

Integrarea sistemelor informatice

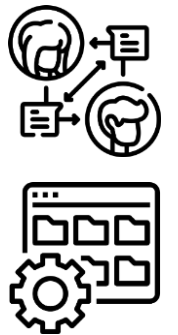


Recapitulare curs teoretic

2024-2025

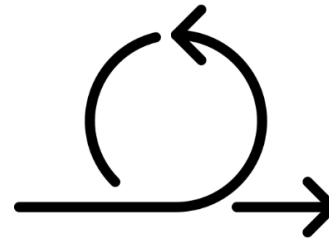
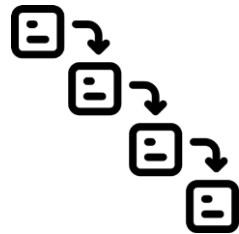
Sisteme informatice

- Un sistem informatic constă din oameni și mașini care produc și/sau folosesc informații care sunt unite prin sisteme de comunicații
- Un sistem informatic este integrat dacă:
 - Procesele de afaceri și procesele informatice care le susțin sunt corelate în profunzime
 - Legătura între diferitele programe este în mare măsură automatizată
 - Datele sunt achiziționate și disponibile pentru toate programele, fiind gestionate în mod centralizat
- Un sistem informatic redă atât procesele productive, interne cât și schimburile din interiorul firmei și dintre firmă și mediul înconjurător



Procesul dezvoltării proiectelor software

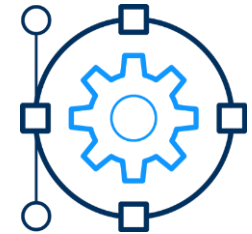
- Waterfall – 1950, evoluat din ingineria “clasică”, preferat de organizații mari, fără implicarea directă a clientului
- Agile – 2001, specific domeniului software, preferat de companii flexibile care dezvoltă produse noi, cu implicarea clientului încurajată la fiecare iterație



Procesul dezvoltării soluțiilor software

Etape și principii de bază (din perspectivă tehnică)

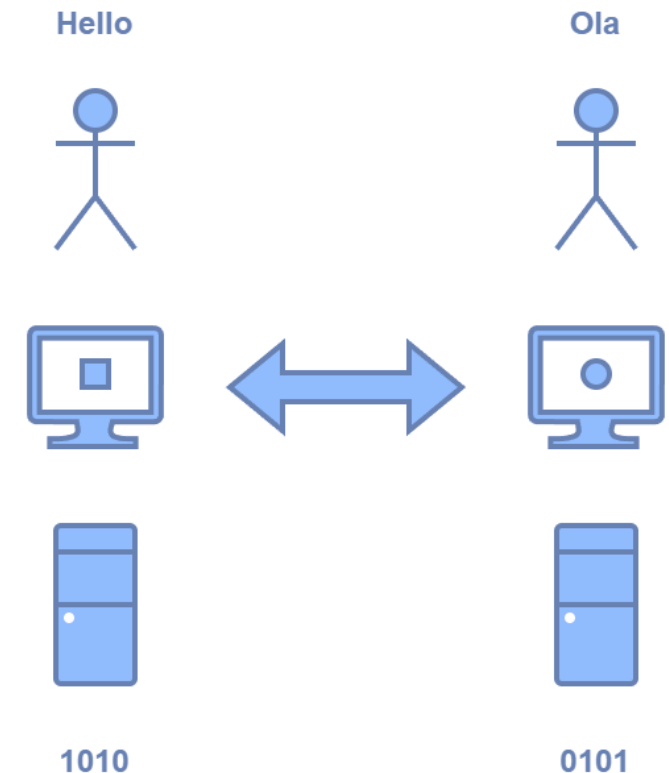
- Înțelegerea problemei și a cerințelor: funcționale (business), non-funcționale (performanță)
- Alegerea unei arhitecturi (structură, organizare, implementare)
- Aplicarea principiilor arhitecturale (design patterns)
- Alegerea unui framework (modelare UML, framework software)
- Documentarea soluției (diagrame, comentarii, prezentări, wiki)
- Actualizarea periodică a soluției (îmbunătățire continuă)



