Utilitatea metodologiei UML în proiectarea bazelor de date

Asist. Catalin STRÎMBEI
Facultatea de Economie si Administrarea Afacerilor, Universitatea „Al. I. Cuza”, Iasi
linus@uaic.ro

UML is the modeling standard involved in the design of software systems based on object technology. Can be this visual modeling language any useful for design of classical (relational) databases? Analyzing the UML artifacts and stages of modeling process we can gather the information concerning all critical decision in database design and we can seamless integrate the database life cycle in the workflow of unified process. Each level of modeling (conceptual, logical and physical) is supported by key UML elements produced in different stages of design process of overall system. The major problems in this approach are related with the mapping of object concepts in physical (relational) constructors. But the structural model of UML has its roots in entity-relational model (because of OMT) so the basic techniques used in this field can be applied in “UML design for databases” too. Thus UML techniques can be successfully used in the database design process and the major advantages of this approach consist in increasing of coherency degree between overall system design and specifically database design, and improved teamwork efficiency.
Keywords: *object modeling, database design, UML language and process, database lifecycle, object/relational mapping.*

Motivatii în sprijinul utilizarii UML pentru modelarea bazelor de date

Complexitatea deosebita a sistemelor actuale a determinat abordarea pe scara larga a principiilor orientarii obiectuale nu doar în ceea ce priveste programarea ci si în proiectarea lor, lucru care a avut consecinte si asupra proiectarii bazelor de date. În general modelarea orientata obiect (gen UML) este privita ca apanajul doar al proiectarii (sau dezvoltarii) aplicatiilor orientate obiect. Aceasta abordare se dovedeste însa **limitata** având în vedere doua aspecte:

1. *Pe de o parte proiectarea aplicatiilor (orientate obiect) cu baze de date separata de proiectarea bazei de date va produce un **nivel suplimentar de mapare** a entitatilor care figureaza în cele doua arii **la nivel conceptual**.* Proiectarea unui sistem integrat din care sa rezulte concertat ce obiecte vor fi stocate persistent în baza de date, ce obiecte vor face parte din package-urile sau modulele aplicatiilor rezidente sub forma componentelor si modul cum vor interactiona între ele se poate dovedi o alternativa mai buna. Singurul substrat suplimentar (care poate fi modelat si el prin stereotipuri UML) ar fi cel al maparii entitatilor conceptuale în structurile oferite de modelul logic al bazei de date pentru stocare si regasire.
2. *Pe de alta parte modelele logice ale bazelor de date relationale comerciale propun astazi o serie de concepte noi cum sunt procedurile stocate si declansatoarele care nu fac parte din modele relationale traditionale si de obicei nu sunt luate în considerare în faza proiectarii bazei de date.* Valorificarea direct în proiectarea conceptuala a acestora se poate face folosind concepte orientate obiect si mecanismele de extensibilitate ale unui limbaj de modelare cum este UML.

Motivatiile pentru care metodologia UML ar fi potrivita si pentru specificarea modelelor implicate în proiectarea bazelor de date (fata metodele traditionale) ar putea fi rezumate astfel:

- modelul structural al UML încorporeaza toate conceptele modelului entitate-relatie (cel mai popular model pentru proiectarea conceptuala a bazelor de date) plus conceptele specifice modelelor semantice (generalizare, agregare) si abordarii orientate obiect (mostenire, polimorfism, supraîncarcare). Limbajul UML poseda mecanismele de ex-

tensibilitate (restrictii, stereotipuri, note, etichete) care fac posibila specializarea modelului în functie de aspectele particulare ale sistemelor cu baze de date (flexibilitate sporita în modelare);

