **Dezcărcare pagini**: Aduce periodic pagini Web de pe Internet de la adresele prezente in Baza de Date cu Adrese de Pagini, şi le stochează în Baza de Date de Pagini. Acest modul poate aduce in mod recursiv şi paginile indicate de link-urile din pagina curentă, de pe acelasi server, până la o anumită adâncime de aducere.
- **Segmentare**: Realizează impărţirea textului din paginile Web din Baza de Date cu Pagini în segmente
- **POST**: Acest modul are funcţia de a prelua segmentele şi a le adnota cu parţile de vorbire corespunzătoare cuvintelor din segment.
- **Extragere evenimente**: Extrage evenimente din segmentele adnotate, primite de la modulul POST, pe baza conceptelor din Dictionarul de Concepte şi a WordNet-ului, depunandu-le în Baza de Date cu Evenimente
- **Creare Dictionar Concepte**: Construieste conceptele specifice unui anumit domeniu, prin algoritmi de învăţare, apoi le adauga în Baza de Date cu Concepte.
- **WordNet**: Furnizeaza relaţii între cuvinte de natură semantica
- **Interfata cu aplicatia client**: Primeşte cereri de căutare evenimente, pe care le obţine din Baza de Date cu Evenimente
- **Interfata client**: Asigură interfaţa cu utilizatorul, introducerea de query-uri de evenimente, setarea de opţiuni, afisarea rezultatelor dupa diverse criterii

DIAGRAMA DE CLASĂ

Ele fac parte din categoria diagramelor statice şi descriu structura internă a sistemului informatic prin identificarea claselor, a atributelor şi a operaţiilor acestora şi a relaţiilor dintre clase. Construcţia lor are loc în faza de elaborare a sistemului informatic, fiind cele mai importante în această fază. Selectarea claselor necesită anumite deprinderi. O abordare metodică a determinării claselor ce modelează soluţia unei probleme informatice este aceea de a extrage substantivele esenţiale din documentul de specificaţie. O metodă mult mai rapidă este aceea de a extrage toate substantivele care apar în diagrama de cazuri de utilizare (atât în denumirile actorilor cât şi a cazurilor de utilizare).

Clasele nu sunt elemente izolate ale sistemului. Astfel trebuie să identificăm relaţiile între clase (Figura 4).

CONCLUZII

Cercetările teoretice şi practice privind proiectarea şi realizarea sistemelor de extragere de evenimente au început să se focalizeze tot mai mult pe utilizarea limbajului UML în modelarea acestor sisteme.

Tehnologia pentru extragerea evenimentelor din pagini web şi arhitectura sistemului informatic influenţează semnificativ felul în care trebuie proiectat şi implementat sistemul de extragere de evenimente.

În acest sens propunem utilizarea modelelor UML pentru definirea cerinţelor şi determinarea contextului informaţional, astfel încât să poată fi creat un scenariu cât mai complet de ceea ce va trebui să acopere sistemul de extragere de evenimente.

De asemenea propunem utilizarea în faza de proiectare, a tehnicii prototipizării prin care să se permită comunicarea directă dintre utilizator, sistem şi proiectantul acestuia. Considerăm, că o determinare bine documentată şi cât mai apropiată de ceea ce se doreşte de la viitorul sistem, încă din faza de concepere, asigură succesul elaborării, implementării şi utilizării sistemului.

Recomandăm abordarea proiectării şi implementării sistemului de extragere de evenimente prin fluxul iterativ al procesului unificat, ceea ce va determina ca versiunea executabilă de bază să fie cât mai stabilă şi să acopere cât mai bine cerinţele utilizatorilor.

BIBLIOGRAFIE

[1] Avornicului, M., 2009. Data mining în Internet.

[2] Avornicului, M., 2010. *Informatikai rendszerek tervezése és menedzsmentje.* Kolozsvár: ÁBEL Kiadó.

[3] Avornicului, M., 2014. Considerations On Objective Methods For Developing Applied Event Extraction Systems. *SEA-Practical Application of Science,* pp. Volume II, Issue 2 (4), pp. 447-456.