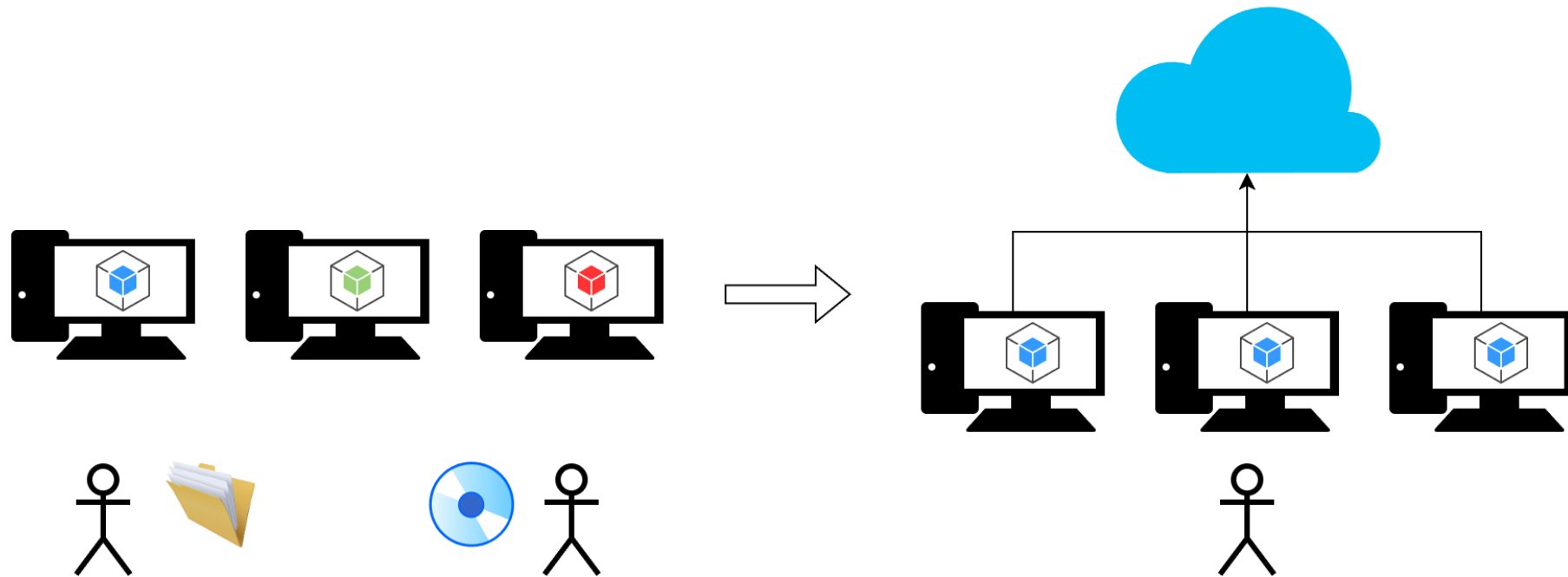
Concepte de bază – Integrare

Integrarea

- Proces de inginerie care creează sau îmbunătățește **fluxul de informații** între sisteme informaționale create pentru diferite scopuri
- Procesul de **interconectare a unui sistem** cu un altul, cu scopul de a asigura un schimb util de informații, date și/sau de control între sisteme.

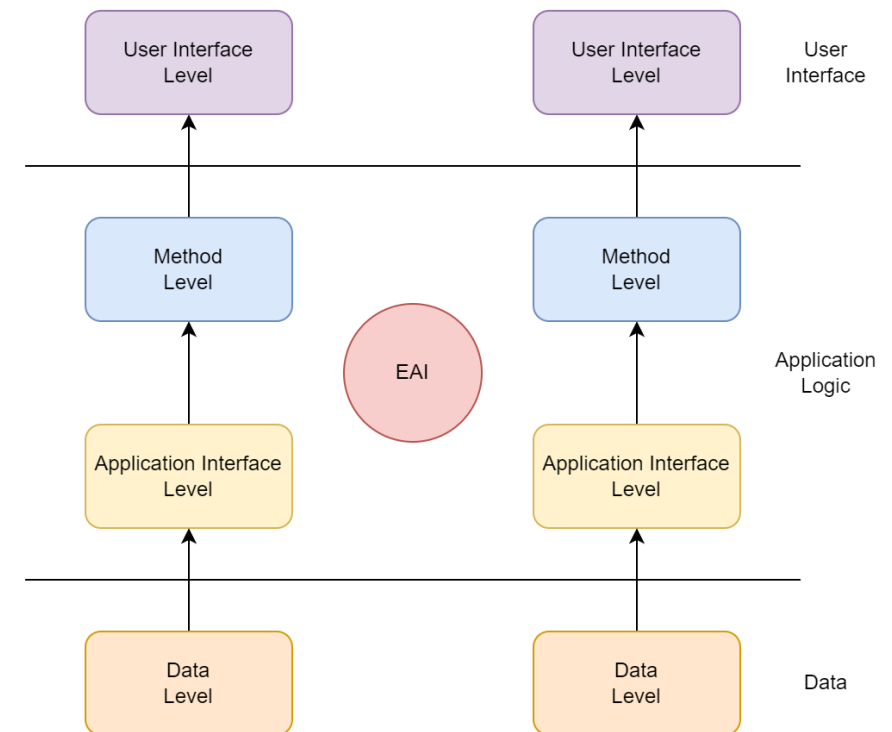


Clasificarea sistemelor informatice din perspectiva integrării



Niveluri de integrare

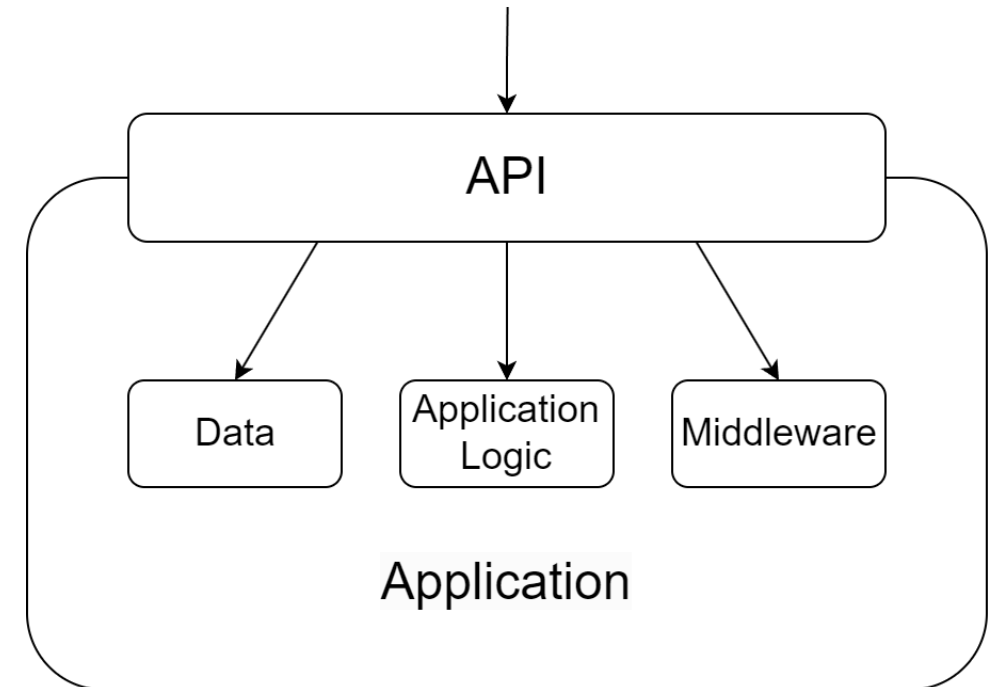
- **EAI** – Enterprise Application Integration
- Niveluri de integrare
 - Nivelul **de date**
 - Nivelul **funcțional** / logică aplicație
 - Nivelul **interfeței utilizator**



