Curs 3 – Limbaje pentru modelarea informaţională

Cuprins

- ✓ Limbaje pentru modelarea informaţională -UML
- ✓ Elemente de bază ale UML
- ✓ Modelarea cu ajutorul diagramelor UML

- ✓ Instrumente CASE: concepte, facilităţi şi obiective
- Arhitectura mediului CASE
- Instrumente CASE pentru analiză şi proiectare orientată obiect

Modelarea în realizarea sistemelor informatice

- Modelarea este una dintre cele mai importante tehnici de concepere a sistemelor informatice.
- Un ***model*** reprezintă o abstractizare a unei părți a realității înconjurătoare în scopul unei mai bune înțelegeri a produsului ce urmează a fi dezvoltat înainte de a proceda la construcția propriu-zisă a acestuia.
- Odată costruit, modelul servește pentru comunicarea atât în interiorul echipei de dezvoltare a acestuia, cât și în exterior, cu utilizatorii.
- Trăsăturile caracteristice ale modelării sunt: *simplificarea, subordonarea la un scop, reprezentarea unei realități, divizarea, ierarhizarea și comunicarea.*

Modelarea în realizarea sistemelor informatice

- Analiza şi proiectarea orientată-obiect foloseşte trei tipuri diferite de modele pentru descrierea unui sistem informatic:
  - ***modelul static*** care descrie obiectele si relaţiile lor în sistem;
  - ***modelul dinamic*** ce descrie interacţiunile între obiecte în cadrul sistemului ;
  - ***modelul funcţional*** care descrie transformarea valorii datelor în sistem.

Limbaje pentru modelarea informaţională

- La modul general, limbajele pentru modelarea informaţională fac posibilă descrierea concisă şi exactă a proprietăţilor sistemelor la diferite nivele de abstractizare.
- Sunt o parte integrantă a procesului de dezvoltare a sistemelor informatice în cadrul căruia pot fi folosite pentru:
  - a face legătura între etapa de analiză a cerinţelor şi cea de implementare;
  - a verifica proprietăţile critice ale unor sisteme;
  - a ajuta la generarea automată de cod şi de cazuri de test.

Categorii de limbaje pentru modelarea informaţională

În funcţie de nivelul de formalizare pe care îl impun, limbajele pentru modelarea informaţională se împart în:

- Limbaje informale: limbajul natural
- Limbaje semi-formale: UML, BPMN sau SysML
- Limbaje formale: OCL (Object Constraint Language), Z

Reguli de sintaxă	NU	DA	DA
Reguli de semantică	NU	NU	DA
Limbaj de specificare a cerințelor	Informal (limbajul natural)	Semi-formal (UML)	Formal (Z,OCL)

