
Analiza și proiectarea sistemelor software

Curs 12

PLAN CURS

MDE (Model Driven Engineering) & MDWE (Model Driven Web Engineering)

IFML (Interaction Flow Modeling Language)

WebRatio Platform

Alte tehnologii de modelare și proiectare a aplicațiilor Web

MDE – Model Driven Engineering

Def. Model – reprezentare simplificată, abstractă a structurii unui sistem complex.

Reflectă o selecție relevantă de proprietăți ale sistemului.

Utilizabil ca înlocuitor al sistemului în raport cu anumite scopuri.

Utilizare tradițională:

- comunicare cu clienții (specificare cerințe, prototipare)
- suport pentru proiectarea software-ului
- specificații pentru programatori
- reprezentare cod (reverse engineering)

Nivele de precizie:

Model ca schiță

- comunicare de idei și alternative

Model ca ghid de implementare

- documentare decizii de proiectare

Model ca program

- generare automată de aplicații

Def. Model-driven engineering - abordare a dezvoltării de software în care un sistem este reprezentat ca set de modele ce pot fi transformate automat în cod executabil.

MDA – Model Driven Architecture

Def. Model Driven Architecture - abordare focalizată pe model a *proiectării și implementării* software-lui care folosește un subset al modelelor UML pentru a descrie un sistem.

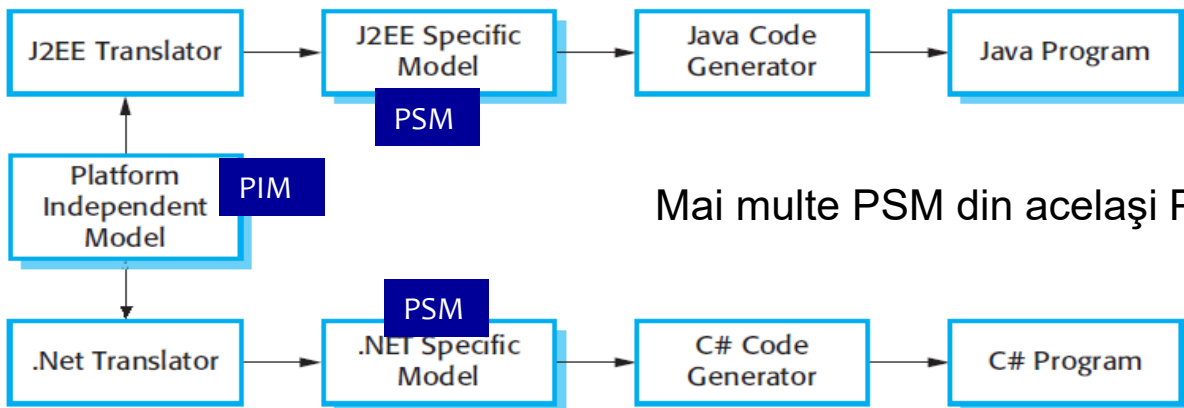
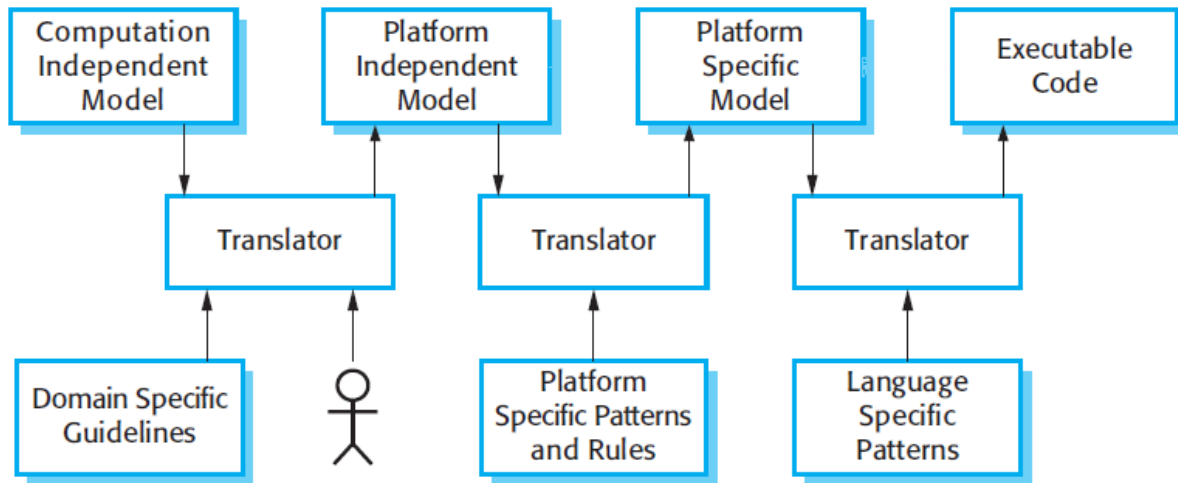
- Dintr-un model de nivel înalt, independent de platformă este posibil, în principiu, să se genereze fără intervenție manuală un program funcțional.

Categorii de modele în MDA:

- *Computation independent model (CIM)*
 - Modelează *abstractizările* importante de domeniu utilizate într-un sistem. CIM-urile sunt numite uneori *modele de domeniu*.
- *Platform independent model (PIM)*
 - Modelează modul de funcționare a unui sistem fără a face referire la implementarea sa.
 - Sunt descrise de obicei folosind modele UML care ilustrează structura statică a sistemului și modul în care acesta răspunde la evenimentele externe și interne.
- *Platform specific models (PSM)*
 - Transformări ale modelului independent de platformă (PIM), cu câte un PSM pentru fiecare platformă de execuție.

MDA – Model Driven Architecture

Transformările MDA



MDWE – Model Driven Web Engineering

MDWE (*Model Driven Web Engineering*) = MDE pentru aplicații web.

Utilizarea paradigmei MDE în metodologiile de dezvoltare de aplicații Web.

