

2010 - 2011



Inginerie Software pentru Comunicatii (ISC / RST)

Titular curs: **Eduard-Cristian Popovici**

Suport curs: <http://discipline.elcom.pub.ro/isc/>

Moodle: <http://electronica.curs.ncit.pub.ro/>

Continut curs

1. Introducere in ingineria software

- 1.1. Necesitatea unei abordari sistematice a dezvoltarii software
- 1.2. Abordari si metodologii larg utilizate in ingineria software

2. Introducere in limbajul UML

- 2.1. Definirea, rolul si istoricul limbajului de modelare unificat (UML)
- 2.2. Tipuri de diagrame UML. Organizarea ierarhica a diagramelor

3. Diagrame UML statice

- 3.1. Diagrame UML de clase
- 3.2. Diagrame UML de obiecte
- 3.3. Diagrame UML de pachete
- 3.4. Diagrame UML de componente
- 3.5. Diagrame UML de structuri compozite

Continut curs

4. Diagrame UML dinamice

- 4.1. Diagramele UML de caz de utilizare
- 4.2. Diagrame UML de comunicare si de robustete
- 4.3. Diagrame UML de secventa si de sumar al interactiunilor
- 4.4. Diagrame UML de masini de stari
- 4.5. Diagrame UML de activitati
- 4.6. Diagrame UML de timp

5. Introducere in procesul de dezvoltare Rational unificat (RUP)

- 5.1. Organizarea iterativa a proiectelor
- 5.2. Fazele si activitatile procesului RUP

6. Introducere in managementul si organizarea proceselor de dezvoltare

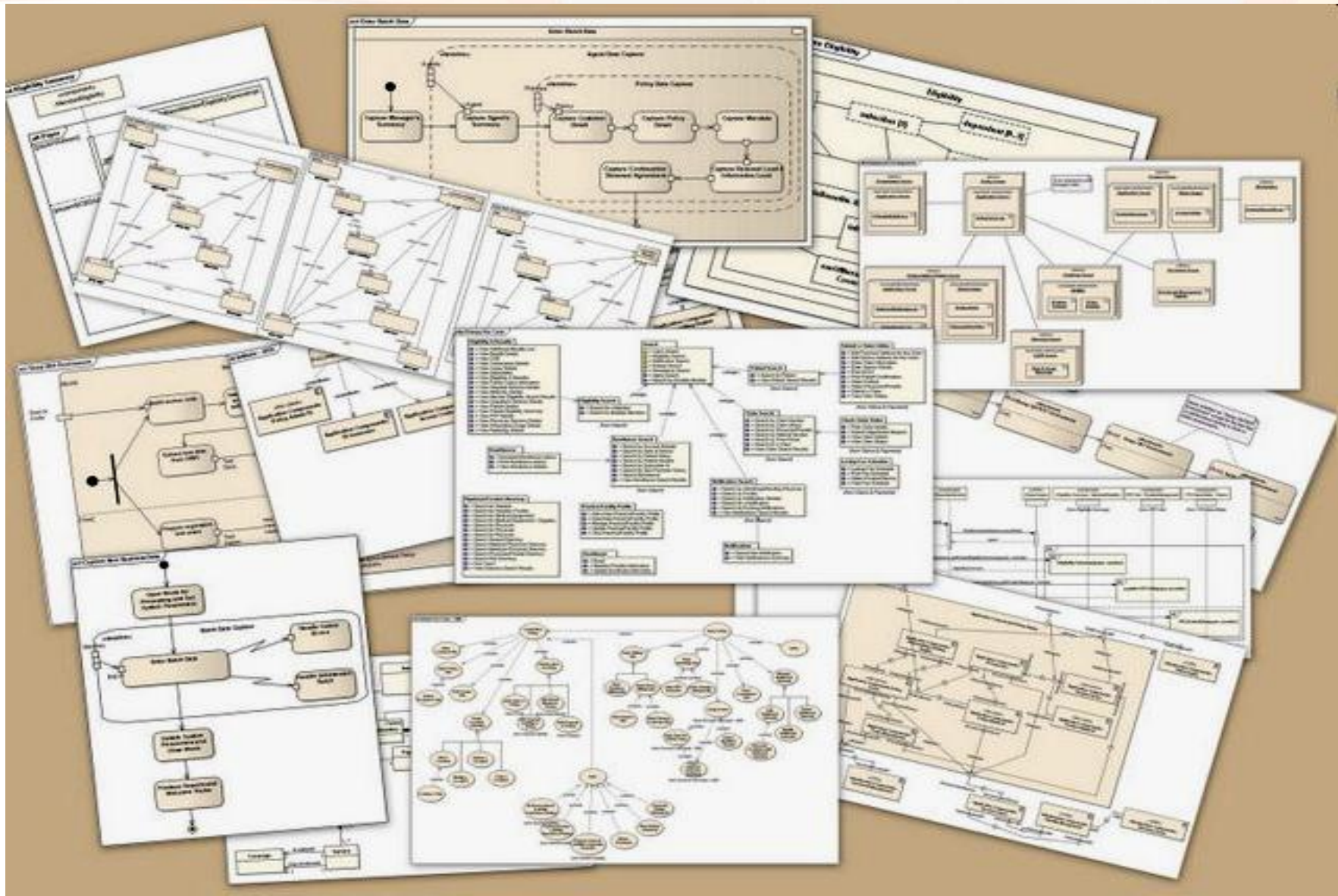
7. Elemente de reutilizabilitate a software-ului. Pattern-uri de proiectare