[4] Chen, G. & Wei, Q., 2002. Fuzzy association rules ant the extended mining algorithms. *Information Sciences,* p. pp. 201–228..

[5] Cutting, D. R., Pedersen, J. O., Karger, D. & Turkey, J., 1992. *A cluster-based approach to browsing large document collections.* Copenhagen, s.n., p. pp. 318–329.

[6] Han, J. & Kamber, M., 2006. *Data Mining: Second Edition Concepts and Techniques.* s.l.:Morgan Kaufman Publishers.

[7] Jing, H. & McKeown, K. R., 1999. *The decomposition of human-written summary sentences.* Berkeley, s.n., p. pp. 129–136.

[8] Kovács, G., 2012. Productivity improvement by lean manufacturing philosophy. *ADVANCED LOGISTIC SYSTEMS: THEORY AND PRACTICE,* p. pp. 9–16..

[9] Lin, B., 2007. Web Data Mining – Exploring Hyperlinks, Contents an Usage Data. *Springer.*

[10] Lin, T. Y., Xie, Y., Wasilewska, A. & Lian, C. J., 2008. Data mining: Foundations and Practice. *Springer.*

[11] Markov, Z. & Larose, D., 2007. *Data Mining the Web.* s.l.:John Wiley & Sons.

[12] Raffai, M., 2005. *UML 2 modellező nyelv.* s.l.:Palatia Nyomda és Kiadó.

[13] Sommerville, I., 2010. *Software Engineering, 9th Edition.* s.l.:Addison-Wesley.

[14] Zaki, M., 2000. *Parallel and distributed data mining: An introduction..* s.l.:Springer-Verlag.