Reprezentarea *conceptelor* folosind *meta-modele* ce sunt independente de platformă.

Procesul de dezvoltare – sprijinit de un set de *transformări și relații între concepte*



- permite dezvoltare agilă,
- asigură consistență între modele.

MDWE – Model Driven Web Engineering

Metodologii MDWE (Exemple)

- (OOHDM – hypermedia design method) – modele fundamentale
 - conceptual
 - navigațional
 - interfață abstractă

Utilizare diagramă de clase pentru modelul conceptual și o extensie a diagramei de clase pentru modelul navigațional

- UWE – UML Web Engineering (tool – MagicUWE)
- HDM → HDM2 → HDM2000/W2000 → WebML→IFML (tool – WebRatio)
- WebRE (Web Requirements)

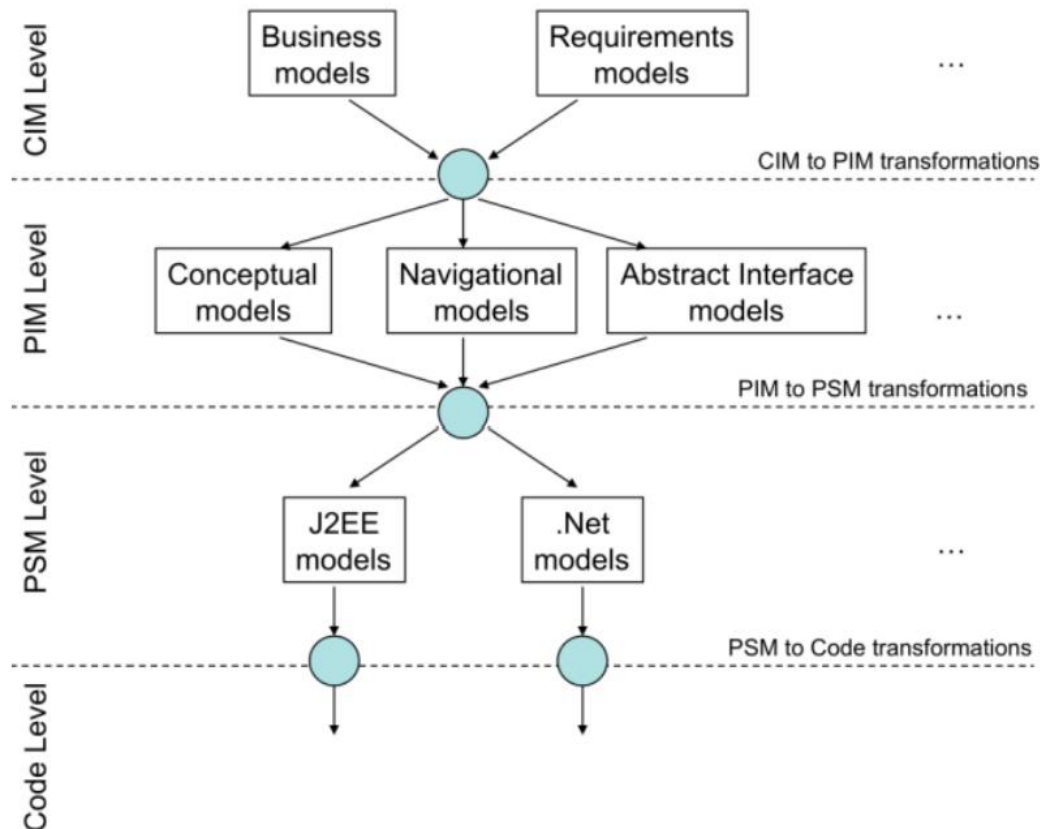
MDWE – Model Driven Web Engineering

CIM - Concepte ce capturează logica sistemului.

PIM - Concepte ce definesc sistemul software fără referire la o platformă de dezvoltare specifică.

PSM – modele dependente de o platformă specifică

Code Model -
Implementarea sistemului



Transformări: CIM-to-PIM, PIM-to-PSM, PSM-to-code,
dar și PIM-to-PIM, bazate pe definițiile din *metamodels*

Evaluare formativă

1. Prin ce se caracterizează abordarea MDE în dezvoltarea de software ?
2. Care sunt categoriile de modele în MDA ?

<https://forms.gle/YGrxqAda7PWsruHd6>

PLAN CURS

MDE (Model Driven Engineering) & MDWE (Model Driven Web Engineering)

IFML (Interaction Flow Modeling Language)

WebRatio Platform

Alte tehnologii de modelare și proiectare a aplicațiilor Web

IFML

IFML – Interaction Flow Modeling Language*

Modelarea independentă de platformă (PIM) a *interfețelor grafice* ale aplicațiilor *business-oriented* și *data-intensive* precum și a *legăturii* acestora *cu modelul domeniului* și *cu logica business* a aplicației.

Limbaj focalizat pe *proiectare front-end (UI)*.

Delegare către alte modele a :