- conceptele limbajului UML propun un vocabular comun între modelele rezultate din proiectarea sistemului ca ansamblu si cele din proiectarea specifica bazei de date;
- metodologia UML (procesul unificat) favorizeaza munca în echipa: scopul folosirii UML pentru modelarea proceselor afacerii, dezvoltarii aplicatiilor si modelarii bazei de date este integrarea echipelor de dezvoltare pentru a împiedica construirea unei arhitecturi fara implicarea corespunzatoare în acest proces a tuturor echipelor care concura la realizarea sistemului în ansamblu;
- integrarea instrumentelor de dezvoltare din diferitele arii de interes din care rezulta arhitectura sistemului pe baza unei platforme conceptuale comune. Instrumentele CASE bazate pe metodologii obiectuale ofereau initial suport pentru proiectarea generala a sistemului si mai putin pentru proiectarea de detaliu. Cu alte cuvinte arhitectura sistemului era sprijinita prin instrumente *upper CASE* destul de slab integrate cu cele *lower CASE*. Adoptarea UML ca limbaj de modelare a produs în acest sens doua mutatii:
  - aparitia de noi instrumente CASE care ofera suport pentru întreg ciclu de dezvoltare pâna la obtinerea unei versiuni complete a sistemului (gen Rational Rose);
  - integrarea de noi facilitati dedicate modelarii în mediile de dezvoltare integrate comerciale (gen suitele de dezvoltare de la Borland, Microsoft sau Oracle) cu posibilitati de proiectare vizuala din care sa rezulte componentele continând codul sursa sau scripturile pentru generarea schemelor bazelor de date.

Integrarea proiectarii bazelor de date în ciclul de proiectare caracteristic metodologiei *UML*

O privire de ansamblu a modului în care diagramele UML care descriu modelele prezentate în figura 1 au implicatii asupra proiectarii bazei de date poate fi prezentata astfel:

Tabel 1. Principalele diagrame UML si rolul lor în specificatiile modelelor bazei de date, adaptare dupa [NAIBURG2001] si [BOOCH99]

Diagrame	Descriere
Cazuri de utilizare	Diagrama cazurilor de utilizare reprezinta un model al functiilor vizate ale sistemului prin care se sustin procesele afacerii (business proces). Acest model serveste ca un contract între clienti si cei care dezvolta sistemul
Interactiuni	Diagramele de interactiune (diagramele de secvente sau colaborare) prezinta modul cum interactioneaza obiectele în cadrul sistemului. Ele pot fi utilizate pentru a identifica si defini cererile (în special sub forma interogarii si actualizarii bazei de date) care vor afecta baza de date si care pot ajuta chiar la clarificarea unor aspecte ce tin de detalii ale schemei fizice (cum ar fi necesitatea optimizarii anumitor interogari prin crearea indexurilor necesari)
Activitati	Diagramele de activitati prezinta în principal fluxul proceselor prin care se prezinta o imagine de nivel înalt a afacerii si a modului cum opereaza
Statechart	Diagramele de stare (statechart) surprind comportamentul dinamic al sistemului sau al obiectelor din cadrul acestuia
Clase	Diagramele de clase reprezinta *modelele logice* care surprind structura de baza a sistemului
Baza de date	Diagrama bazei de date descrie structura bazei de date incluzând aici tabele, coloane etc.
Componente	Diagrame de componente pot prezenta structura fizica a bazei de date cu referire la sistemul de gestiune a bazei de date, structurile de organizare a spatiului de stocare la nivelul intern al bazei de date (de exemplu spatii pentru tabele – tablespace-uri - în cadrul bazelor de date Oracle). De asemenea,

	aceste diagrame pot include si aplicatiile plus interfetele acestora pentru accesul la bazele de date.
Instalare (deployment)	Diagramele de instalare sau desfasurare (deployment diagrams) prezinta configuratiile hardware folosite pentru aplicatiile cu baze de date

Pentru a avea o imagine a modului în care proiectarea bazelor de date se înscrie în ciclul de dezvoltare al procesului unificat propus de Jacobson, Booch si Rumbaugh putem reconstitui imaginea (tinând cont de modelele si diagramele propuse de-a lungul ciclului de dezvoltare din figura 1):