Integrarea aplicațiilor – API

Application Programming Interface

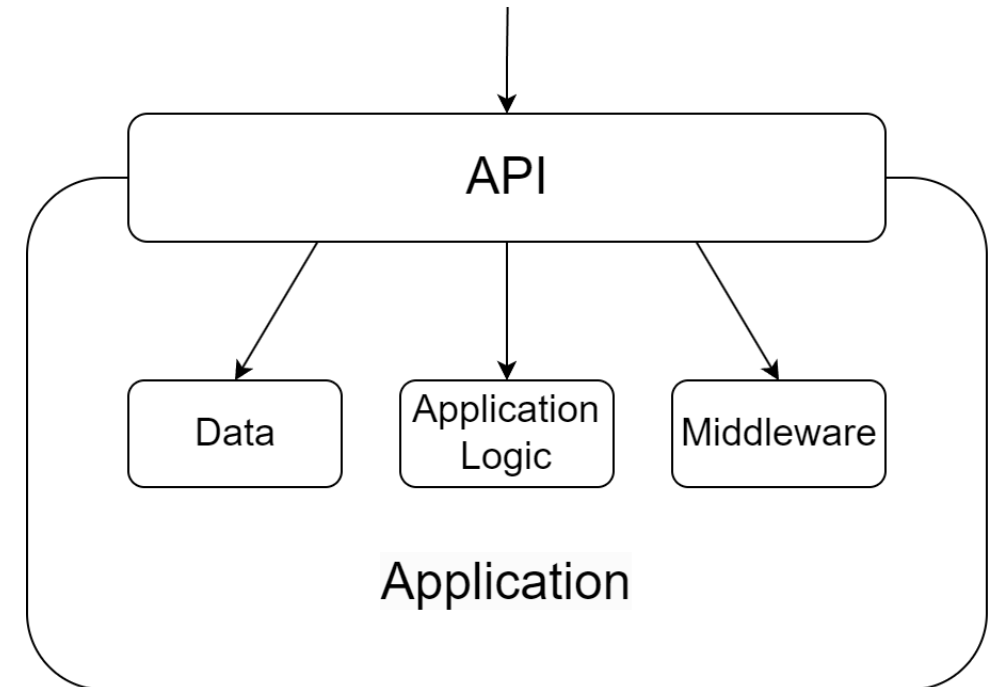
- Un API este un mecanism bine definit pentru conectarea la una sau mai multe componente ale unei aplicații
- Reprezintă principala metodă de integrare la nivel funcțional – logică/procese de afaceri



Arhitectura unui API

Modele arhitecturale de API

- **SOAP** – XML, aplicații enterprise
- **RESTful** – HTTP, servicii web
- **GraphQL** – cereri structurate
- **gRPC** – comunicație microservicii
- **WebSocket** – aplicații de timp real
- **Webhook** – HTTP, notificări /callback



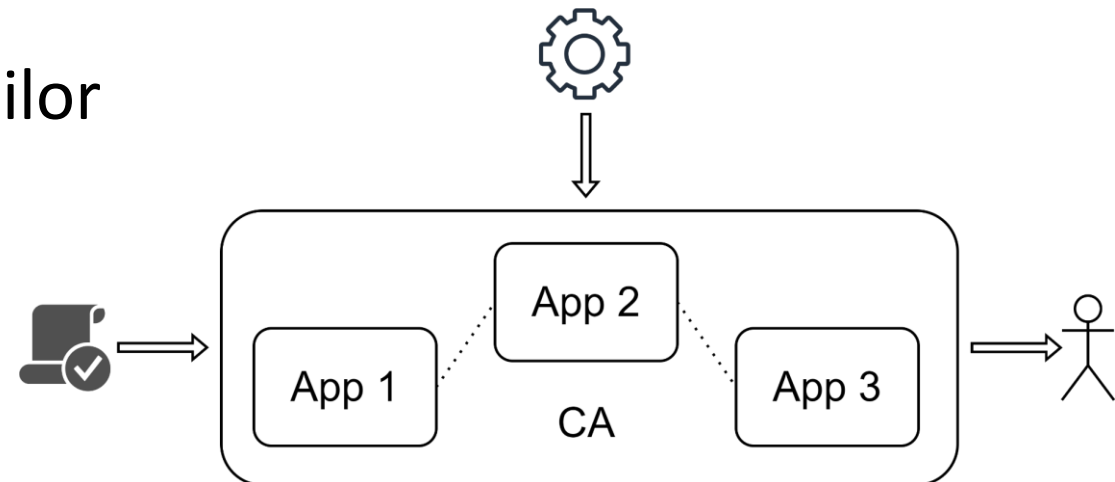
Aplicații compozite

Cum gândim problema?

- **Aplicații compozite / CA:** combinarea aplicațiilor / componentelor software existente într-o aplicație integrată

Cum abordăm proiectarea CA?