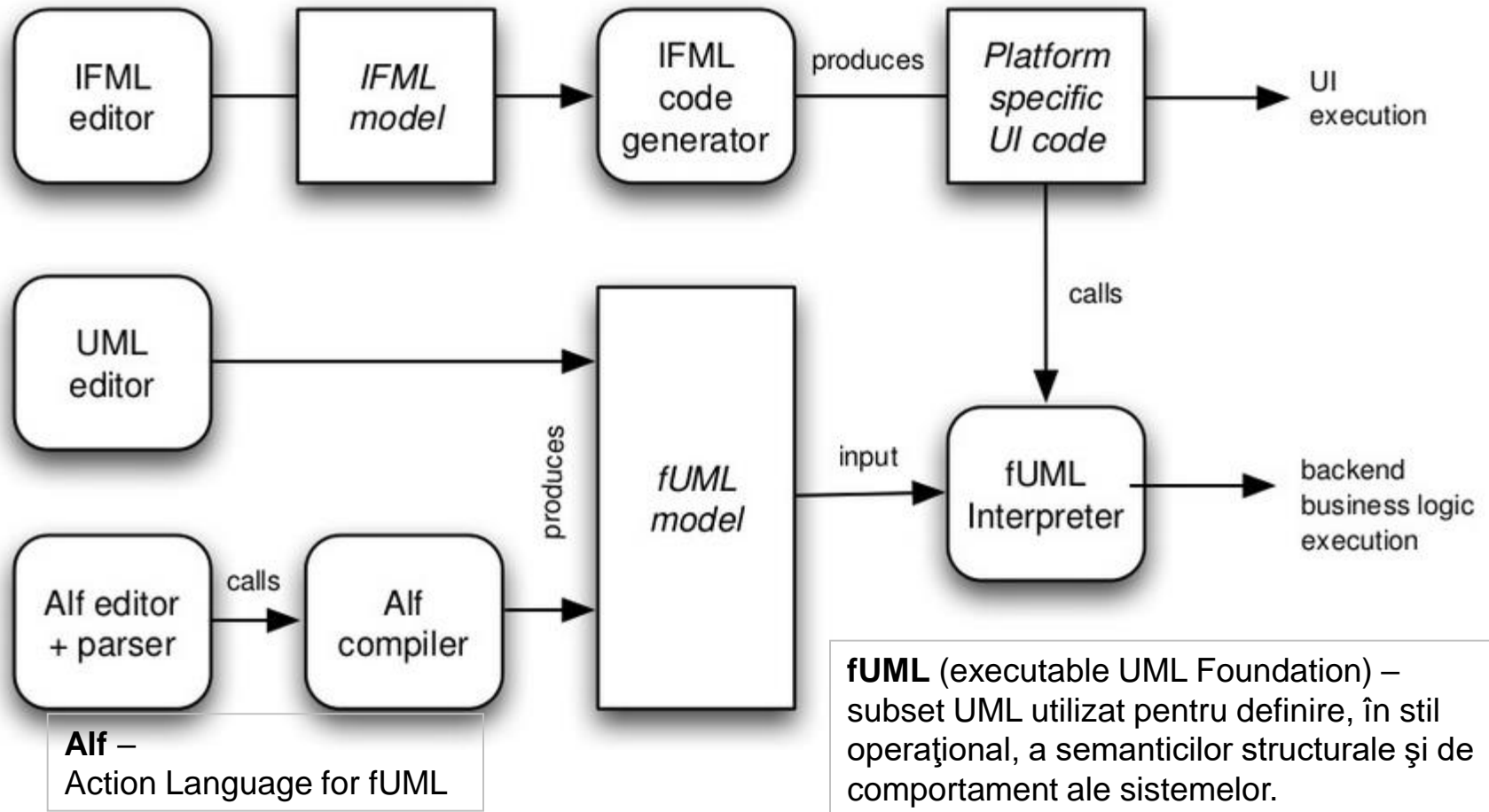
- modelării domeniului (ex. diagrame UML de clase, diagrame ER, ontologii)
- modelării orchestrării componentelor ce conțin logica business (ex. diagrame UML de secvențe sau activitate).

Obs. - **nu** e destinat aplicațiilor cu grafică bi/tri-dimensională, jocuri video, alte aplicații puternic interactive.

*www.omg.org/spec/IFML/Current

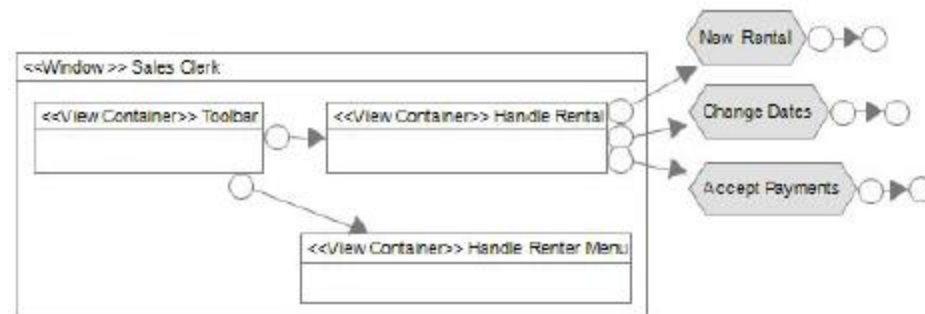
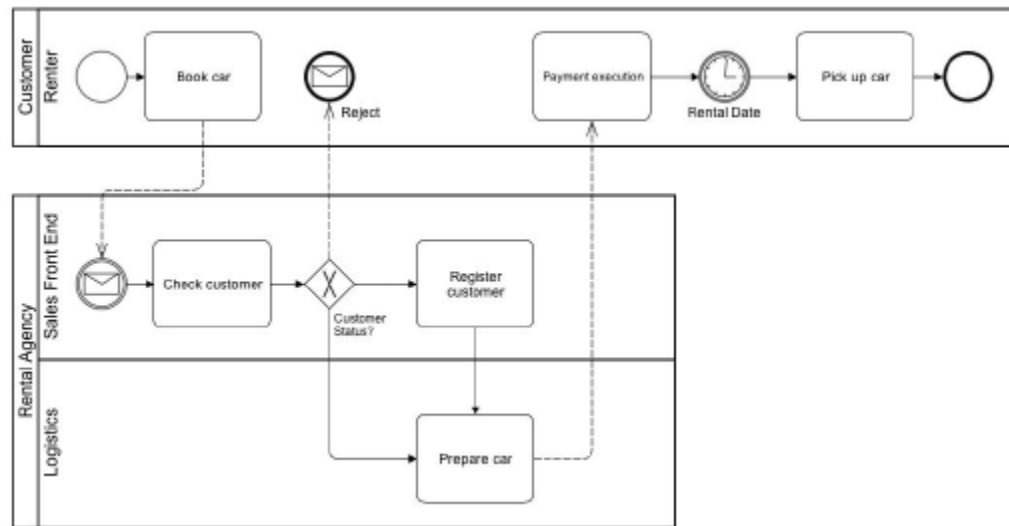
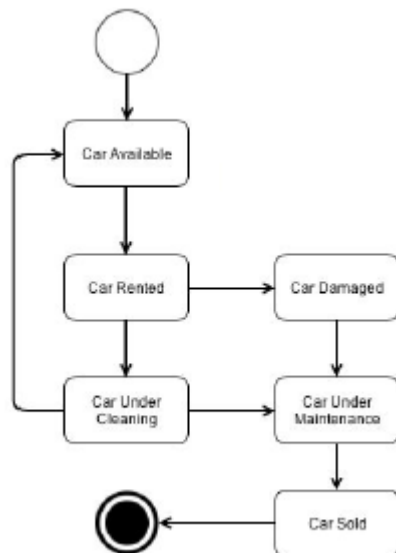
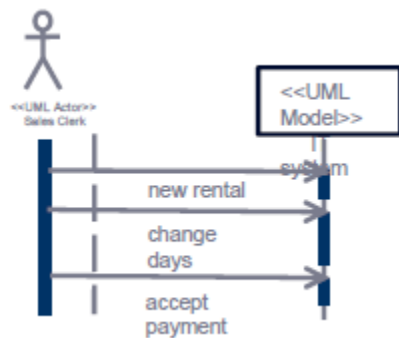
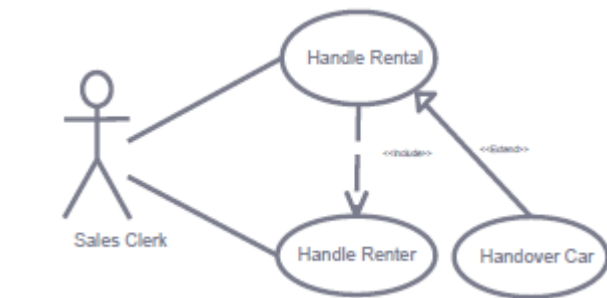
IFML în MDWE