2. Introducere in limbajul UML

- 2.1. Definirea, rolul, istoricul limbajului de modelare unificat (UML)
- 2.2. Tipuri de diagrame UML. Organizarea ierarhica a diagramelor

Definirea, rolul si istoricul limbajului de modelare (vizuala) unificat (UML)

2. Introducere in limbajul UML



Limbajul de modelare unificat (UML)

UML (*Unified Modeling Language*):

- limbaj de **modelare** standardizat **de uz general** in ingineria software
- produsul fuzionarii notatiilor
 - **Metodei Booch** (creata de Grady Booch),
 - **OMT** (*Object Modeling Techniques*),
 - **OOSE** (*Object Oriented Software Engineering*), si a altora
- **standard OMG** (*Object Management Group*)
- include un **set de de notatii grafice**
 - pentru crearea unor **modele vizuale** ale sistemelor software
 - prin **descrierea artefactelor dezvoltarii** acestora
- **independent** de procesul de dezvoltare folosit

Limbajul de modelare unificat (UML)

UML (*Unified Modeling Language*):

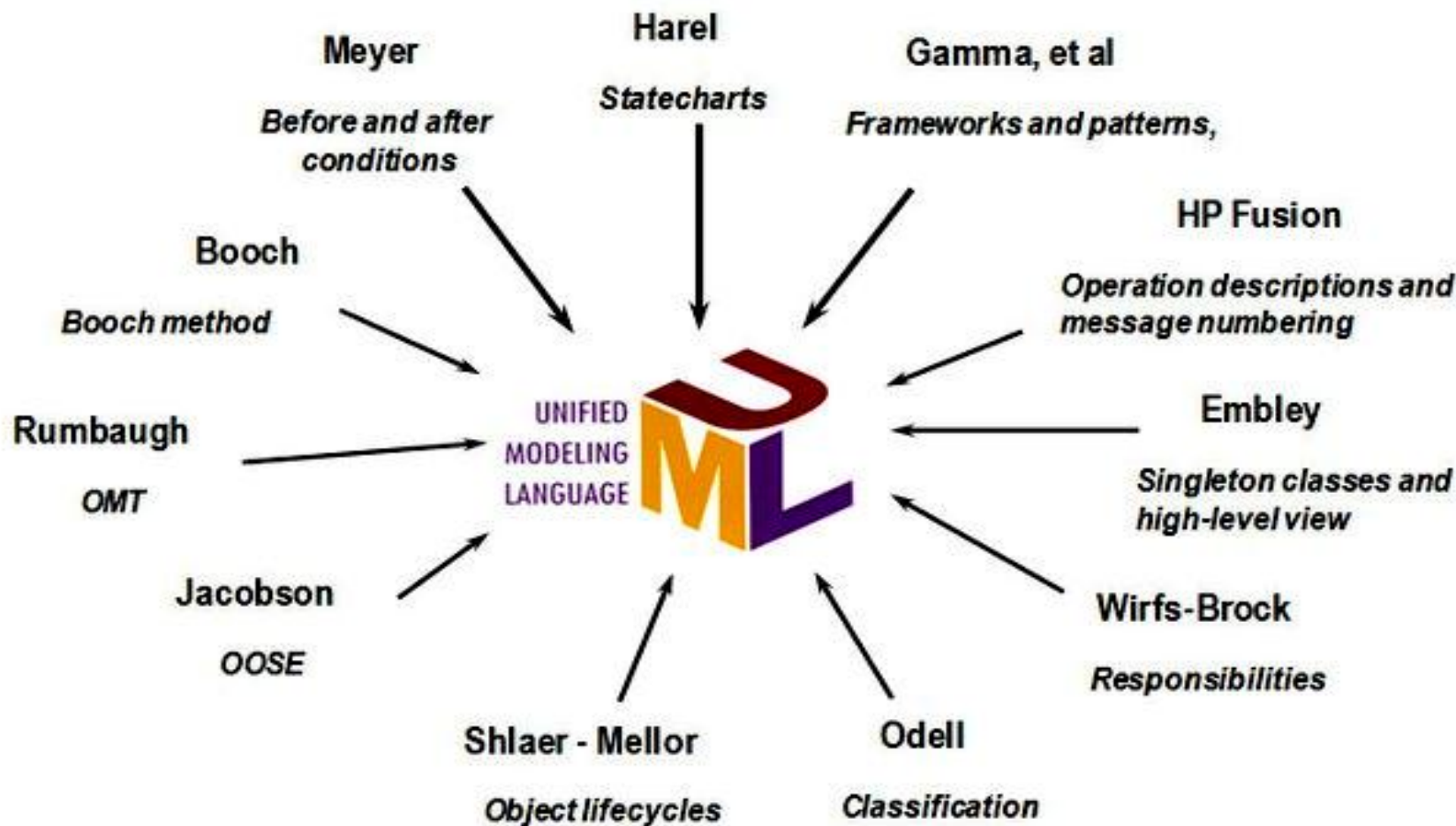
- a fost gândit pentru a fi **intuitiv**, **omogen**, **coerent**
- **nu are o notatie ferma**, el fiind
 - **generic**,
 - **extensibil**,
 - **configurabil**
- **majoritatea diagramelor se prezinta**
 - sub forma unor **grafuri**, compuse din
 - **elemente** (noduri) si
 - **arce** (legaturi intre noduri)