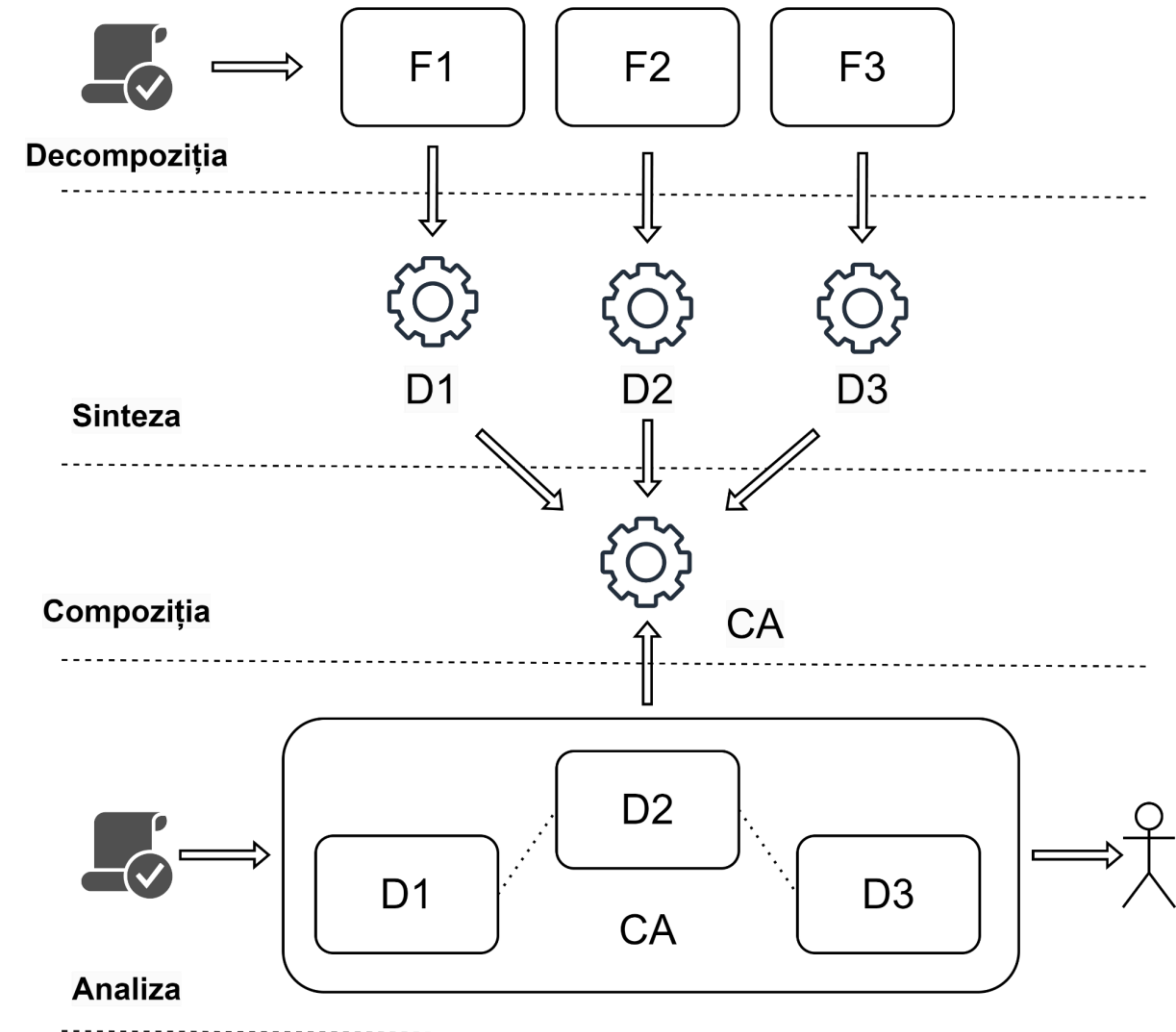
- **Ingineria proiectării:** aplicarea principiilor moderne de dezvoltare de software
- **Niveluri de abstractizare:** proiectarea arhitecturilor software integrate



Proiectarea aplicațiilor compozite

Ingineria proiectării aplicațiilor compozite implică următoarele activități de bază

- **Decompoziția** – func. / comport.
- **Sinteza** – componente / dispozitive
- **Compoziția** – CA
- **Analiza** – testare / evaluare



Modelarea aplicațiilor compozite – Arhitecturi

Tipuri de arhitecturi

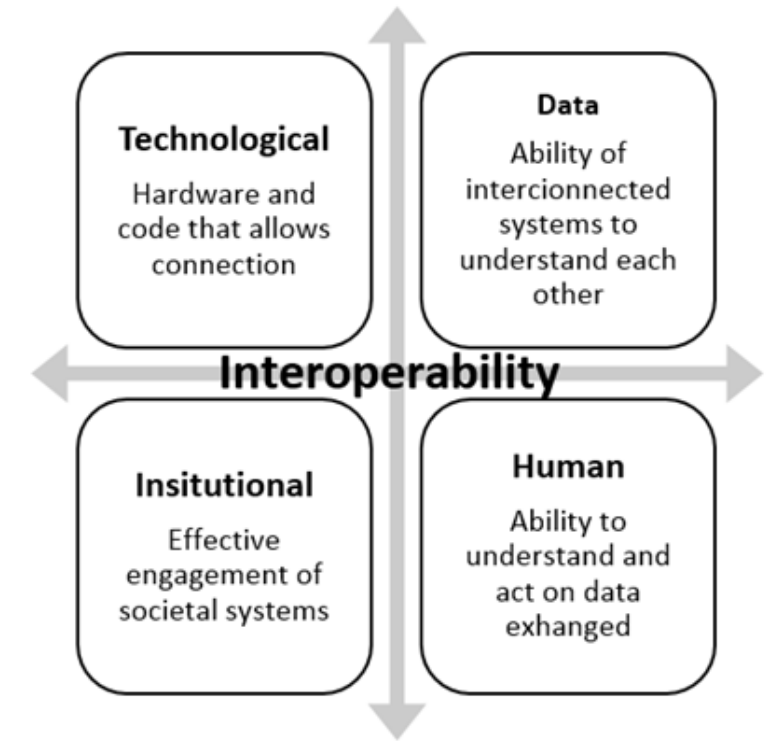
- Arhitectura centralizată
- Arhitectura “two-tier”
- Arhitectura “three-tier”
- SOA (Service-Oriented Architecture)
- Arhitectura bazată pe microservicii
- Arhitecturi serverless



Image by Freepik

Probleme de integrare

- Probleme de natură tehnică
 - Conflicte de conexiune
 - Conflicte sintactice
 - Conflicte de control
 - Conflicte legate de calitatea serviciului
 - Conflicte legate de consistența datelor
 - Conflicte semantice
 - Conflicte funcționale
- Probleme legate de politica sistemului
- Probleme legate de logistică