Propunere procedură

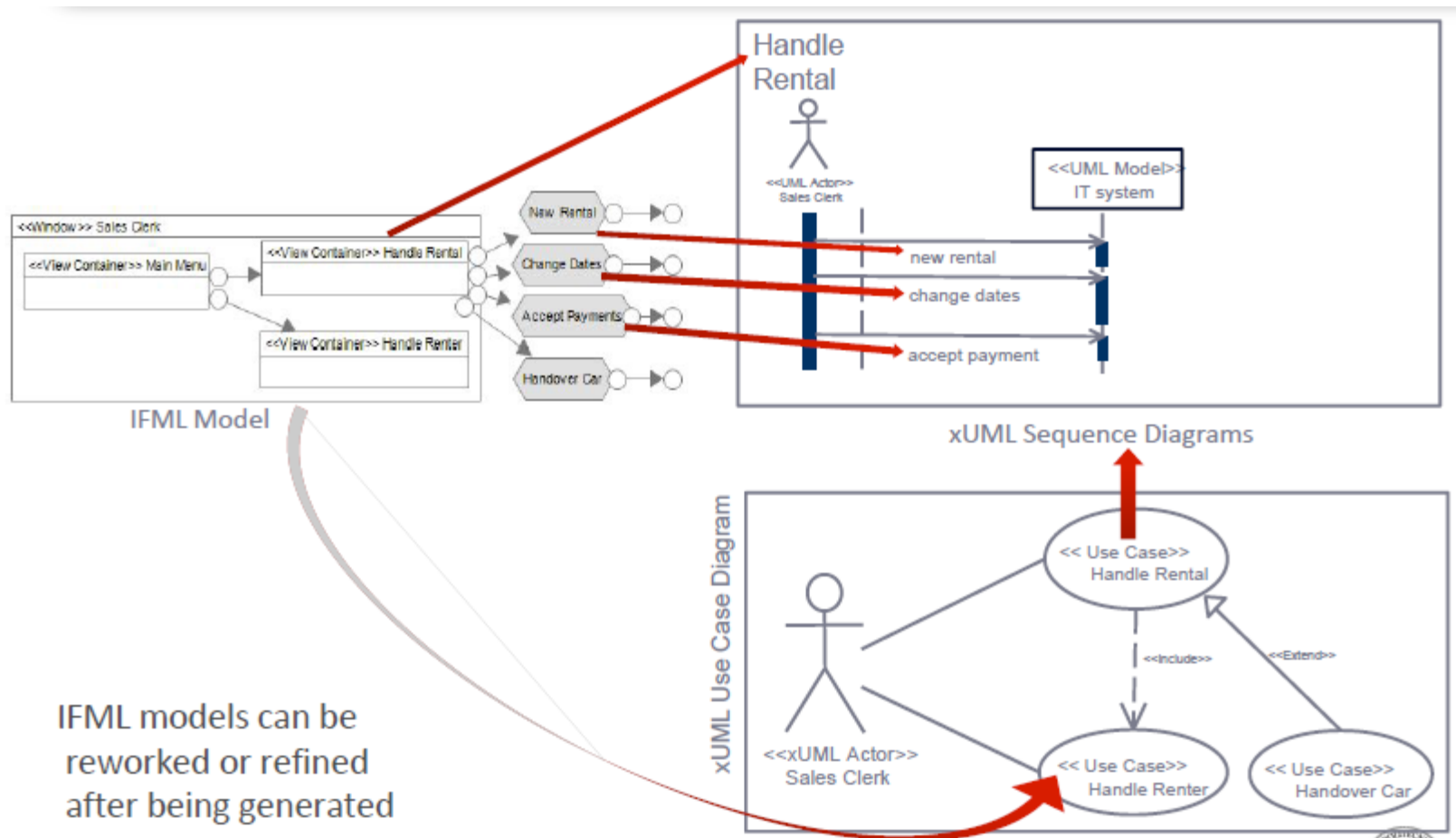


Integrare IFML în modelare app web

Analiză (analiza procesului business și modelarea cerințelor)