2. Introducere in limbajul UML



Originea limbajului de modelare unificat (UML)

UML este produsul fuzionarii notatiilor **Booch**, **OMT** (Object Modeling Techniques), **OOSE** (Object Oriented Software Engineering), si a altora



Originea limbajului de modelare unificat (UML)

Notatiile **Booch**, **OMT**, si **OOSE** au fost create de **Grady Booch**, **James Rumbaugh** si respectiv **Ivar Jacobson** (*the three amigos*), pe cand erau colegi la **Rational Software** (acum divizie software a **IBM**)



Grady Booch

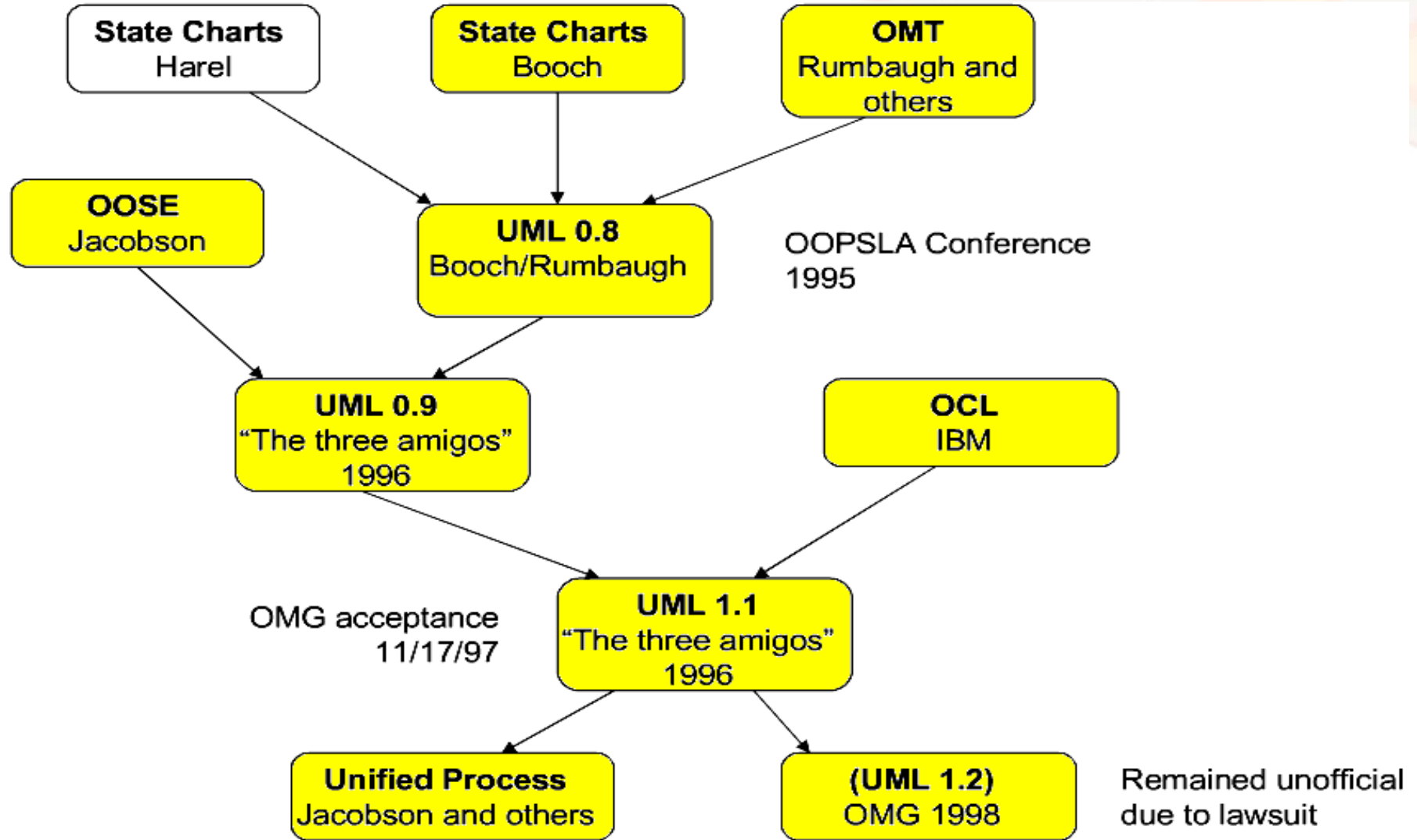


James Rumbaugh

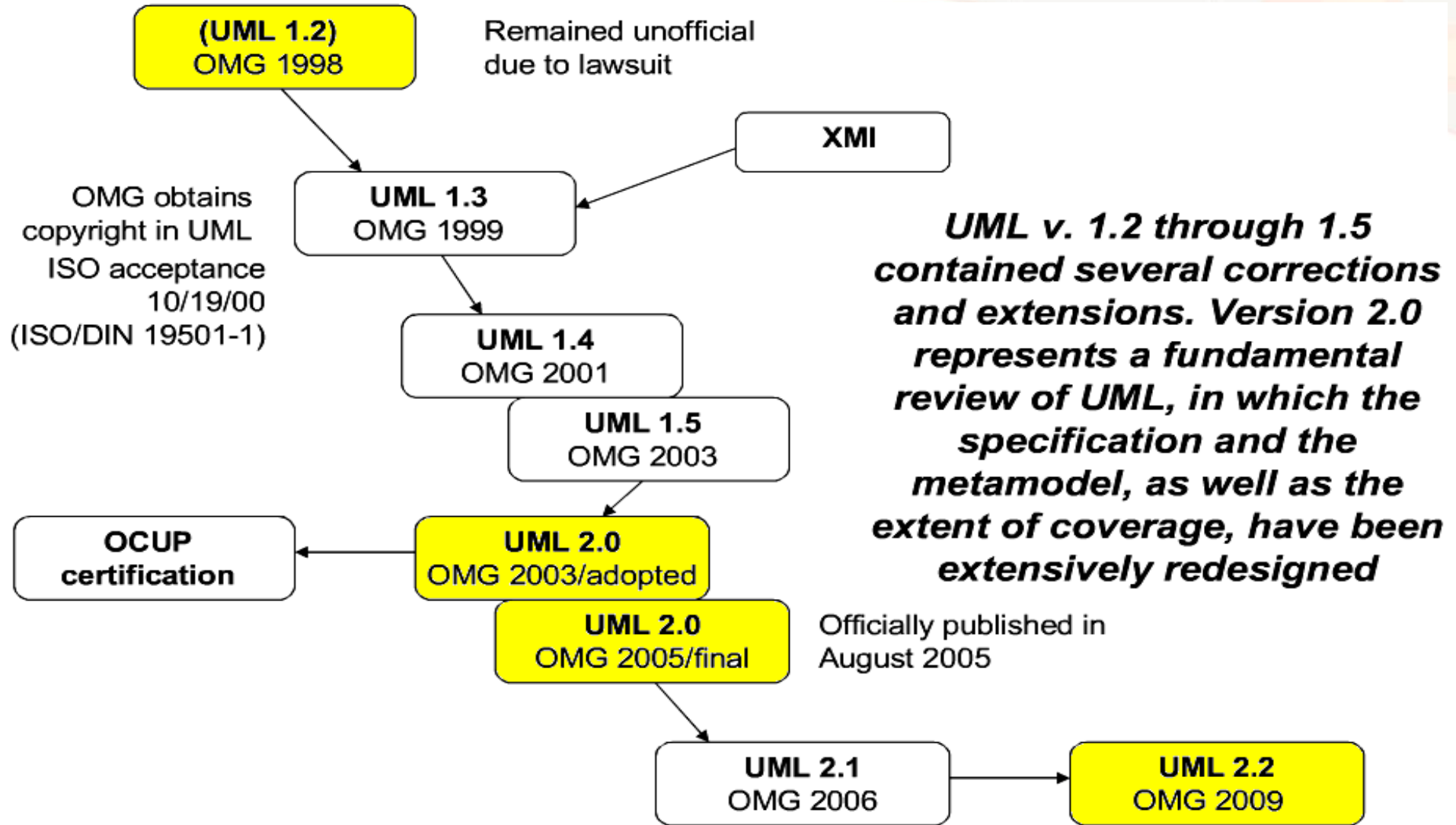


Ivar Jacobson

Istoricul limbajului de modelare unificat (UML)



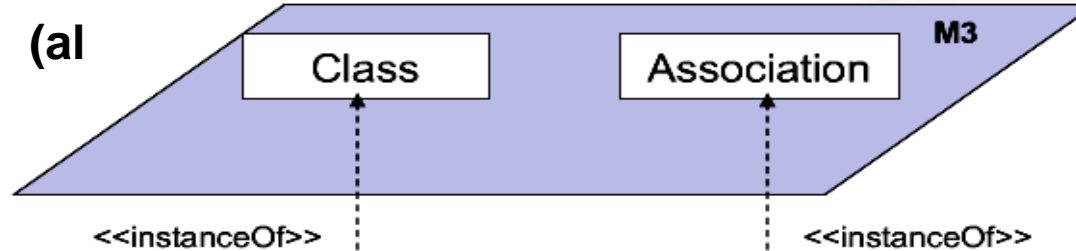
Istoricul limbajului de modelare unificat (UML)