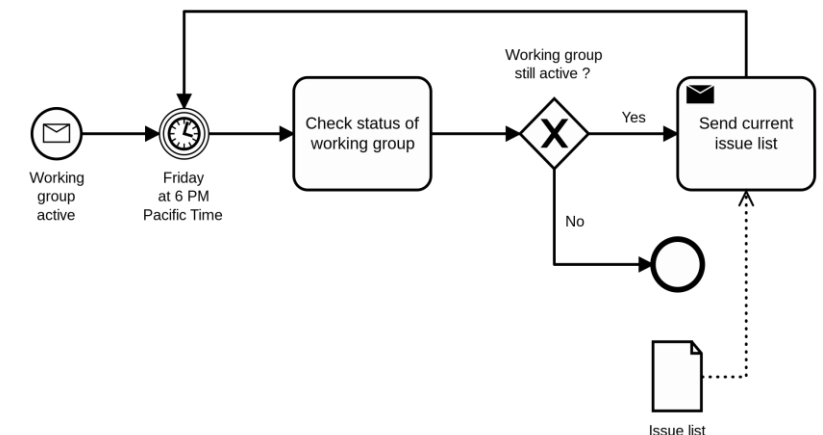
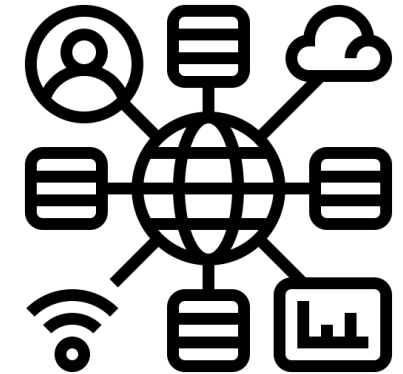


Interoperability | The IT Law Wiki | Fandom



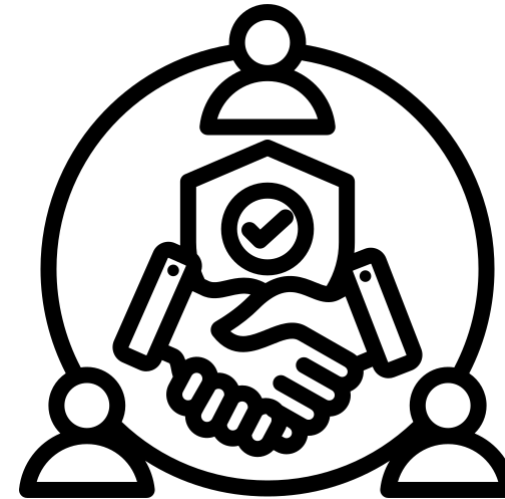
Procesul de integrare

- Înțelegerea **domeniului** organizației și al problemelor
- Înțelegerea **datelor** – identificare, modelare
- Înțelegerea **proceselor** de afaceri – modelare, integrare
- Identificarea **interfețelor** aplicațiilor
- Identificarea **evenimentelor** din sistem
- Identificarea scenariilor de **transformare a datelor**
- Maparea **transferurilor de informație**
- Utilizarea **tehnologiilor**
- Evaluarea **rezultatelor** – testarea, performanța
- Crearea procedurilor de **întreținere**



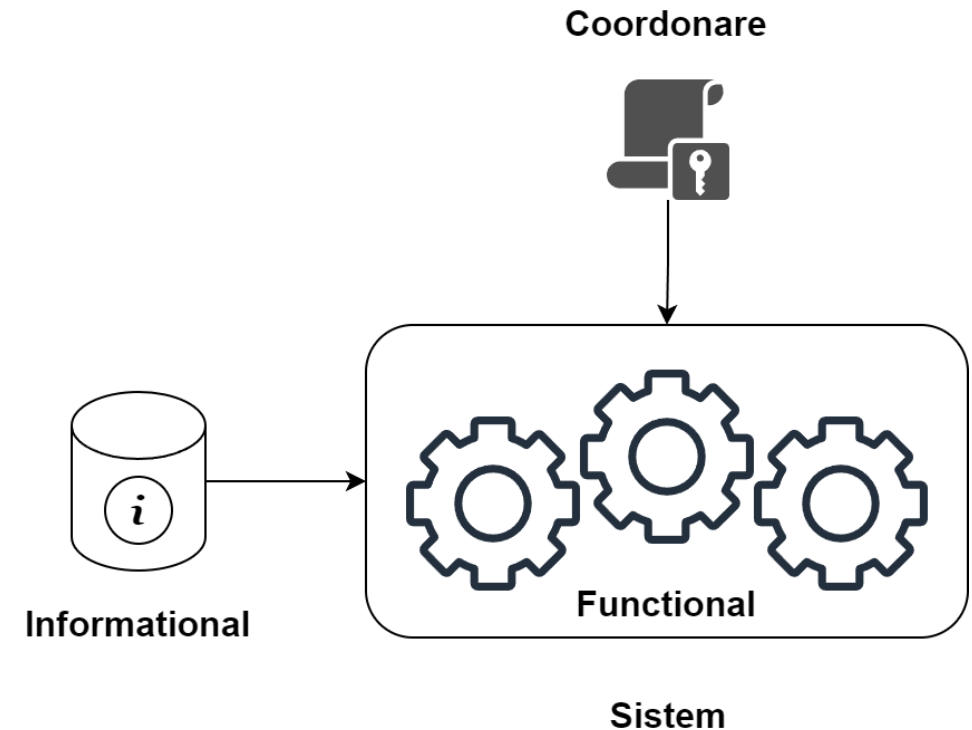
Aspecte de integrare

- Un **aspect de integrare** este un set de teme ce trebuie avute în vedere la integrare, privind compatibilitatea componentelor cu rolul lor în sistem, precum și cu actorii cu care interacționează într-un set dat de interacțiuni
 - Aspecte tehnice
 - Aspecte semantice
 - Aspecte funcționale
 - Aspecte de politică
 - Aspecte logistice



Modelarea domeniului

- **Domeniul** – reunește conținuturi, concepte și idei necesare pentru descrierea arhitecturii unui sistem software particular
- **domain model** – “a product of domain analysis that provides a representation of the **requirements** of the domain.” (ISO/IEC/IEEE 24765)



Domenii de abstractizare

Pentru a separa nivelurile de abstractizare, modelele se diferențiază prin perspective sau prin componenta specifică a sistemului pe care o descriu