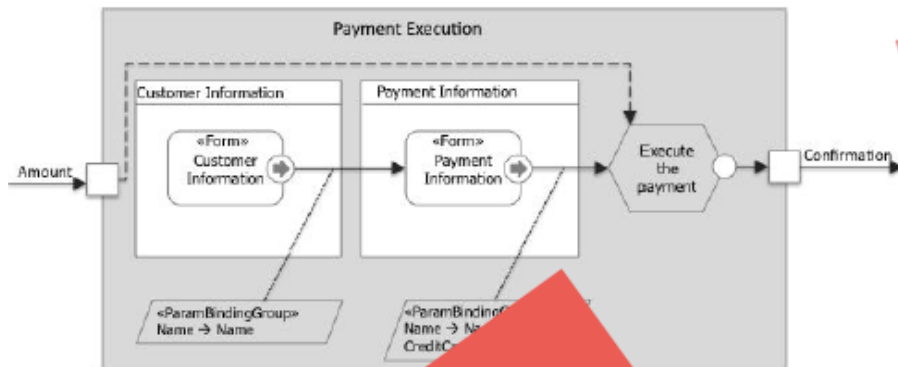


Mapare IFML - UML

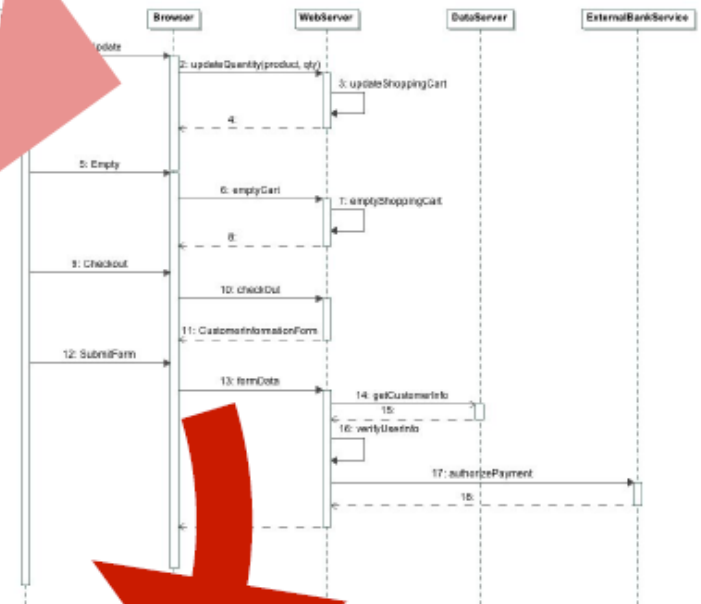


Integrare IFML în modelare app web Proiectare

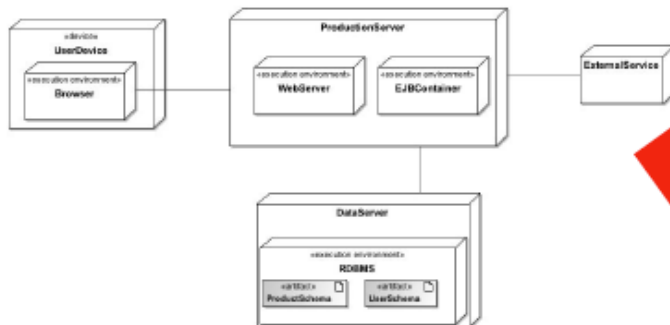
IFML



UML Sequence



UML Deployment



UI Mockup models



IFML

Exprimarea:

- conținutului
- interacțiunii cu utilizatorul
- controlului comportamentului

pentru front-end-ul aplicațiilor din următoarele domenii:

- aplicații web tradiționale (HTML+HTTP(S))
- RIA (Rich Internet Apps) suportate de HTML5
- aplicații mobile
- aplicații client-server
- aplicații desktop
- interfețe embedded om-mașină pentru aplicații de control
- aplicații multichannel și context-aware

IFML

Exprimarea:

- conținutului
- interacțiunii cu utilizatorul
- controlului comportamentului

Categoriile de elemente modelate cu IFML:

- structura prezentării conținutului (model compoziție site)
- legarea la modelul domeniului (stratul de persistență)
- căile de navigare
- evenimentele
- legarea la logica business

IFML – perspectivele de proiectare

View structure specification – definirea *containerelor de vederi*, relațiilor lor de încuibare, vizibilitatea și accesibilitatea acestora.

View content specification – definiția *componentelor de tip vedere* (i.e. conținutul și elementele de intrare/afișare date conținute în containerele de vederi).

Events specification – definiția *evenimentelor* ce pot afecta *starea UI*.
Evenimentele pot fi produse de gesturile utilizator, de aplicație sau de sisteme externe.