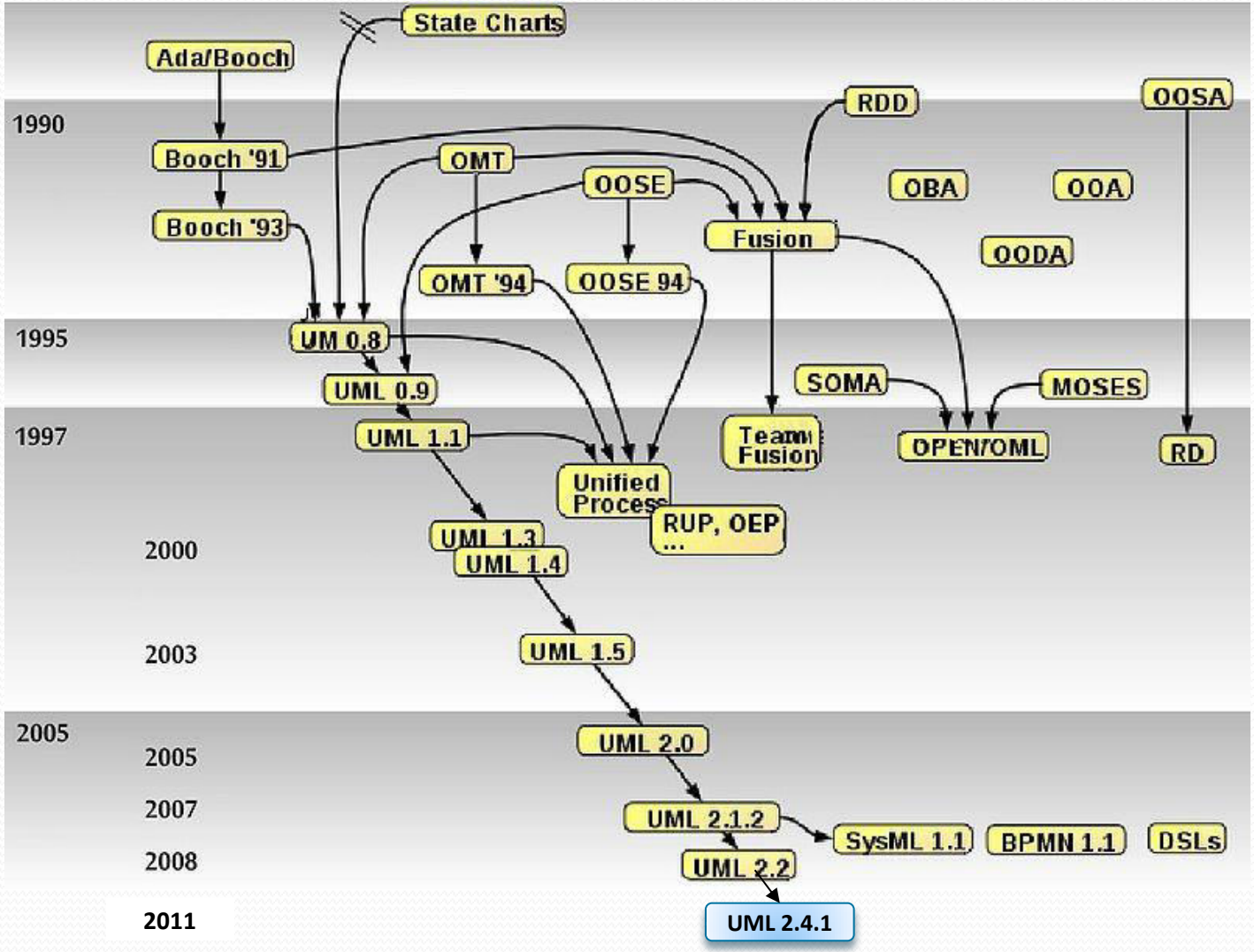
Limbajul UML

- Limbajul standardizat ***UML*** (***Unified Modeling Language***) a apărut din necesitatea unei standardizări a tipologiei, semanticii și a reprezentării rezultatelor.
- În prezent, ***UML*** este un standard de modelare recunoscut de catre OMG (Object Management Group), standardizarea fiind realizată în noiembrie 1997, iar până în prezent realizându-se o perfecționare continuă a acestuia.
- UML poate fi definit ca fiind un limbaj de vizualizare, specificare, construire si documentare a modelelor. Valoarea lui constă în faptul că este un standard deschis, realizează întreg ciclul de dezvoltare al software-ului, acoperă multe tipuri de aplicații, este bazat pe marea experiență a celor care l-au realizat și poate fi implementat de multe produse de tip CASE.

Limbajul UML

- UML are notații standard și semantică adecvată modelării sistemelor orientate obiect.
- Odată cu apariția acestui limbaj, proiectanții de sisteme pot înțelege mai ușor documentațiile de sistem.
- Înaintea acestei standardizări, un proiect orientat-obiect putea fi descris folosind una din multele metode orientate-obiect disponibile, iar dacă era necesară o revizuire, se pierdea un timp important cu analiza notațiilor și semanticii metodei folosite, înainte de a începe logica proiectării.