Domeniul informațional

- Modelul informațional
- Modelul semantic
- Modelul datelor

Domeniul funcțional

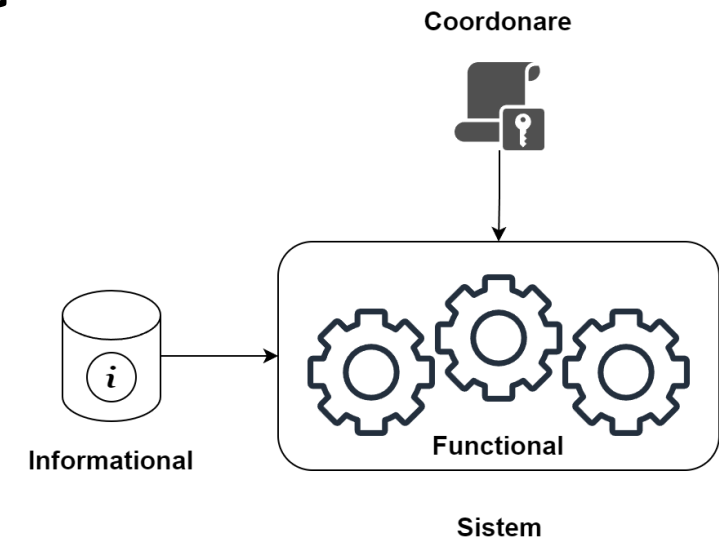
- Modele funcționale
- Modele de execuție
- Modele de stare
- Modele de proces

Domeniul de coordonare

- Modele de interfațare
- Modele de coordonare
- Protocoale
- Modele de prezentare