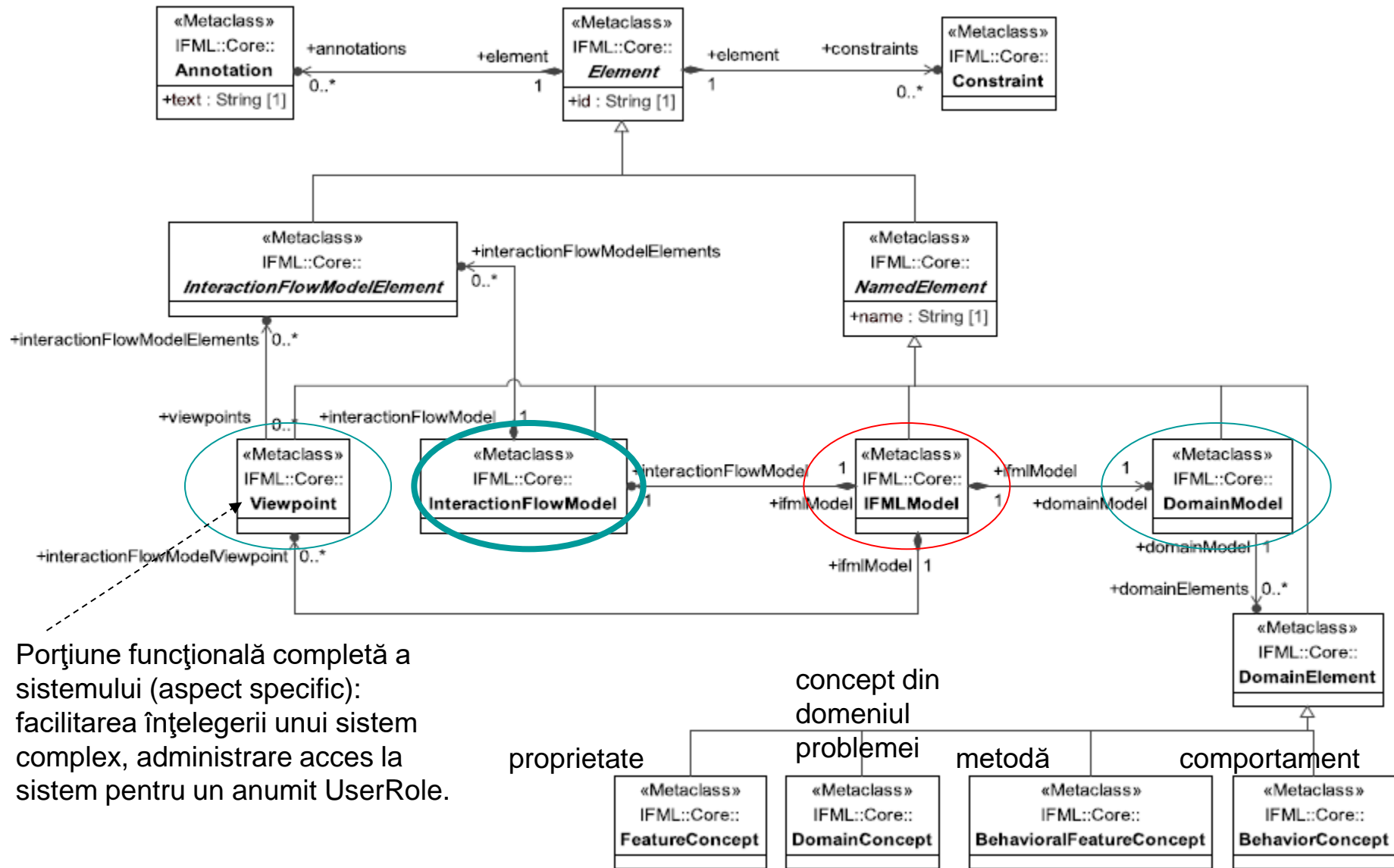
Event transition specification – definirea *efectului* unui eveniment *asupra UI*; apariția unui eveniment provoacă o tranziție de stare care produce o modificare în UI.

Evenimentele sunt atașate la componente și efectul unui eveniment este reprezentat de o conexiune din ***fluxul interacțiunii*** de la componenta sursă a evenimentului la componenta destinație a acestuia, modelând astfel o schimbare de stare a UI.

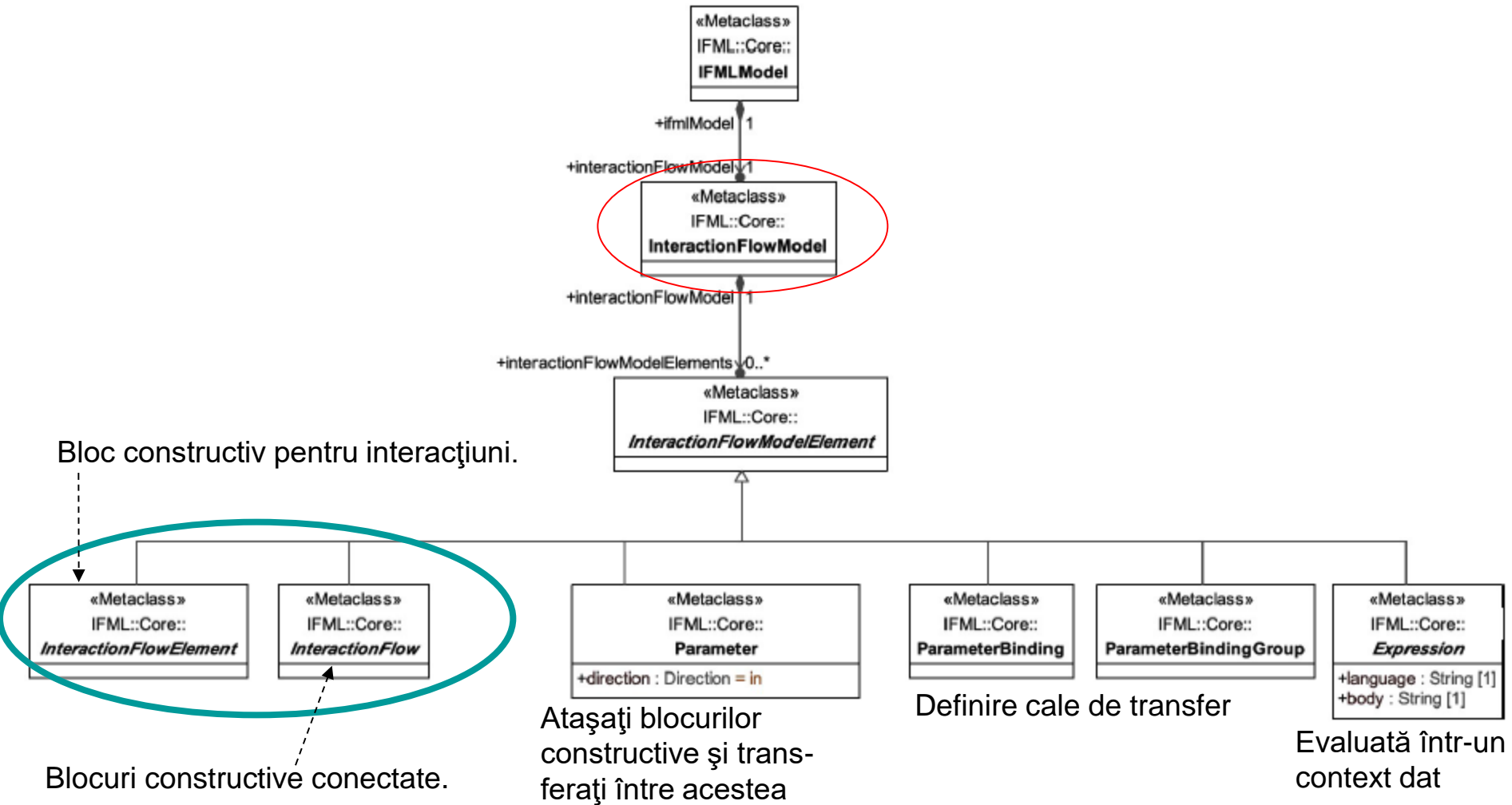
Parameter binding specification – definiția dependențelor I/E *între componente și între componente și acțiuni*.