Istoria UML

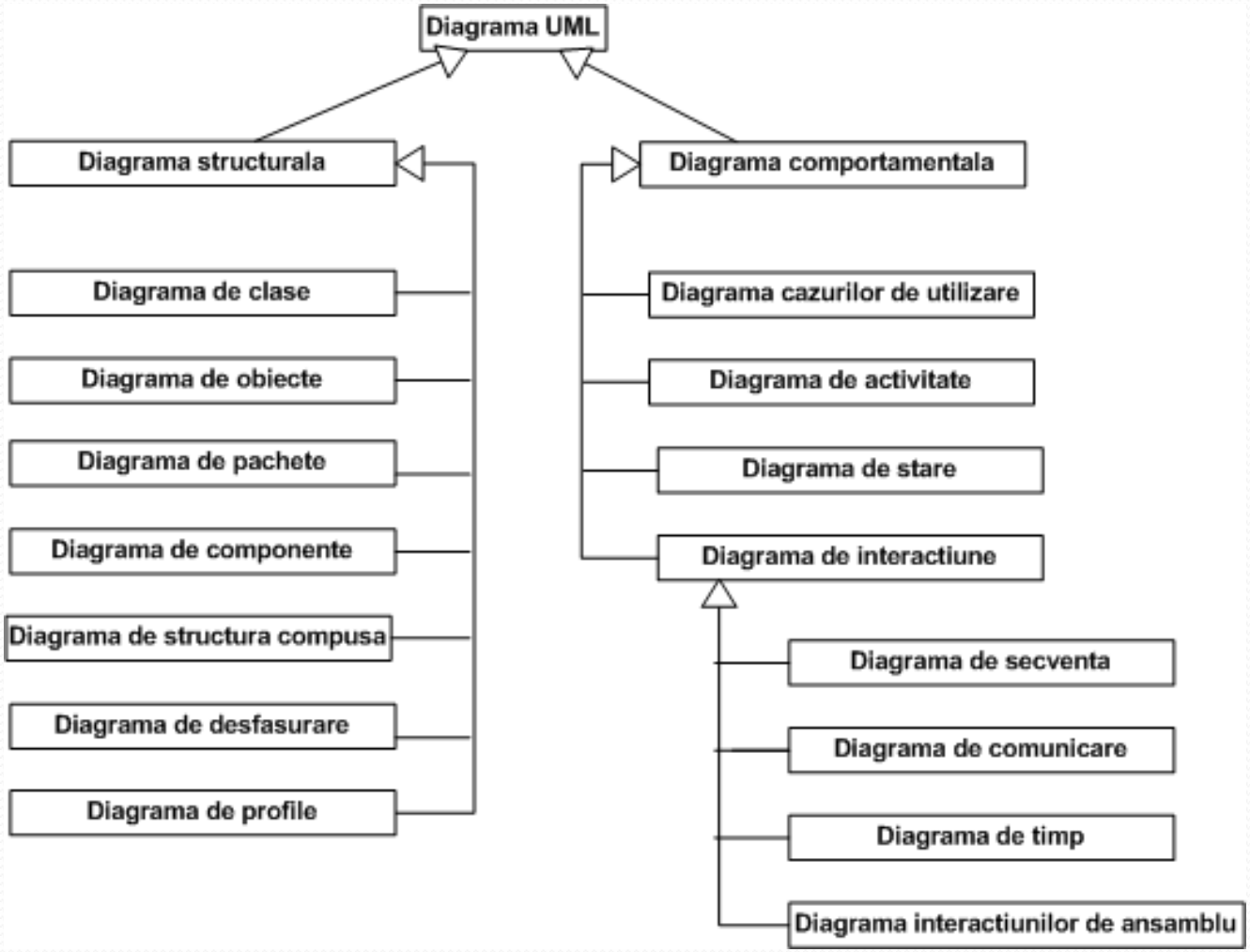

State Charts
Ada/Booch
1990
Booch '91
Booch '93
OMT
OOSE
RDD
OOSA
OBA
OOA
Fusion
OODA
OMT '94
OOSE 94
1995
UM 0.8
UML 0.9
SOMA
MOSES
1997
UML 1.1
Team Fusion
OPEN/OML
RD
Unified Process
RUP, OEP ...
2000
UML 1.3
UML 1.4
2003
UML 1.5
2005
UML 2.0
2005
2007
UML 2.1.2
SysML 1.1
BPMN 1.1
DSLs
2008
UML 2.2
2011
UML 2.4.1

Elementele de bază ale UML

1. Metamodel pentru modelarea orientată obiect
   - Set coerent de definiţii ale unor **concepte** şi a **relaţiilor** dintre ele;
   - Se defineşte, folosind **o sintaxă precisă**, fiecare element utilizat în modelare (exemplu: definirea unei clase);
   - Limbaj suport pentru transmiterea modelelor vizuale între diferite instrumente;

Elementele de bază ale UML

2. Tipuri de diagrame



Elementele de bază ale UML

3. Mecanisme de extensie

- ***Stereotipurile*** caracterizează un element din model sau o relație între elemente (există sterotipuri predefinite).
- ***Comentariile (notele)*** descriu suplimentar un element din model.
- ***Contrângerile*** limitează utilizarea unui element din model.
- ***Valori etichetate*** reprezintă atribute definite pentru un stereotip.
- ***Profilele*** personalizează metamodelul prin construcţii care sunt specifice unui anumit *domeniu*, *platformă* sau *metodă de dezvoltare*.