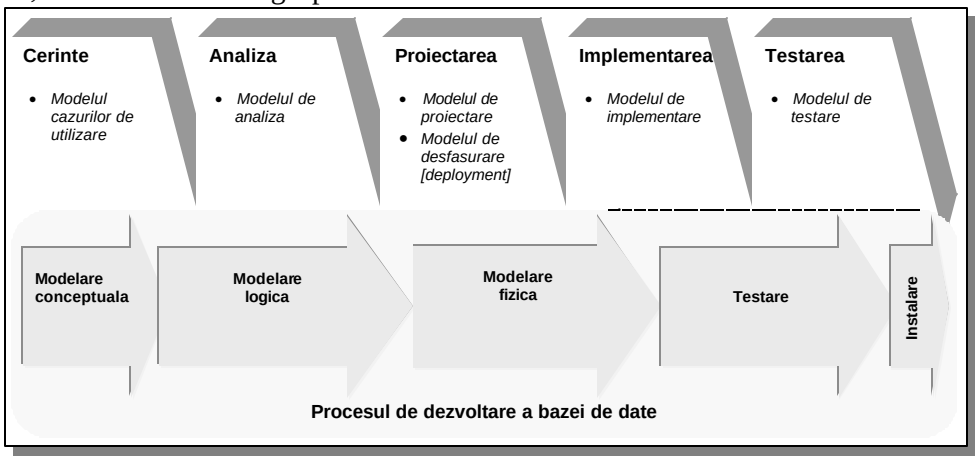

Fig. 1. Integrarea procesului de dezvoltare a bazei de date cu fluxul de activitati de baza caracteristice procesului de proiectare unificat propus de metodologia UML

Modul în care artefactele care formeaza universul obisnuit al unui proiectant de baze de date rezulta din procesul de dezvoltare propus în figura 1 (figura 2).

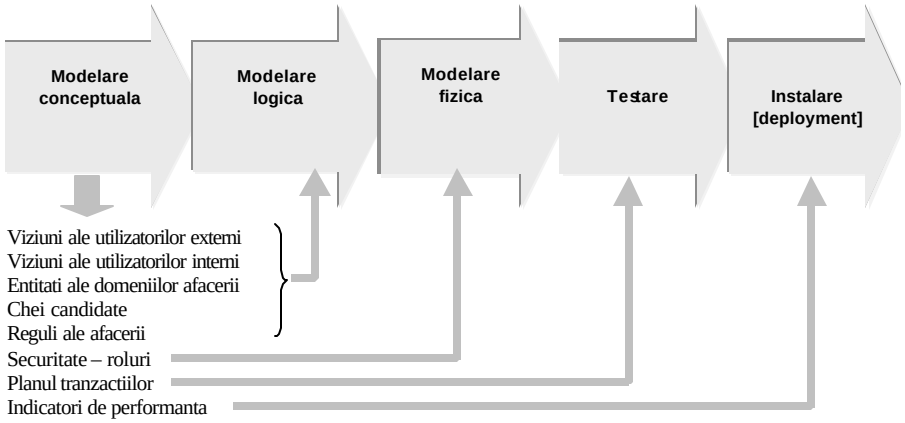

Fig. 2. Procesul de dezvoltare al bazei de date si artefactele esentiale, adaptare dupa [NAIBURG2001]

Modelarea conceptuala

Metodologia UML propune în privinta modelarii construirea unui *model al afacerii* (*Business Model*) care este constituit din doua parti:

1. Modelul cazurilor de utilizare ale afacerii care se concentreaza asupra modului cum este privita afacerea de catre actorii externi si asupra interactiunilor acestora. Modelul cazurilor de utilizare ale afacerii prezinta procesele afacerii (business process) în ter-

menii cazurilor de utilizare ale afacerii (business use cases) si actorilor afacerii (business actors). Modelul cazurilor de utilizare ale afacerii este descris prin intermediul diagramelor cazurilor de utilizare. O astfel de diagrama arata un set de cazuri de utilizare împreuna cu actorii si relatiile lor [JACOBSON99]. Un caz de utilizare prezinta o secventa (sau mai multe secvente alternative) de activitati care au loc în contextul respectiv. Prin urmare cazurile de utilizare ale afacerii vor fi însotite de diagramele de secvente corespunzatoare.

2. Modelul obiect al afacerii se concentreaza asupra modului cum utilizatorii din interiorul sistemului (afacerii) realizeaza procesele afacerii. Modelul afacerii (business model) este descris intern prin modelul obiect al afacerii (object-business model) care arata cum fiecare caz de utilizare al afacerii (business use case) este realizat de catre un set de *lucratori (workers)* care utilizeaza un set de *entitati ale afacerii (business entities)* si *unitati de lucru (work unit)* [JACOBSON99].

