

## Limbaajul unificat de modelare. Concepte si notatii

Dan Laurentiu JISA  
 EWIR – Est West Informatique Roumanie  
[dan.jisa@ewir.ro](mailto:dan.jisa@ewir.ro)

*The Unified Modelling Language (UML) is a standard modelling language for the specification, visualisation, construction and documentation of software systems. This paper present an analysis of UML, identifying and discussing the base concepts and diagrams used by it. Taking into account that UML has mainly resulted from the unification of G. Booch, I. Jacobson and J. Rumbaugh's methodologies, the paper focused on identifying the method representing the source of each of the main concepts and diagrams of the language.*

**Keywords:** UML, object oriented analysis and design, OMT, OOSE, OOAD.

### 1 Scurt istoric

Primele limbaje de modelare orientate obiect au aparut si s-au cristalizat ca metodologii în perioada cuprinsa între mijlocul anilor '70 si sfârșitul anilor '80, ca urmare a noilor generatii de limbaje de programare orientate obiect si a creșterii complexitatii aplicatiilor. În perioada cuprinsa între 1989 si 1994 numarul metodelor orientate-obiect a crescut de la mai puțin de 10 la peste 30. Treptat însa, pe baza experientei acumulate, noi generatii de metode au început sa apara, cu un numar relativ restrâns de metode importante: metoda lui G. Booch, OOSE (Object-Oriented Software Engineering), a lui I. Jacobson, OMT (Object Modelling Technique), a lui J. Rumbaugh etc.

Procesul de unificare a metodelor a început sa prinda contur la mijlocul anilor '90, când Grady Booch, Ivar Jacobson si James Rumbaugh au început sa introduca în propriile metode concepte utilizate de catre ceilalti. Obiectivele principale ale unificarii metodelor au fost stabilite dupa cum urmeaza:

- sa permita modelarea de sisteme, de la concept si pâna la artefactele executabile, folosind tehnici orientate obiect;
- sa fie abordate aspectele de scala, inerente sistemelor complexe, cu functionalitati critice;
- sa fie creat un limbaj de modelare utilizabil atât de oameni cât si de calculatoare.

UML este un limbaj de modelare si nu o metodologie. Poate fi definit, pe scurt, ca un limbaj de vizualizare, specificare, construire si documentare a modelelor.

### 2. Concepte utilizate de limbajul unificat de modelare

Pentru a înțelege UML, trebuie înțelese trei elemente majore: *blocurile constructive* ale UML, *regulile* care dicteaza modul în care aceste blocuri pot fi combinate, precum si *mecanismele generale*, utilizate de UML [FMC01].

*Blocurile constructive* se împart în trei categorii: elementele, relatiile si diagramele. Pentru fiecare dintre acestea s-a încercat identificarea metodelor de analiza si proiectare din care provin. Au fost luate în considerare cele trei metodologii principale care stau la baza UML : metoda lui G. Booch, OOSE a lui I. Jacobson si OMT a lui J. Rumbaugh.

*Elementele* sunt abstractiuni care reprezinta piesele fundamentale într-un model. La rândul lor se împart în: elemente structurale (tabelul 1), elemente comportamentale (tabelul 2), elemente de adnotare (tabelul 3) si elemente de adnotare (tabelul 4) grupare [FMC01].

*Relatiile* sunt abstractiuni ce leaga elementele între ele. O relatie este o conexiune între elemente. În modelarea orientata obiect, exista patru tipuri de relatii: dependentele, generalizarile, asocierile si realizările [FMC01].

Diagramele sunt prezentari grafice ale unui set de elemente, cel mai adesea exprimate ca un graf de noduri (elementele) si arce (relatiile).

mate ca un graf de noduri (elementele) si arce (relatiile).

**Tabelul 1. Elemente structurale**

Elemente structurale	OMT	OOADA	OOSE
Clasa	****	****	****
Clasa abstracta	****	****	****
Clasa parametrizabila		****	
Clasa utilitara	****	****	
Metaclasa	****		
Clasa radacina			
Clasa frunza			
Clasa a asocierii	****	****	
Interfata			
Atribut	****	****	****
Atribut calificator	****		
Operatie	****	****	****
Operatie abstracta	****	****	
Operatie frunza			
Vizibilitate a unei operatii / atribut		****	
Obiect	****	****	****
Obiect activ		****	
Instanta prototip			
Instanta concreta			
Componenta		****	
Nod		****	

**Tabelul 2. Elemente comportamentale**

Elemente comportamentale	OMT	OOADA	OOSE
Actor	****		****
Actor abstract			****
Caz de utilizare			****
Caz de utilizare abstract			****
Mesaj		****	stimul
Eveniment	****	****	
Stare	****	****	****
Stare compusa		****	
Stare initiala / finala	****	****	
Istoric al starilor imbricate		****	
Tranzitie	****	****	****
Actiune	****	****	
Activitate	****	****	
Conditie de control (al tranzitiei)		****	