Referințele la acțiunile declanșate de *gesturile utilizator*.

IFML - metamodel

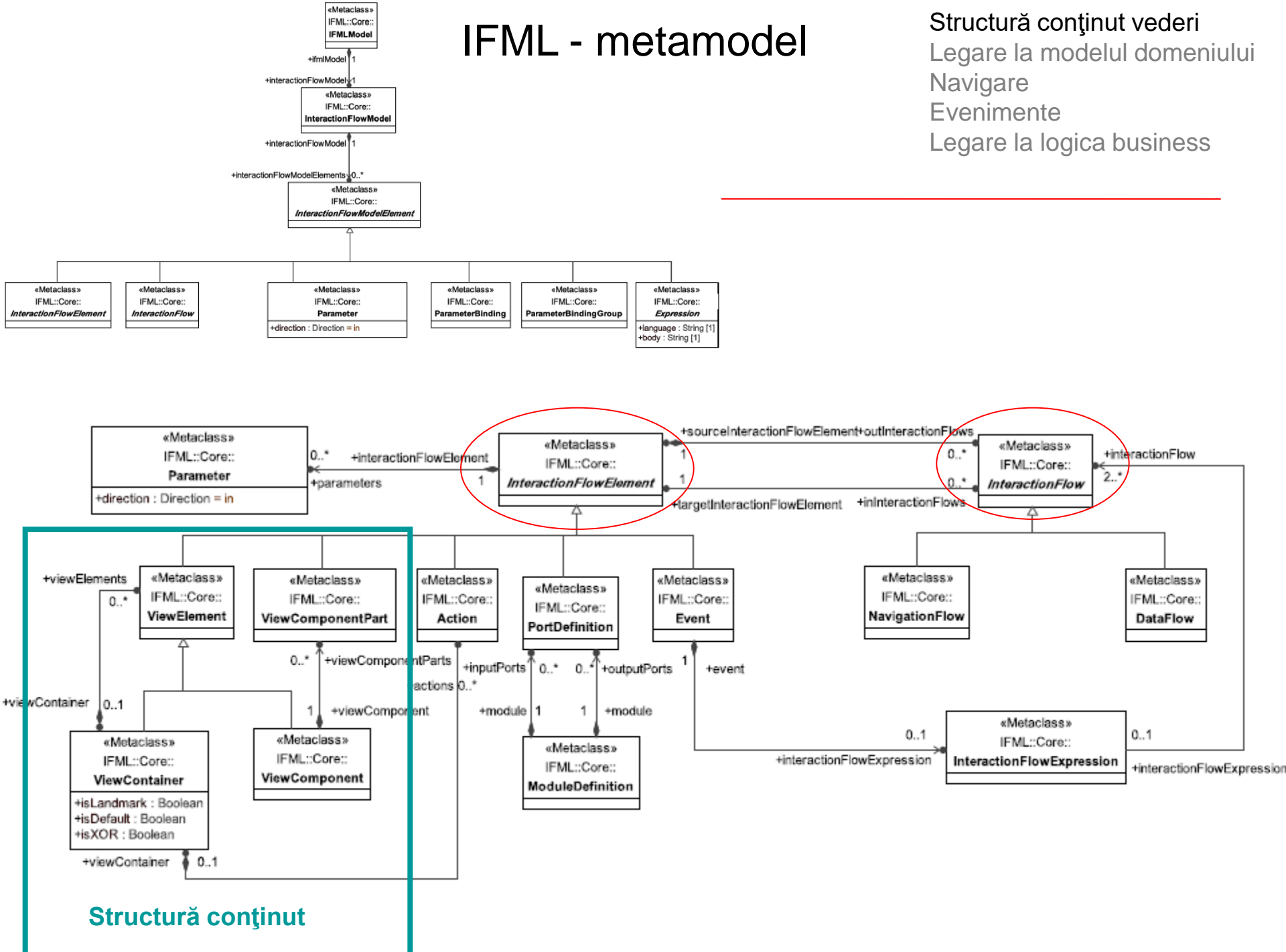


IFML - metamodel



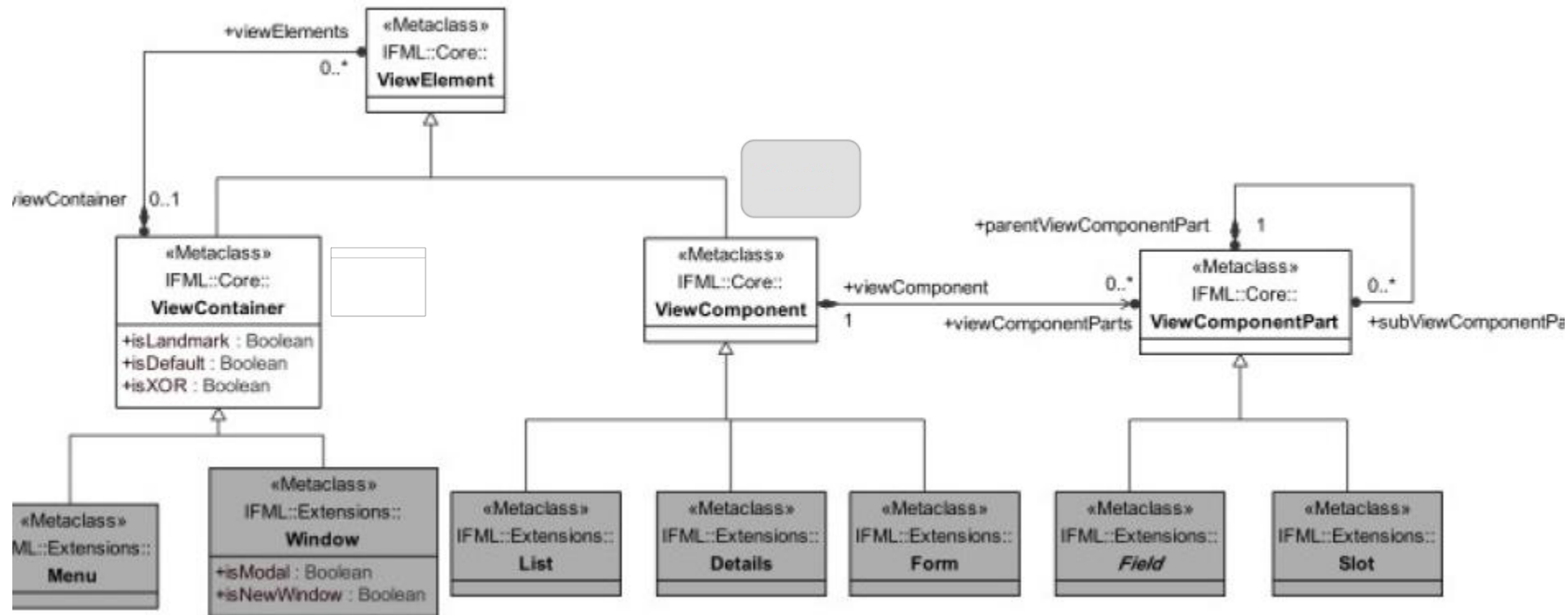
IFML - metamodel

Structură conținut vederi
Legare la modelul domeniului
Navigare
Evenimente
Legare la logica business



IFML – Modelare structură conținut vederi

Structură conținut vederi
Legare la modelul domeniului