Diagrama cazurilor de utilizare ale afacerii si diagramele de secvente care le însotesc dau o prima idee asupra actorilor externi (business actors), actorilor interni (business workers) si entitatilor care intervin si care vor fi preluate în *diagramele de clase* prin care sunt prezentate cazurile de utilizare din modelul cazurilor de utilizare ale afacerii. De asemenea seturile diagramelor de secvente initiale asociate fiecarui caz de utilizare (diagrame care prezinta viziunea externa a desfasurarii proceselor afacerii) sunt detaliate în noi seturi de diagrame de secvente care prezinta fluxul de activitati dintre actorii externi, lucratorii si entitatile afacerii din perspectiva interna.

Pe baza entitatilor afacerii (business entities) desprinse din modelul obiect al afacerii se poate realiza un *model conceptual preliminar* printr-o diagrama de clase care sa prezinte inclusiv relatiile (asociatii, generalizari, agregari) dintre aceste entitati. Diagramele de secvente dau o idee generala asupra modului în care aceste entitati sunt invocate si intra în interactiune cu actorii externi si lucratorii din domeniul afacerii. Prin urmare din cele doua modele ale afacerii (cazurile de utilizare si obiecte) *proiectantul bazei de date* poate face deja consideratii asupra:

- *informatiei necesare a fi stocata în baza de date si cui îi va fi destinata* aceasta (modelul cazurilor de utilizare si diagramele de activitati pentru interactiunile acestora cu sistemul);
- *entitatile importante din domeniul afacerii, reguli ale afacerii care trebuiesc respectate, informatii privind accesul la date,* din modelul obiect al afacerii, adica diagramele de clase ce includ actori externi (business actors), actori interni (business workers), entitati din domeniul afacerii (business entities) si diagramele de secvente pentru cazurile de utilizare ale afacerii

Modelarea logica

Metodologia UML presupune în faza stabilirii cerintelor (premergatoare analizei) transformarea modelelor afacerii în modele ale sistemului adica obtinerea *modelului cazurilor de utilizare ale sistemului* din modelul cazurilor de utilizare ale afacerii. Un model al cazurilor de utilizare ale unui sistem contine actorii, cazurile de utilizare si relatiile dintre ele. Modalitatea practica de a prezenta un astfel de model o constituie *diagramele cazurilor de utilizare.* Modelul cazurilor de utilizare ale sistemului constituie în fapt specificarea cerintelor care trebuie satisfacute de respectivul sistem. Prin urmare la acest moment proiectantul bazei de date va avea clarificate principiile de baza ale arhitecturii datelor.

Specificarea cerintelor se realizeaza în fapt printr-o serie de activitati legate de *traducerea, interpretarea si structurarea* modelului (tabelul 2).

De asemenea, entitatile afacerii din diagramele de clase ale cazurilor de utilizare ale afacerii vor fi preluate în modele de analiza si proiectare. Acest proces de transformare poate presupune si adaugarea unor cazuri de utilizare, actori sau entitati noi. Formarea unui model al sistemului ca atare începe din faza de analiza, iar principalele artefacte care rezulta sunt *diagramele de clase, diagramele de secvente* si *diagramele de stare.* Pornind

de la modelul cazurilor de utilizare obtinut din transformarea modelului afacerii, în faza de analiza si proiectare sunt construite *diagramele de clase* care vor reprezenta punctul central sau fundatia atât pentru proiectantii aplicatiilor cât si pentru proiectantii bazei de date, fiindca aceste diagrame stabilesc entitatile principale din sistem si relatiile dintre ele atât pentru aplicatii cât si pentru entitatile din baza de date.

Tabel 2 Transformarea modelului afacerii în modelul cazurilor de utilizare

În modelul afacerii		*În modelul cazurilor de utilizare ale sistemului*
Cazurile de utilizare ale afacerii	*devin*	Subsisteme
Actorii afacerii	*devin*	Actori
Lucratorii afacerii (business workers)	*devin*	Actori si cazuri de utilizare
Activitatile lucratorilor afacerii	*devin*	Cazuri de utilizare