**Tabelul 3. Elemente de grupare**

Elemente de grupare	OMT	OOADA	OOSE
Pachet	Modul	Categorie de clase	

**Tabelul 4. Elemente de adnotare**

Elemente de adnotare	OMT	OOADA	OOSE
Nota		****	

**Tabelul 5. Relatii**

Relatii	OMT	OOADA	OOSE
Asociere	****	****	****
Asociere de extensie			****

Relatii	OMT	OOADA	OOSE
Asociere de utilizare			****
Asociere de includere			
Legatura	****	****	
Agregare	****	****	****
Generalizare / specializare	****	****	****
Relatie dependenta			
Relatie de realizare			
Relatie de instantiere		****	
Relatie de utilizare		****	

*Regulile* stabilesc cum trebuie grupate blocurile constructive. Exista reguli detaliate pentru definirea modelelor bine formate. Sunt admise însa si modele mai putin decât bine formate, în sensul ca se pot admite si modele cu elemente ascunse, modele incomplete sau modele inconsistente, cel puțin pentru iteratiile preliminare ale procesului de dezvoltare [FMC01].

*Mecanisme generale* simplifica dezvoltarea modelelor UML. Exista patru mecanisme generale, aplicate uniform în întreg limbajul: specificatii, ornamente, diviziuni generale si mecanisme de extensie [FMC01].

### 3. Diagramele folosite de limbajul de modelare universal

Diagramele UML pot fi împartite în: diagrame pentru modelarea structurii statice si diagrame pentru modelarea comportamentului.

*Diagrame structurale* sunt folosite pentru a vizualiza, specifica, construi si documenta aspectele statice ale sistemului. Acestea sunt organizate în jurul grupelor majore de elemente ce pot fi gasite în modelarea unui sistem. Din aceasta categorie fac parte :

- *Diagrame de clase* - contin o multime de clase, interfete si colaborari, precum si relatiile dintre ele. Sunt cele mai folosite diagrame în modelarea sistemelor orientate obiect, si ilustreaza vederea statica de proiectare a unui sistem.
- *Diagrame de obiecte* - contin o multime de obiecte si relatiile dintre ele. Sunt folosite pentru a ilustra structurile de date, imagini statice ale instantelor elementelor din diagramele de clase.
- *Diagrame de componente* - contin o multime de componente si relatiile dintre

ele. Se folosesc pentru a ilustra vederea statica de implementare a unui sistem.

- *Diagrame de amplasare* - contin o multime de noduri si relatiile dintre ele, ilustrând vederea statica de amplasare a unei arhitecturi.

*Diagrame comportamentale* sunt folosite pentru a vizualiza, specifica, construi si documenta aspectele dinamice ale unui sistem. Acestea sunt organizate în jurul modalitatilor principale de a modela dinamica unui sistem. Din aceasta categorie fac parte:

- *Diagrame ale cazurilor de utilizare* - arata o multime de cazuri de utilizare, de actori (un tip special de clase) si relatiile dintre ele.
- *Diagrame de secventa*. Diagrama de secventa este o diagrama de interactiune care pune în evidenta ordinea temporală a mesajelor. O diagrama de secventa arata o multime de obiecte împreuna cu mesajele expediate si receptionate de catre acestea.
- *Diagrame de colaborare*. Diagrama de colaborare este o diagrama de interactiune care pune în evidenta organizarea structurala a obiectelor care trimit si receptioneaza mesaje. O diagrama de colaborare arata o multime de obiecte, legaturile dintre aceste obiecte, împreuna cu mesajele expediate si receptionate de catre acestea.
- *Diagrame de tranzitie a starii* - contin masini cu stari, ce constau în stari, tranzitii, evenimente si activitati. Aceste diagrame se folosesc pentru a ilustra vederea dinamica a unui sistem.
- *Diagrame de activitate* - arata fluxul dintr-o activitate în alta în cadrul unui sistem. O activitate arata un set de activitati, fluxul secvential sau ramificat de la o acti-

vitatie la alta, si obiectele care actioneaza sau sunt actionate de consecinta.

Tabelul 6 prezinta metodele de analiza si proiectare din care provine fiecare tip de diagrama UML.

**Tabelul 6.** Diagrame UML

Diagrame UML	OMT	OOADA	OOSE
Diagrame de clase	****	****	****
Diagrame de obiecte	****		
Diagrame ale cazurilor de utilizare			****
Diagrame de secventa		****	****
Diagrame de colaborare		****	
Diagrame de activitati			****
Diagrame de tranzitie a starilor	****	****	
Diagrame de componente		****	
Diagrame de desfasurare		diagrame de proces	

### Bibliografie

[FMC01], Fabian C., Mihalca R., Chichernea V., Goron S., Botezatu C., Iacob I., "Proiectarea sistemelor informatice. Metode de realizare", Sylvi, Bucuresti 2001  
 [BRJ99], Booch G., Raumbaugh J., Jacobson I., "The Unified Modelling Language. User Guide", Addison/Wesley 1999  
 [BOO94], Booch G., "Object/Oriented Analysis And Design – With Applications Second Edition", Redwood City, CA.: The Benjamin/Cummings Publishing Company, Inc., 1994

[JAC96], Jacobson I., "Object-Oriented Software Engineering", Edison-Wesley, 1996

[RBP91], Rumbaugh J., Blaha M., Permelani W., Eddi F., "Object-Oriented Modeling and Design", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1991