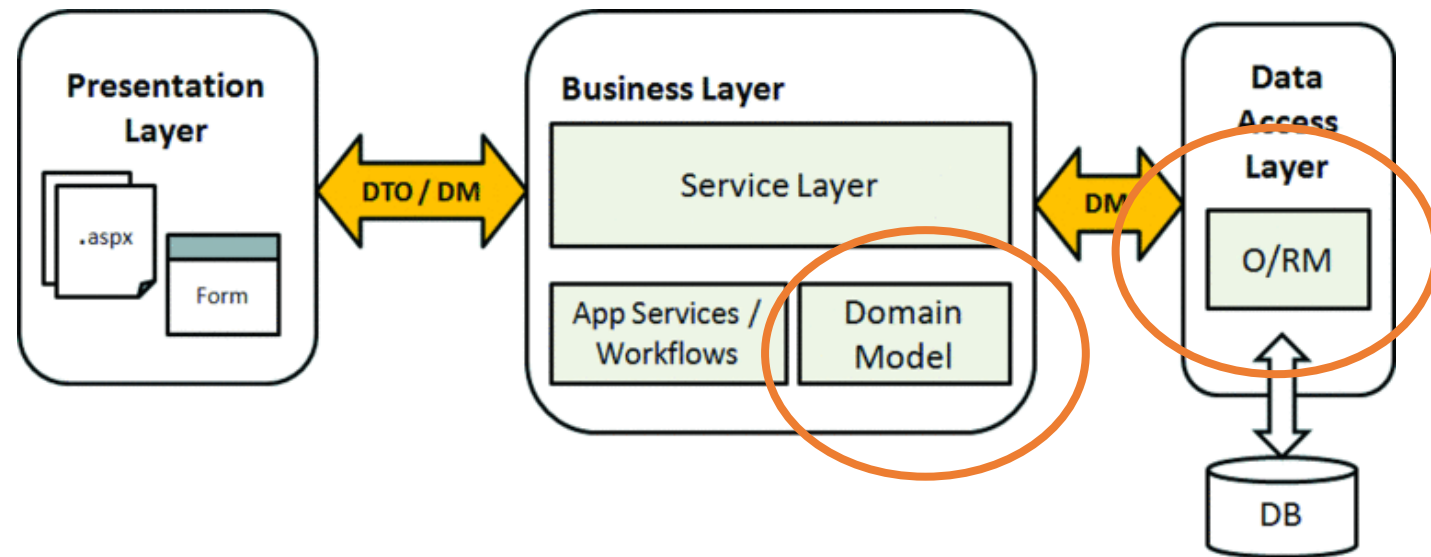
Domeniul de sistem

- Modele de structură
- Modele de politică
- Modele de rețea



Modelarea domeniului – tehnic

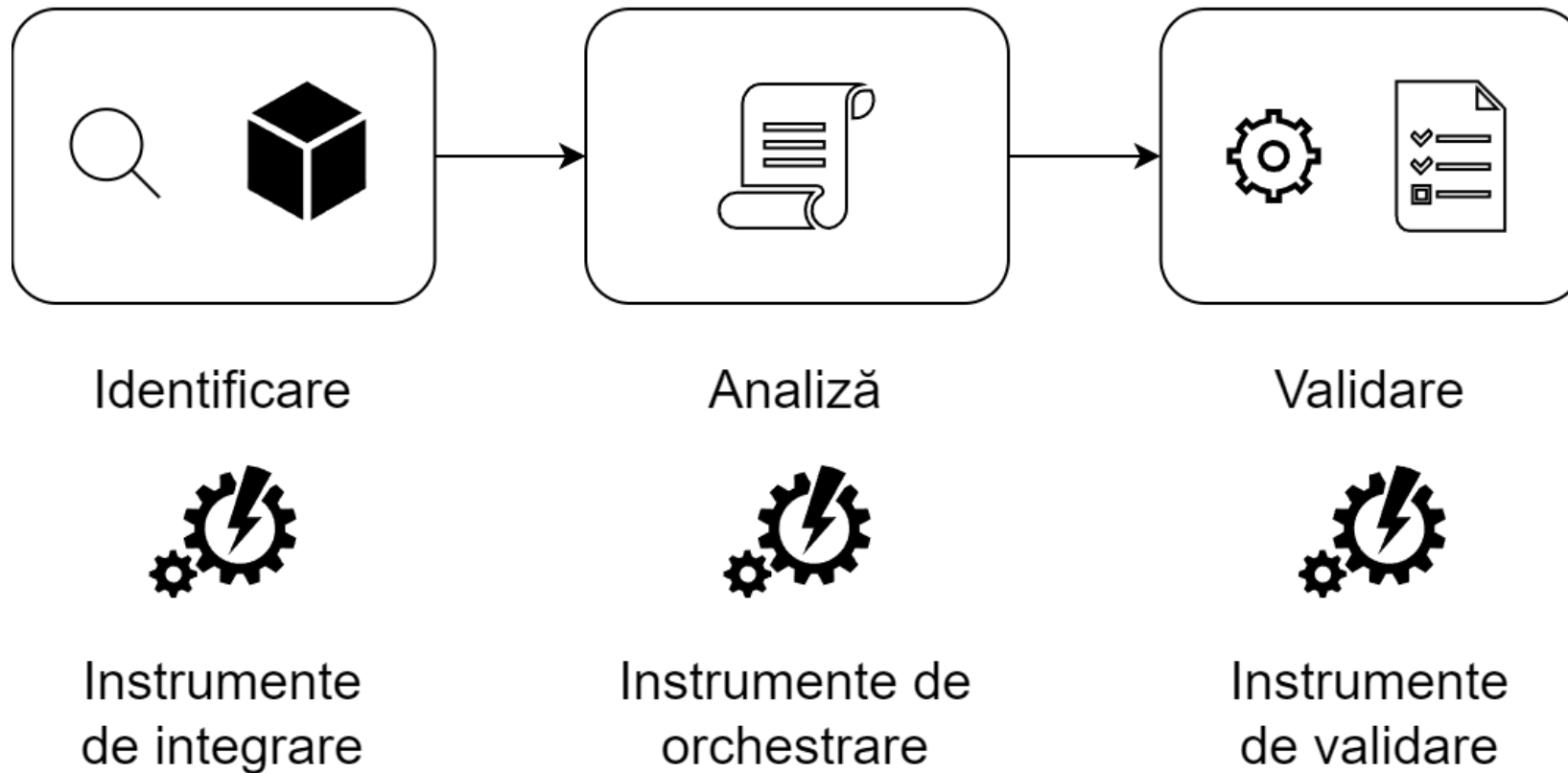
- Modelul datelor – Domain Model / Entity Data Model
- Modelul funcțional – Business Logic
- Nivelul de translație
 - DTO (Data Transfer Object)
 - ORM (Object Relational Mapping)



Sursa: Microsoft Learn

Ingineria sistemelor

Ingineria sistemelor software



Modelarea specificațiilor

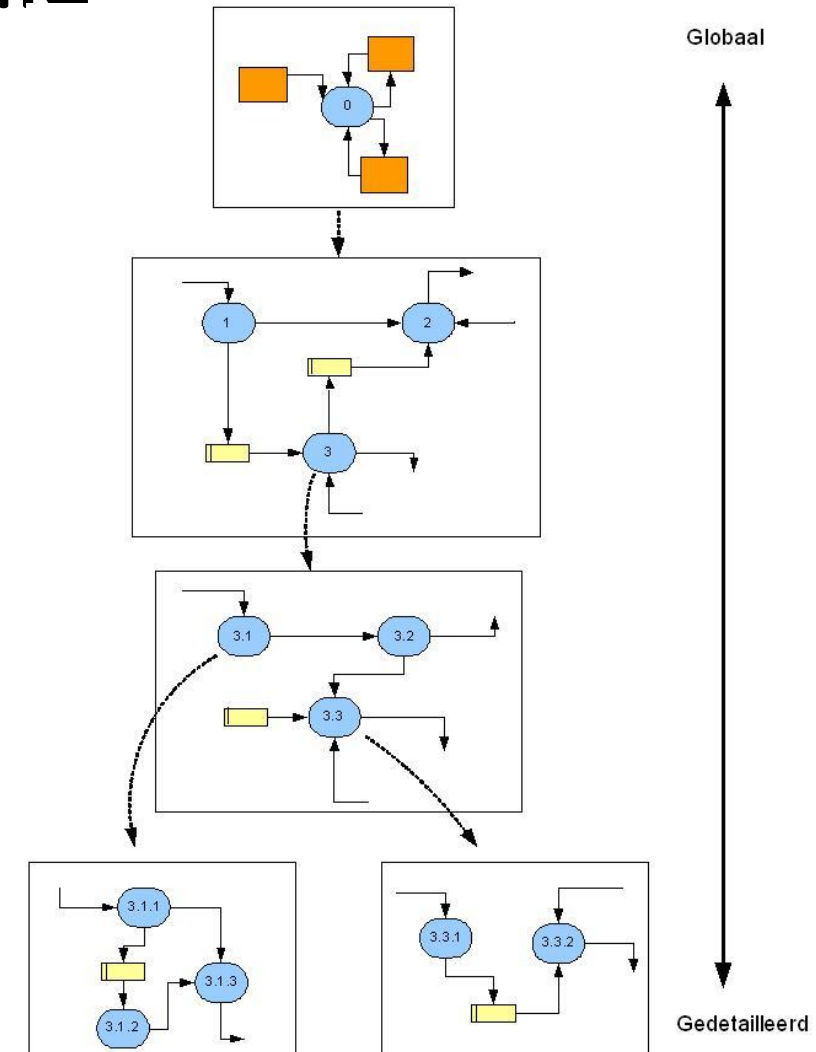
Ingineria specificațiilor

- Cerințe
 - **funcționale** – descriu funcționalitatea sistemului (use case)
 - **juridic-contractuale** – clauze de întreținere, drept de proprietate intelectuală, versiuni ulterioare – update-uri
 - **tehnologice** – scalabilitate, extensibilitate, etc.
 - **de calitate** – durata maximă de nefuncționare, etc.
 - **asistarea utilizatorului** – legătura dintre pagini, workflow
 - **modularizare** – ce pachete se livrează
 - **activități** – planificare