Tipuri de diagrame UML. Organizarea ierarhica a diagramelor

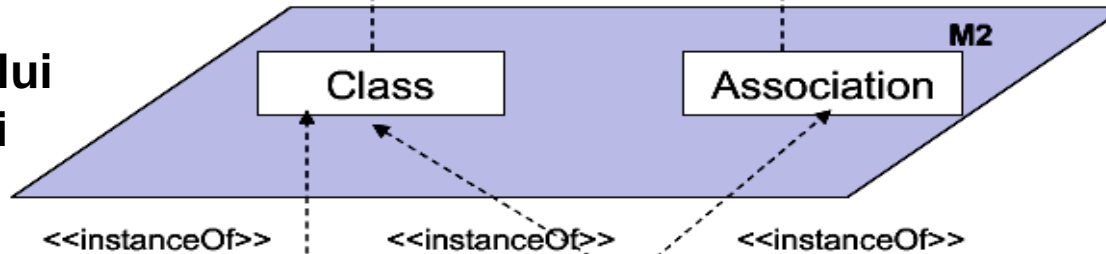
Arhitectura in 4 niveluri a limbajului de modelare unificat (UML)

Nivelul MOF (al conceptelor)



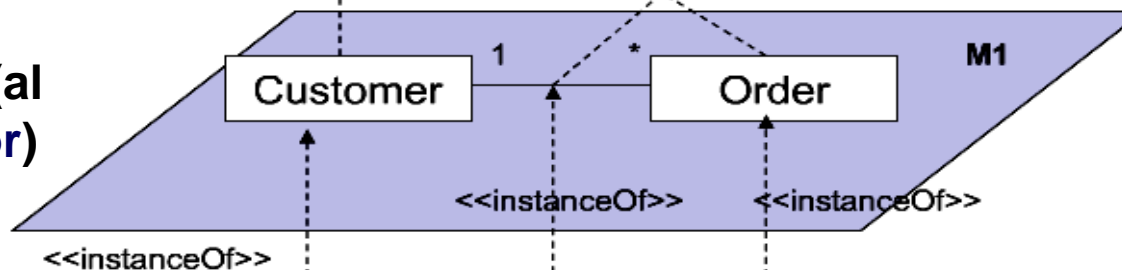
Meta-metamodel
Meta-Object Facility (MOF)
used to model, for example,
the UML metamodel

Nivelul metamodelului (al limbajului UML)



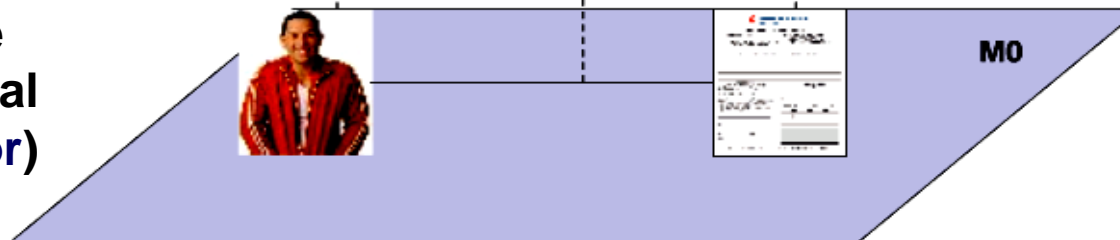
UML metamodel,
classes for all
available UML
elements

Nivelul modelului (al diagramelor)



UML Model, user
modeling level

Nivelul de executie (al instantelor)