ANEXE

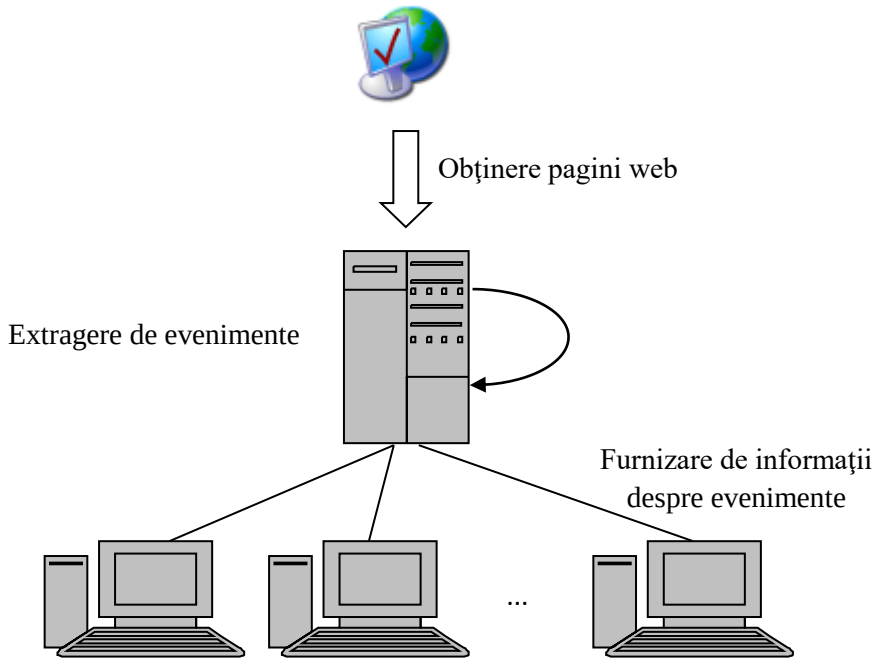

Figura 1 Aspectul sistemului

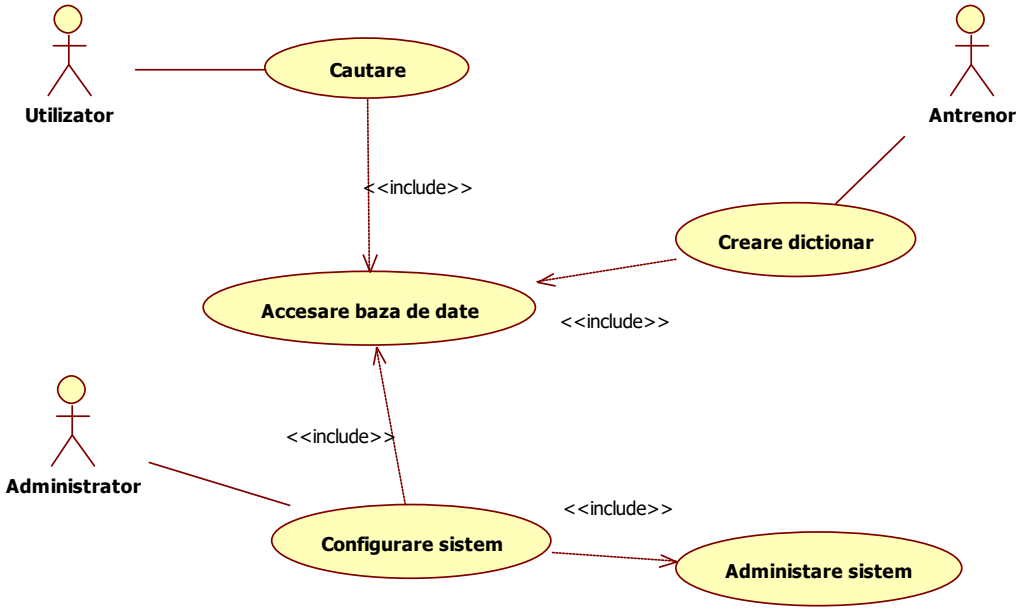

Figura 2. Diagrama ”use case”