Modelarea cu ajutorul diagramelor UML

- *Modelarea proceselor de afaceri* se realizează folosind **diagrama cazurilor de utilizare**. Această diagramă dirijează întreg procesul de dezvoltare a sistemului în cazul metodelor orientate pe cazuri de utilizare.

- *Modelarea structurii statice* se face cu ajutorul **diagramei claselor** (pentru modelarea structurii statice a claselor sistemului) și **diagramei obiectelor** (pentru modelarea structurii statice a obiectelor sistemului). **Diagrama pachetelor** este un mijloc de grupare a elementelor diagramelor în pachete.

Modelarea cu ajutorul diagramelor UML

- *Modelarea dinamicii* este realizată utilizând diagrame de interacţiune şi diagrame care descriu comportamentul. UML utilizează două diagrame de interacţiune, una pentru modelarea circuitului mesajelor între obiecte, numită **diagrama de secvenţă** şi alta, pentru modelarea interacţiunilor între obiecte, denumită **diagrama de comunicare.** Ca diagrame ce descriu comportamentul, se utilizează **diagrama de stare** pentru modelarea comportamentului obiectelor din sistem şi respectiv **diagrama de activitate** pentru modelarea comportamentului cazurilor de utilizare, obiectelor sau a operaţiilor.

- *Modelarea implementării* se face cu ajutorul a două diagrame. Modelarea componentelor se realizează utilizând **diagrama componentelor**, iar modelarea distribuirii sistemului se obţine folosind **diagrama de desfăşurare**.

Computer Aided Software Engineering

- **Instrumentele CASE** sunt pachete software care se bazează pe o anumită metodologie/metodă și oferă suport proiectantului în realizarea unui produs informatic.
- **Utilitatea** instrumentelor CASE este de ordin **cantitativ** (reducerea timpului și a costului de realizare) și **calitativ** (aplicarea riguroasă a unei metodologii de realizare, reducerea deficiențelor de implementare, obținerea unei documentații standard, ușurința la reproiectare).
- Instrumentele CASE reduc substanțial sau elimină multe din problemele de proiectare și dezvoltare a aplicațiilor informatice.

Analiză între dezvoltarea tradiţională şi dezvoltarea bazată pe CASE a SI

Dezvoltarea tradiţională a sistemelor informatice	Dezvoltarea sistemelor bazată pe CASE-uri
Pune accentul pe codificare şi testare	Pune accentul pe analiză şi proiectare
Specificaţiile sunt bazate pe hârtie	Prototipizare interactivă rapidă
Codificarea manuală a programelor	Generarea automată a codului
Generarea manuală a documentaţiei	Generarea automată a documentaţiei
Testarea software-ului în mod continuu	Validarea automată
Întreţinerea codului şi a documentaţiei	Întreţinerea specificaţiilor de proiectare

Obiective

- specificarea corectă și completă a **cerințelor** sistemului;
- reducerea **timpului** și **costului** de proiectare și dezvoltare;
- obținerea unor **specificații de proiectare** exacte, actuale, cu prezentare vizuala;
- integrarea activităților de **proiectare** și **dezvoltare** prin utilizarea unor metodologii/metode comune;
- **standardizarea procesului** de proiectare și dezvoltare a sistemelor informatice;
- simplificarea și îmbunătățirea procesului de **testare**;
- realizarea unor **documentații** de calitate;
- îmbunătățirea **managementului proiectelor**;
- simplificarea etapei de **întreținere** a sistemelor informatice;
- **reutilizarea** modulelor aplicațiilor și a documentației;
- îmbunătățirea **portabilității** aplicațiilor;
- **flexibilitate**