În etapa de analiza si faza preliminara a etapei de proiectare se construieste *un model logic al sistemului* la care participa atât proiectantii de aplicatii cât si proiectantii bazei de date. Prin urmare diagramele de clase dezvoltate la acest moment vor constitui suportul comun pentru sincronizarea muncii celor doua echipe. Diagramele de clase constituite pe baza modelelor obiect anterioare sunt însotite de diagramele de catre secvente care vor include controversatele *reguli ale afacerii* (business rules), astfel încât, înainte ca echipele de proiectare a aplicatiilor si cele de proiectare a bazei de date sa se separeu, se poate face o analiza si o distribuire a modalitatilor de implementare a acestor reguli evitându-se eventuale le conflicte. Principalele concepte implicate sunt:

- *diagramele de stare* (statechart) care prezinta comportamentul dinamic al claselor;
- *clasele de control* sau active (control classes) care controleaza comportamentul uneia sau mai multor clase;
- *clasele pasive* care nu sunt active si deci nu sunt implicate în actiuni de control privind sistemul.

Prin urmare prin diagramele de clase vor fi specificate care entitati vor rezida în baza de date si ce aplicatii le vor accesa prin intermediul claselor active. Astfel, proiectantul bazei de date va folosi diagramele de clase pentru structurarea modelului datelor.

Diagramele de clase sunt însotite de catre *diagramele de interactiuni* (diagramele de secvente de exemplu) care vor prezenta modul în care sunt accesate si modificate datele si au implicatii de regula asupra proiectarii aplicatiilor, însa, având în vedere modul cum regulile afacerii au fost repartizate între aplicatii si baza de date (preluarea a o parte din logica de procesare în baza de date), vor avea incidenta si asupra constituirii trigger-ilor, restrictiilor si procedurilor stocate.

Modelul logic constituit din diagramele de clase va cuprinde prin urmare specificatiile claselor împreuna cu atributele, operatiile, relatiile de generalizare, asociere, agregare etc. Este de la sine înteles ca *modelul logic dezvoltat folosind constructorii metodologiei UML* este *un model orientat obiect*. Din acest motiv *modelarea fizica* presupune transformarea modelului logic orientat obiect într-un model logic ce poate fi implementat într-o baza de date. Tot modelarea fizica se va ocupa de construirea componentelor necesare constituirii bazei de date pe suportul unui SGBD existent.

Modelarea fizica

În metodologia UML modelarea fizica presupune doua etape:

- *Maparea modelului logic orientat obiect într-un model logic caracteristic unei baze de date* – transformarea diagramelor de clase
- *Construirea componentelor fizice prin care se va constitui baza de date* adica scripturi DDL, baze de date (scheme), diagrame de componente, diagrame de instalare

În sprijinul proiectarii bazelor de date metodologia UML ofera un cadrul de lucru – ***UML Profile for Database Design*** – ale carui elemente se concentreaza nu asupra con-

struirii aplicatiilor ci asupra modelarii bazelor de date, reprezentând de fapt o extensie a calificatorilor fundamentali din UML catre domeniul modelarii datelor, fiind stereotipizati specific în acest sens. Folosind acest cadru de lucru se vor modela *restrictiile, declansatoarele (trigger-ele), schemele, indecsii, procedurile stocate* etc.

Obtinerea unui model al bazei de date este important pentru mai multe categorii de utilizatori: proiectantii bazei de date îl vor utiliza pentru a aplica regulile bazei de date (normalizare, migrarea cheilor etc.), dezvoltatorii de aplicatii îl vor utiliza pentru a întelege cum vor accesa baza de date aplicatiile pe care le construiesc (mapare între modelul aplicatiilor si modelul bazei de date), dezvoltatorii interfetelor pentru utilizatorii finali îl vor utiliza pentru a verifica consistenta dintre câmpurile formularelor si coloanele din baza de date (inclusiv tip de date, scala, precizie). În acest sens se încurajeaza folosirea urmatoarelor elemente:

- *Package-uri* pentru gruparea logica a elementelor ce formeaza modelul datelor;
- *Diagrama bazei de date* care poate include tabele, coloane, chei primare, chei straine, relatii de identificare, relatii fara identificare, view-uri, proceduri stocate sau domenii (stereotipizate corespunzator).

Pentru a construi ***diagrama bazei de date*** va trebui parcurs un proces de mapare a modelului logic obiectual (extragerea din diagramele de clase acele clase ale caror obiecte vor deveni persistente) cu un model logic ce poate fi portat într-o baza de date obisnuita. Modelele bazelor de date actuale suporta în general urmatoarele paradigme: modelul relational, modelul relational-obiectual si modelul obiectual. Maparea modelului obiectual din UML într-un model relational implementabil într-o baza de date presupune implementarea identitatii, domeniilor, claselor, asociatiilor si relatiilor de generalizare.