Runtime Model

2. Introducere in limbajul UML



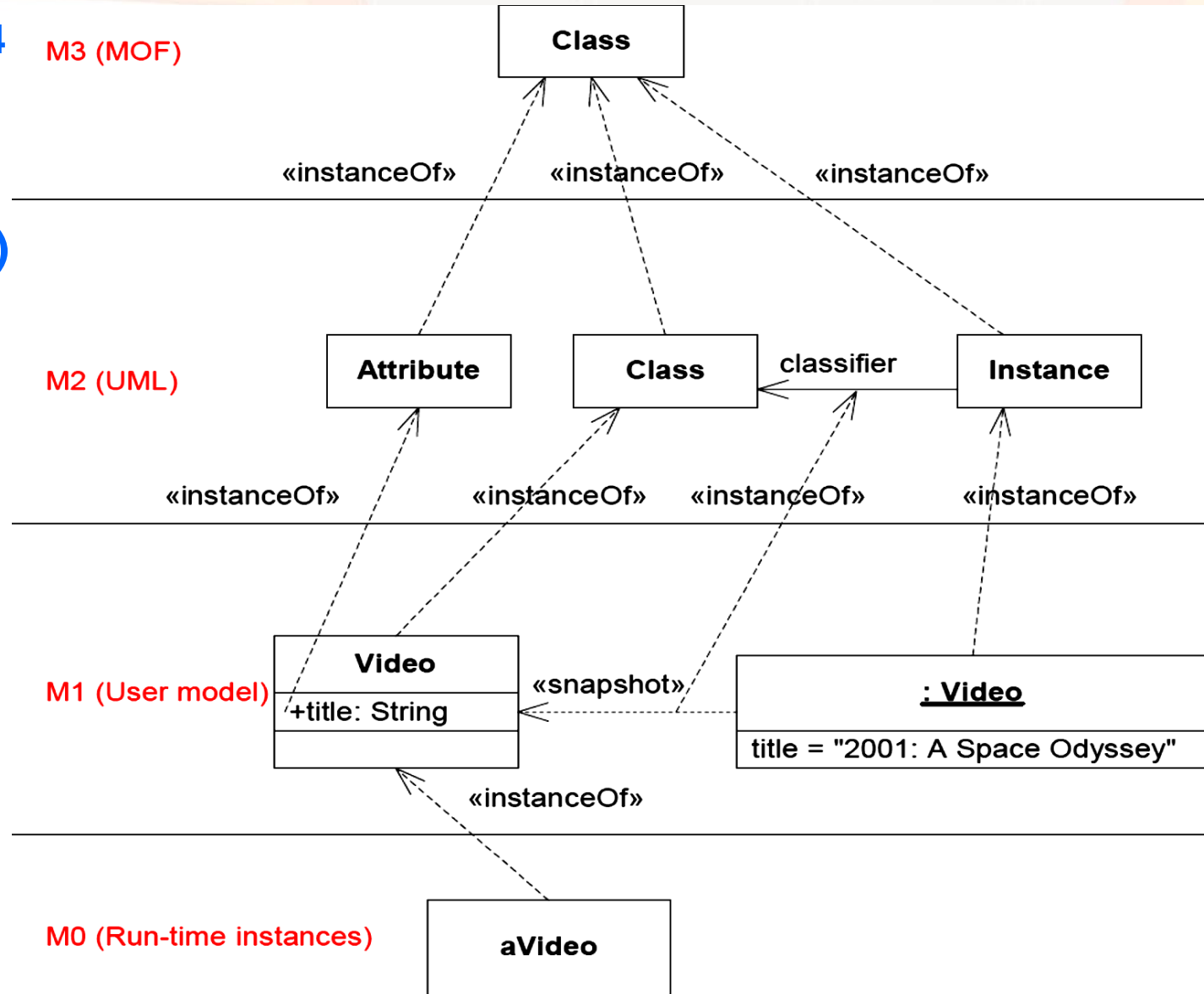
Arhitectura in 4 niveluri a limbajului de modelare unificat (UML)

M3 (MOF)

M2 (UML)

M1 (User model)

M0 (Run-time instances)





Categorii de diagrame UML 2

(si exemplu
de diagrama
de pachete)