Facilităţi oferite de CASE

- suport pentru conducerea proiectului (management resurse si versiuni)
- generarea **documentaţiei** de realizare a sistemului informatic;
- generarea automată a **codului de program**, pornind de la specificaţiile de proiectare;
- tehnica de inginerie inversă (**reverse ingineering**) prin care se permite revenirea la o etapă precedentă pentru modificări .
- suport pentru una sau mai multe **metode de analiză şi proiectare** a sistemelor informatice. Principalul suport oferit îl constituie **editoarele de diagrame şi text**;
- stocarea şi regăsirea datelor din **depozitul central** de date (**repository**) prin utilitare specifice;
- verificarea automată a **consistenţei** şi **completitudinii** datelor printr-un **analizor** ce conţine reguli specifice pentru fiecare metodologie/metodă;
- suport pentru realizarea de **prototipuri**, prin limbaje de programare de nivel înalt şi generatoare;

Tipologia instrumentelor CASE

- *După* ***aria de cuprindere a ciclului de realizare a aplicației***:
  - ***Instrumente CASE front-end*** (sau ***upper CASE***) care oferă suport pentru primele etape de realizare a aplicațiilor informatice (analiza și specificarea cerințelor, proiectarea logică).
  - *Instrumente* ***CASE back-end*** (sau ***lower CASE***) care oferă suport pentru ultimele etape de realizare a aplicațiilor informatice (proiectarea fizică, elaborarea programelor, testarea, întreținerea sistemului).
  - *Instrumente* ***CASE cross life cycle*** care oferă suport pentru activitățile ce apar în mai multe etape ale procesului de proiectare și dezvoltare a sistemelor informatice (de exemplu, instrumente utilizate pentru managementul proiectului, generatoare de documentație).