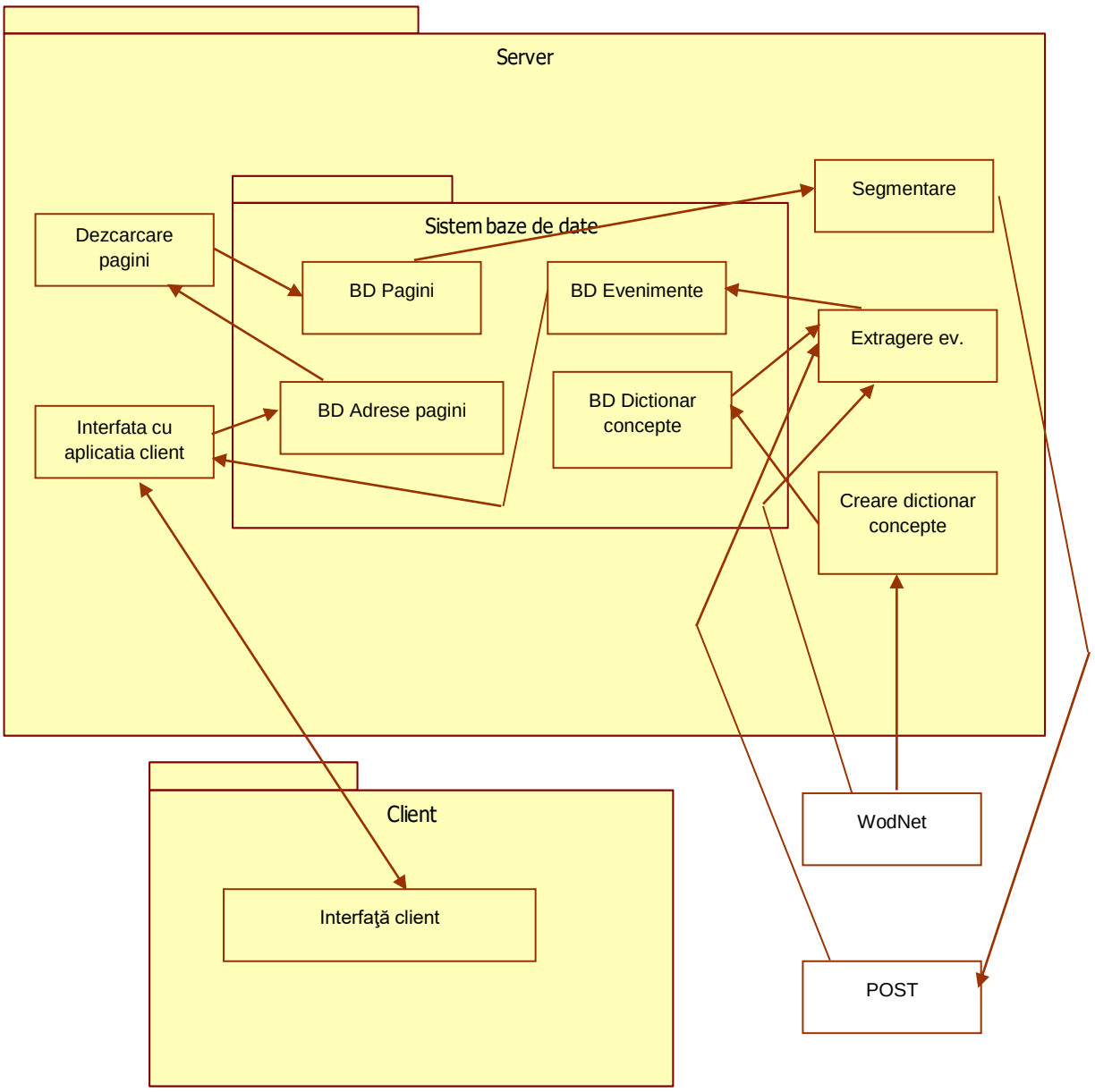

Figura 3. Dependenţe între module

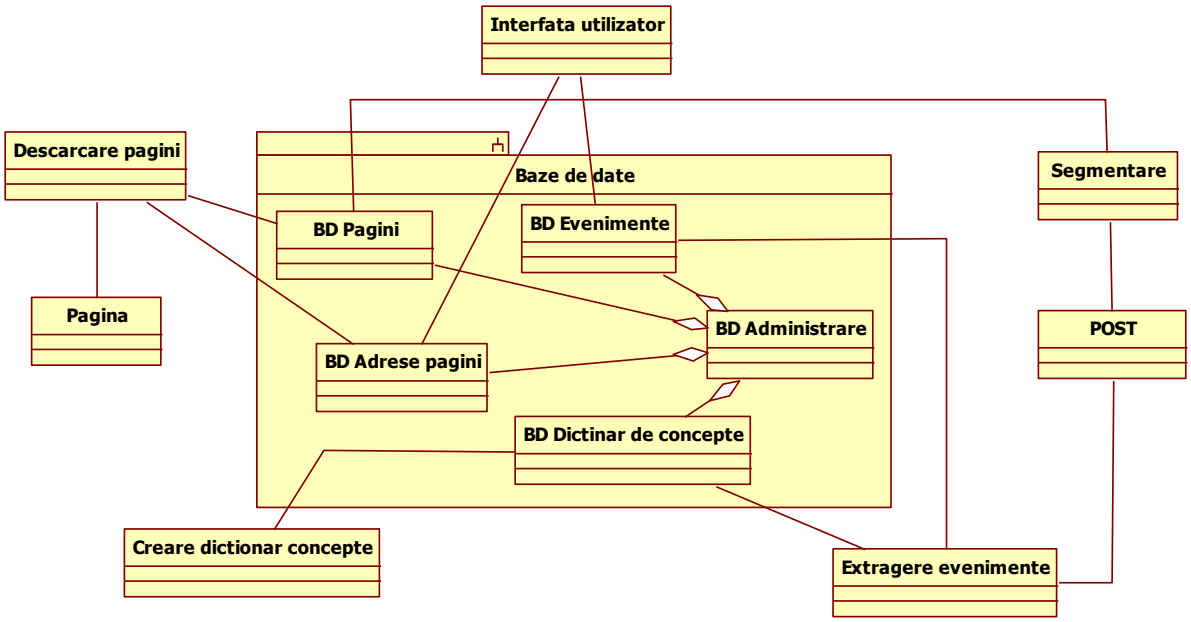

Figura 4. Relaţiile între clase