UML Diagrams

Structure Diagrams

Class Diagrams

Package Diagrams

Object Diagrams

Composite Structure Diagrams

Component Diagrams

Deployment Diagrams

Behavior Diagrams

Activity Diagrams

State Machine Diagrams

Use Case Diagrams

Interaction Suite

Sequence Diagrams

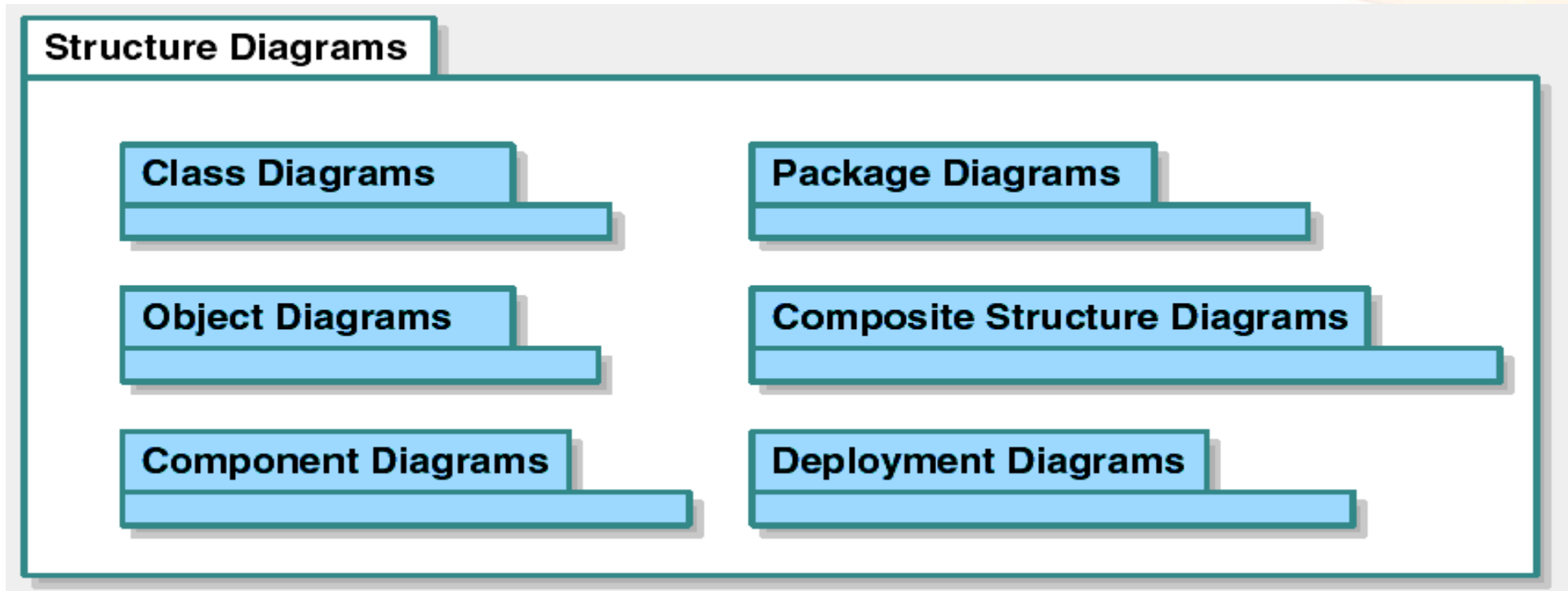
Interaction Overview Diagrams

Communication Diagrams

Timing Diagrams



Diagramele UML structurale



Diagramele UML structurale

Diagramele UML 2 structurale si rolurile lor

➤ diagramele de **clase**

- prezinta **structura statica** în termeni de

- **clase** si

- **relatii între clase** de obiecte (asocieri, agregari, compuneri, extinderi prin mostenire, implementari de interfete)

➤ diagramele de **obiecte**

- prezinta **obiectele si legaturile lor**

- fiind niste **diagrame de comunicatie simplificate**, fara reprezentarea mesajelor trimise între obiecte

➤ diagramele de **pachete (packages)**

- arata felul in care clasele sau componentele sunt **grupate**

Diagramele UML structurale

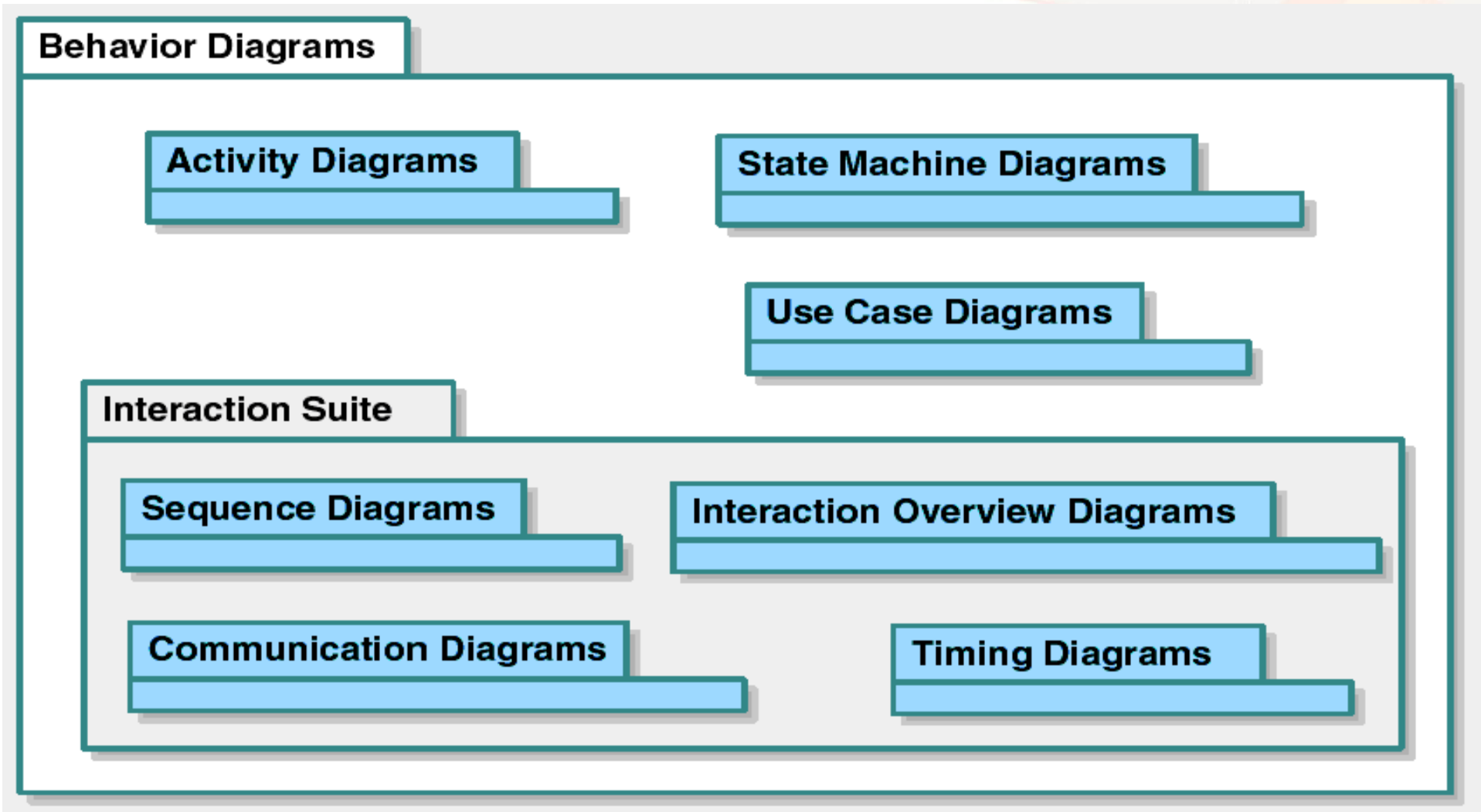
Diagramele UML 2 structurale si rolurile lor