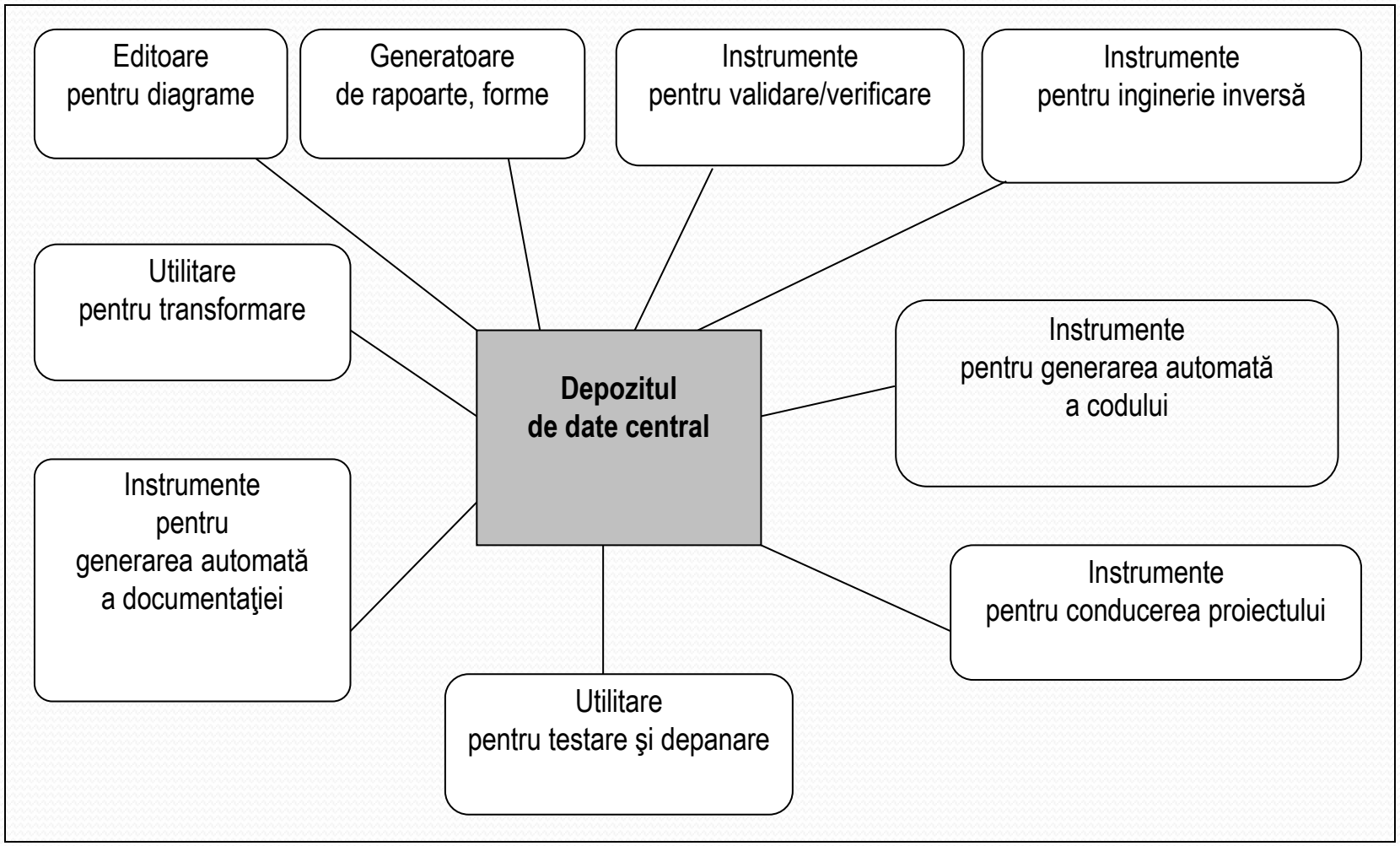
Tipologia instrumentelor CASE

- ***După dimensiunea suportului oferit***:
  - ***Instrumente CASE propriu-zise*** ce oferă suport pentru o singură activitate din cadrul unei etape de realizare a aplicaţiilor (de exemplu, editoare de diagrame şi text, instrumente pentru analiza consistenţei şi completitudinii specificaţiilor de sistem, depanatoare etc.);
  - ***Bancurile de lucru CASE*** **(workbenches)** care oferă suport pentru o etapă din ciclul de realizare a aplicaţiei;
  - ***Mediile CASE*** care oferă suport pentru cea mai mare parte (sau toate) dintre etapele de realizare a sistemului informatic. Din această categorie fac parte următoarele instrumente: Oracle Designer/Oracle, IBM Rational Architect (Rational Rose)/IBM, Cradle/3SL, Corporate Modeler/CASEWise Inc etc.

Tipologia instrumentelor CASE

- ***După scopul utilizării***:
  - Instrumente CASE **pentru analiză** care utilizează în general metode de analiză structurată și orientată obiect.
  - Instrumente CASE **pentru proiectare** (instrumentele pentru generarea automată a codului, generatoare de interfețe, instrumente pentru inginerie inversă, instrumente pentru modelarea datelor).
  - Instrumente CASE **pentru testare și depanare** (generatoare de date pentru testare, instrumente de depanare interactive).
  - **Generatoare de interfață** (generatoare de forme, generatoare de rapoarte, generatoare de meniuri).
  - **Generatoare de documentație** (editoare de imagini, instrumente pentru formatarea paginii (layout)) etc

Arhitectura mediului CASE



Arhitectura mediului CASE

- ***Depozitul de date central*** **(nucleul unui mediu I-CASE)** stochează toate obiectele și informațiile necesare pentru proiectarea, modelarea și generarea aplicațiilor. *Contine:*
  - *Depozitul de informații* conține informațiile despre afacerile organizației și portofoliul ei de aplicații.
  - *Dicționarul de date* gestionează și controlează accesul la depozitul de informații. El stochează descrieri ale datelor și ale resurselor de prelucrare a datelor.
- ***Editoarele pentru diagrame*** permit reprezentarea vizuală a unui sistem și a componentelor lui. Diagramele sunt foarte eficace pentru reprezentarea fluxurilor de procese, a structurilor de date și a structurilor de program.
- ***Utilitarele pentru transformare*** convertesc elementele obținute cu instrumentele de analiză în elemente ale proiectării.

Arhitectura mediului CASE

- ***Generatoarele de forme şi rapoarte*** sunt utilizate pentru a crea, modifica şi testa prototipurile de forme şi rapoarte şi pentru a identifica datele care vor fi afişate sau colectate pentru fiecare formă şi raport
- ***Instrumentele CASE pentru validare/verificare*** generează rapoarte prin care se identifică inconsistenţele, redundanţa şi lipsurile din diagrame, forme şi rapoarte.
- ***Instrumente pentru generarea automată a codului*** pornind de la specificaţiile de proiectare conţinute în depozitul de date central (instrumente pentru generarea obiectelor bazei de date şi a modulelor aplicaţiei).