O importanta parte a problemelor de mapare au fost rezolvate odata cu definirea principiilor de mapare din E/R în relationale în special în ceea ce priveste asociatiile dar chiar si relatiile de generalizare pentru modelul E/R extins (modelul structural al UML deriva din metodologia OMT care are la baza o varianta extinsa a modelului E/R propus de Chen) - [TEOREY99], [SIMSION2001], [OPREA99], [FOTACHE97].

Problema identitatii îsi gaseste o abordare coerenta în [BLAHA&PREMERLANI98] care propune doua abordari privind ***implementarea identitatii***: identitatea bazata pe existenta, identitatea bazata pe valoare. De asemenea în aceeasi lucrare gasim si o abordare în extenso a problemei implementarii domeniilor privind *domeniul (secvential) al unui identificator, domeniile enumerative, domeniile structurate si domeniile multivaloare*. În legatura cu ***implementarea claselor***, în mod normal fiecare clasa se mapeaza într-o tabela separata, în care fiecare atribut este o coloana. Se cer coloane aditionale pentru identificatorul generat (identitatea bazata pe existenta), asociatiile incluse si discriminatorii de generalizare.

Cu alte cuvinte exista metodologii coerente pentru a rezolva probleme esentiale de mapare dintr-un model obiectual cum este cel al UML în tabele relationale. Aceste principii se dovedesc apoi extrem de importante atunci când se ajunge la implementarea modelului aplicatiei (obiectual) si cel al bazei de date (relationale). Construirea unui strat intermediar care sa asigure persistenta transparenta a obiectelor din spatiul aplicatiei în baza de date poate beneficia din plin de aceste principii pentru construirea unui *framework* eficient.

Concluzie

Implicarea metodologiei UML în proiectarea bazelor de date nu constituie un pas artificial ci o integrare naturala cu procesul mai larg al construirii sistemelor informationale pe principii obiectuale.

Întregul ansamblu al artefactelor UML, materializate sub forma diagramelor de clase, interactiuni, activitati, componente etc. contin inclusiv informatiile necesare proiectantului bazei de date pentru constituirea unui model logic coerent orientat obiect al datelor care vor deveni în mod necesar persistente. Maparea obiectual/relational, care constituie subiectul esential al modelarii fizice pentru baze-

le de date, presupune o serie de probleme specifice, dar ale caror solutii principiale pot fi deduse în special din abordarile cu privire la maparea modelelor entitate-relatie. Acest lucru se datoreaza în special faptului ca modelul structural UML se fundamenteaza pe modelul OMT care la rândul sau constituie o varianta extinsa a modelului E/R.

Bibliografie

[BLAHA&PREMERLANI98] Michael Blaha and William Premerlani, *Object-oriented modeling and design for database application*, Prentice-Hall, Inc. 1998

[BOOCH99] Grandy Booch, James Rumbaugh, Ivar Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley Longman, Inc., Computer and Engineering Publishing Group. Massachusetts USA, 1999

[CONNOLLY2002] Thomas M. Connolly, Carolyn E. Begg *Database systems: a practical approach to design, implementation and management*, third edition, Addison-Wesley Pearson Education Lmt., 2002

[FOTACHE97] Marin Fotache *Baze de date relationale. Organizare, interogare si normalizare*, Junimea Iasi 1997

[JACOBSON99] J. Jacobson, Booch, G., Rumbaugh, I. 1999 *UML Development Process* Addison Wesley Longman, Inc., Computer and Engineering Publishing Group. Massachusetts USA

[NAIBURG2001] Eric J. Naiburg, Robert A. Maksimchuck *UML for database design*, Addison-Wesley 2001

[OPREA99] Dumitru Oprea *Analiza si proiectarea sistemelor informationale economice*, Polirom Iasi 1999

[SIMSION2001] Graeme C. Simsion *Data Modeling Essentials 2nd Edition*, Coriolis 2001

[TEOREY99] Toby J. Teorey *Database modeling & design 3rd Edition*, Morgan Kaufmann Publishers, 1999