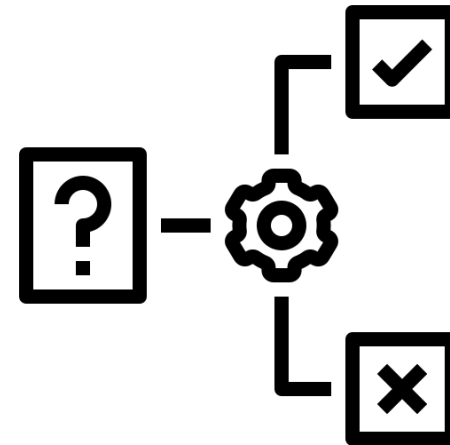
Modelarea funcțională

- Modelarea **activităților**
 - reprezintă succesiunea pașilor printr-un graf parțial ordonat
 - fiecare task este împărțit în subtask-uri interdependente
- Modelarea **proceselor**
 - formă specială de modelare a activităților
 - se axează pe interacțiuni – input-urile și output-urile părților componente ale sistemului

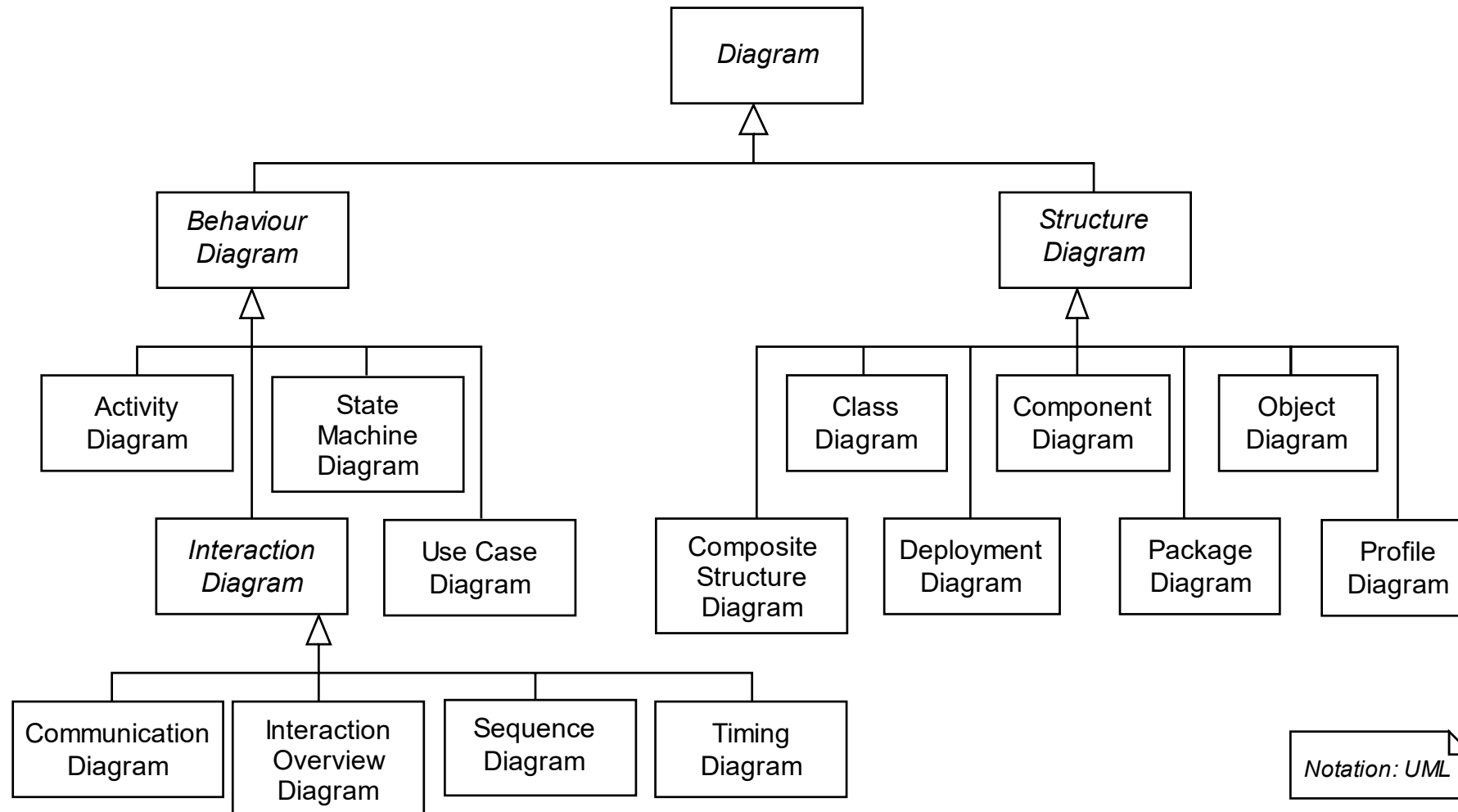


Modelare bazată pe reguli

- Modele ce descriu **funcții și comportamente** ale sistemelor ca un set de secvențe **situație – răspuns**
- Reprezentate prin
 - Pseudocod
 - Tabele de decizie
 - Arbori de decizie
- Condiții
 - Completitudine
 - Consistență



Diagrame UML



Domeniul aplicației / soluției tehnice

Contextul modelării UML

- Domeniul aplicației (analiza cerințelor)
 - Mediul în care operează sistemul
- Domeniul soluției tehnice (proiectarea sistemului)
 - Tehnologiile disponibile pentru construirea sistemului



Diagrame UML

- Diagrame de **cazuri de utilizare** – descriu **comportamentul funcțional** al sistemului așa cum este văzut de utilizator.
- Diagrame de **clasă** – descriu **structura statică** a sistemului
 - Obiecte, attribute, asocieri
- Diagrame de **secvență** – descriu **comportamentul dinamic** dintre actori și sistem sau între componentele sistemului
- Diagrame de **stare** – descriu comportamentul dinamic al unui obiect individual din cadrul sistemului
 - Automat de stare
- Diagrame de **activitate** – modelează **comportamentul dinamic** al unui sistem, în special **fluxul de lucru**
 - Nume alternativ: diagrame de flux

Întrebări?