Instrumente CASE pentru analiză şi proiectare orientată obiect

- Instrumentele CASE pentru analiză şi proiectare orientată obiect sunt printre cele mai noi tipuri de instrumente CASE.
- Ele promovează **realizarea iterativă** a sistemelor şi aplicaţiilor informatice, ceea ce permite revenirea la etapele anterioare pentru efectuarea unor completări sau modificări pe măsură ce se conturează arhitectura sistemului informatic.

- Proiectarea orientată obiect încurajează modularitatea, extensibilitatea şi reutilizarea codului. Aceste instrumente conţin:
  - **utilitare** pentru descrierea obiectelor, claselor şi a proprietăţilor lor (diagrame pentru descrierea statică şi dinamică), precum şi lucrul cu acestea;
  - **generatoare** specializate pentru generarea **codului**, generarea **documentaţiei** etc.

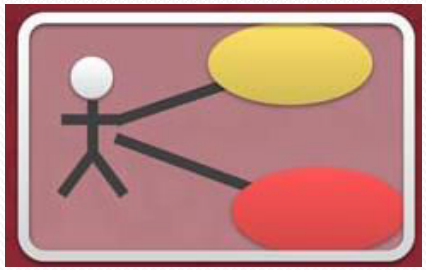
Instrumente CASE pentru analiză şi proiectare orientată obiect

- La ora actuală există multe instrumente CASE, care utilizează metodele de proiectare orientată obiect, implementează limbajul UML, iar limbajele orientate obiect sunt foarte populare (C++, Java pentru aplicaţii Web etc).

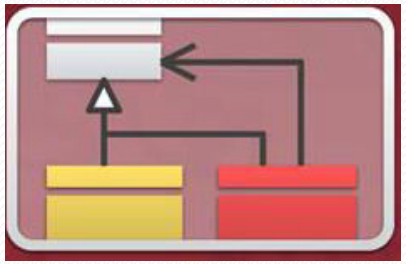
- Exemple de instrumente CASE bazate pe UML:
  - Visual Paradigm for UML
  - Rational Software Modeler
  - MagicDraw
  - Microsoft Visio
  - Poseidon for UML
  - Enterprise Architect
  - BOUML
  - Modelio
  - UModel

Visual Paradigm

- **Visual Paradigm pentru UML (VP-UML) este un instrument CASE de ultimă generație**
- **Se axează pe trei direcții principale:**



Identificarea cerințelor



Construirea de modele

Generarea de cod și baze de date

- **Asigură interoperabilitatea cu alte instrumente CASE (Visio, Visual UML, Rational) și integrarea cu instrumentele IDE de marcă (Net Beans)**
- **Acoperă mare parte a ciclului de viață al unui sistem informatic**

Visual Paradigm

Include modele ale unor limbaje standard

- **Modelarea UML-** Pot fi create toate tipurile de diagrame UML 2.x prin construirea de modele de cazuri de utilizare, comportamentale, de interacțiune, structurale, de amplasare, de cazuri de test.
- **Modelarea BPMN –** Pot fi create: diagrame de procese de afaceri, diagrame de flux de date, diagrame de hărți de procese, diagrama lanțului de procese conduse prin evenimente, organigrame. Diagramele de procese de afaceri **pot fi exportate în BPEL**.
- **Modelarea SysML -** SysML este un limbaj general pentru ingineria aplicațiilor și sistemelor informatice. VP-UML permite crearea **diagramei de cerințe** specifice limbajului SysML.

Visual Paradigm

Modelarea cerințelor

Identifică cerințele prin intermediul a mai multe mecanisme:

- **Diagrame de cerințe SysML** pentru a identifica cerințele funcționale sau non-funcționale ale sistemului.
- **Analiza textuală** oferă editor de text prin intermediul căruia sunt înregistrate cerințele în format textual și care permite identificarea termenilor sau obiectelor importante (clase, cazuri de utilitare) pentru descrierea problemei.
- **Cardurile CRC** conțin informații precum descrirea clasei, atributele acesteia și responsabilitățile. Au formate proprii de afișare a informațiilor.
- **Editorul de interfețe utilizator** prin intermediul căruia se crează machete ale ecranelor.
- **Managementul glosarului de termeni** prin care se identifică și se descrie vocabularul proiectului.