- diagramele de **componente**
 - prezinta **componentele reutilizabile si relatiile dintre ele in cadrul sistemului software**
 - insistand pe **interfetele oferite si necesare**

- diagramele de **structuri composite** (*composite structure*)
 - prezinta felul in care structurile sunt **agregate**

- diagramele de **amplasare** (*deployment*)
 - prezinta
 - **componentele**
 - **manifestarile componentelor sub forma artefactelor si**
 - **amplasarea artefactelor pe suportul hardware**

Diagramele UML dinamice



Diagramele UML dinamice

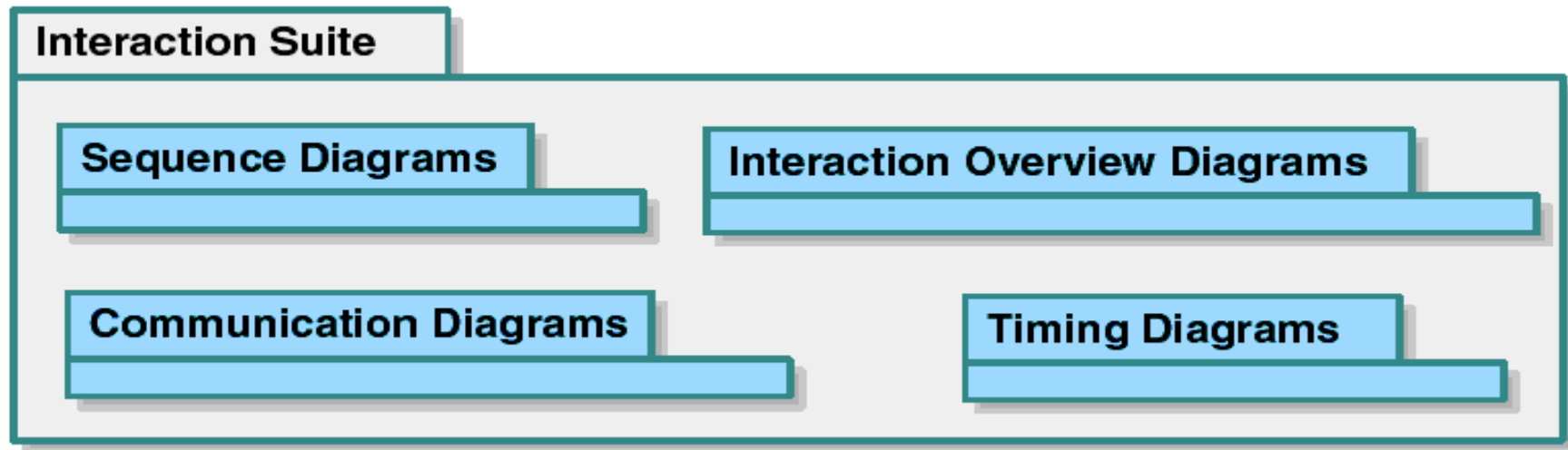
Diagramele UML 2 dinamice si rolurile lor

- diagramele **cazurilor de utilizare** (*use cases*)
 - prezinta **functiile sistemului** din punct de vedere al utilizatorului, adica
 - **modurile in care sistemul este utilizat**

- diagramele de **activitati**
 - asemanatoare **organigramelor** (*Flow Charts*)
 - reprezinta **comportamentul unei operatii în termeni de actiuni**
 - sub forma fluxurilor de activitati si de obiecte

- diagramele **masini de stari** (foste de *stari-tranzitii*)
 - numite si **diagrame FSM** (*Finite State Machines*), **automate**, **State Charts**, etc.
 - prezinta **comportamentul unei clase în termeni de stari si de tranzitii între stari**

Diagramele UML de interactiune



Diagramele UML de interactiune

Diagramele UML 2 de interactiune si rolurile lor

- diagramele de **secventa**, numite si **diagrame MSC** (*Message Sequence Charts*)
 - prezinta temporal interactiunile intre obiecte
- diagramele de **comunicatie** (foste de *colaborare* in UML 1)
 - alternative ale diagramelor de secventa
 - reprezentari spatiale ale obiectelor, legaturilor si interactiunilor
- diagramele de **timing** (sincronizare)
 - reprezinta **constrangerile temporale** ale mesajelor si starilor obiectelor (un fel de **diagrame MSC** rotite cu 90°)
- diagramele de **ansamblu al interactiunilor** (*Interaction overview*)
 - arata **pasii individuali ai unei activitati** folosind diagrame de secventa/comunicatie

Limbajul de modelare unificat (UML)

Diagramele UML formeaza modele folosite in procesul de productie a sistemelor software bazate *Model Driven Architecture (MDA)*

➤ exemplu de proportii ale codurilor generate automat din diagrame UML pentru subsistemele unei aplicatii create folosind MDA