Visual Paradigm

Modelarea bazelor de date

- Se pot crea două tipuri de diagrame pentru modelarea bazelor de date: **diagrame Entitate-Relație** și diagrame **ORM** Object Role Modeling (pentru a vizualiza maparea dintre modelul de obiecte și modelul de date).
- Se pot modela nu numai caracteristici ale **tabelelor**, ci și **proceduri stocate, declanșatori, secvențe și viziuni** ale bazei de date într-o diagramă Entitate-Relație.
- Diagramele se pot construi de la zero sau prin inginerie inversă plecând de la o bază de date existentă.
- **Sincronizare** între diagrama de clase și diagrama Entitate-Relație pentru a asigura consistența între cele două modele.
- **Generarea de cod SQL** pornind de la modele.

Visual Paradigm

Generare de cod

- Generatoarele de inginerie inversă și directă oferă suport pentru ingineria modelelor. Facilitatea **Java Round-Trip engineering** permite sincronizarea continuă a codului și a modelului pentru limbajul Java.

Model	Inginerie directă	Inginerie inversă
Java	x	x
C++	x	x
XML Schema	x	x
PHP	x	x
Python Source	x	x
Objective-C	x	x
CORBA IDL Source	x	x
.NET dll sau fişiere .exe		x
CORBA IDL Source		x
XML (structure)		x
JDBC		x
Hibernate		x
C#	x	
VB.NET	x	
ODL	x	
ActionScript	x	
Delphi	x	
Perl	x	
Ada95	x	
Ruby	x	

Visual Paradigm

Integrarea cu medii IDE

- Oferă suport pentru întreg ciclul de dezvoltare a unui sistem informatic folosind pentru etapa de programare următoarele produse de tip IDE :
  - Eclipse
  - NetBeans/Sun ONE
  - IntelliJ IDEA

Generarea documentației

- **Documentația** poate fi partajată și proiectată împreună cu beneficiarii sistemului folosind unul dintre formatele: **HTML** (report generation) , **HTML** (project publisher) , **PDF , Word**.

Tendinţe în dezvoltarea instrumentelor CASE

- În ultimul deceniu, tehnologiile CASE şi piaţa pentru CASE au devenit mature. Un factor poate stimulează piaţa pentru instrumente CASE este dorinţa organizaţiilor de a extinde durata de viaţă a sistemelor informatice existente, prin utilizarea:
  - i) instrumentelor CASE cu **facilităţi de inginerie inversă** ce permit adaptarea programelor la noile configuraţii hardware. Analistul poate restructura codul pentru a corespunde la cerinţele curente ale afacerii;
  - ii) instrumentelor CASE cu **facilităţi de reinginerie** care permit modificarea sistemului informatic existent, în scopul de a îmbunătăţi calitatea sau performanţa lui.
  - De asemenea, instrumentele CASE orientate obiect şi cele vizuale se vor dezvolta rapid, iar inteligenţa artificială va fi inclusă în mediile de dezvoltare (utilizarea unor agenţi inteligenţi).

Tendinţe în dezvoltarea instrumentelor CASE

- Creşterea nivelului de integrare a instrumentelor CASE depinde în mare măsură de definirea unui **standard pentru depozitul central** de date care să fie recunoscut de majoritatea instrumentelor.
- Cu toate că tehnologia CASE este considerată de unii specialişti ca fiind încă imatură, ea oferă un suport important în realizarea sistemelor informatice.
- Deşi tind să acopere integral ciclul de realizare a sistemelor, instrumentele CASE nu vor reuşi să înlocuiască echipa de realizare a sistemelor, dar vor aduce **schimbări importante în modul de realizare** a sistemelor. Astfel, **primele etape** de realizare a sistemelor (analiza şi proiectarea) vor căpăta o importanţă tot mai mare, în detrimentul etapelor finale de realizare a sistemelor (elaborarea programelor, implementarea şi menţinerea în funcţiune).