Navigare
Evenimente
Legare la logica business

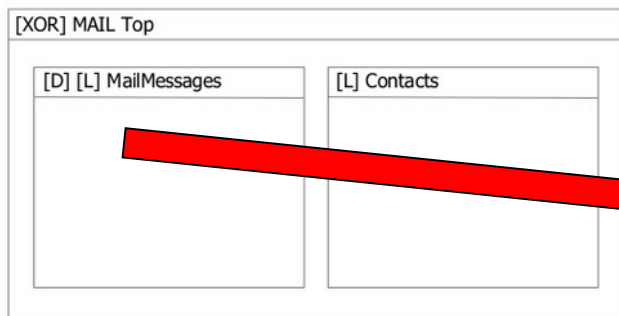


IFML - Modelare compoziție conținut vederi

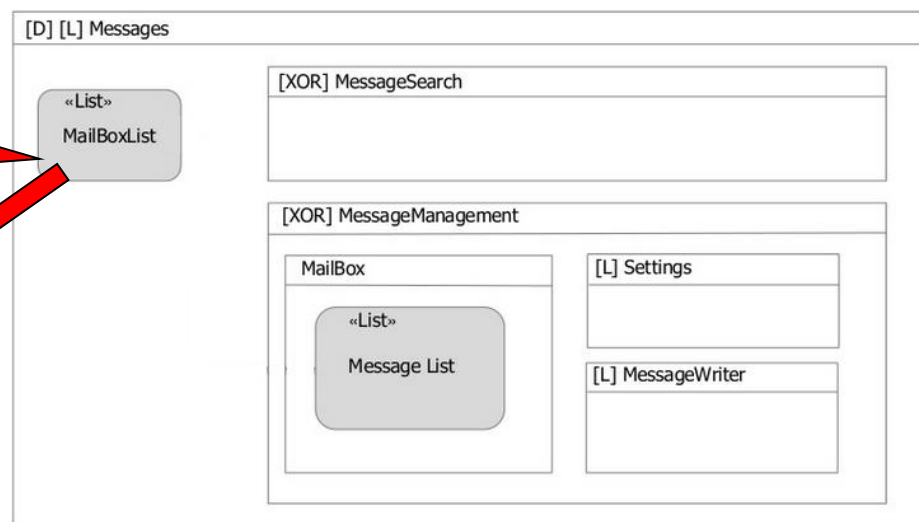
Exemplu

Structură conținut vederi
Legare la modelul domeniului
Navigare
Evenimente
Legare la logica business

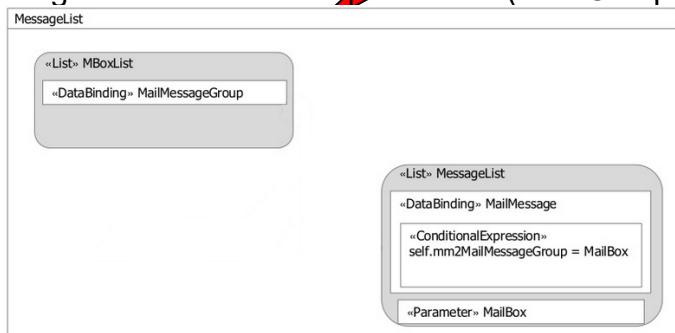
1. Specificare containere (ViewContainer de tip Window)



2. Adăugare componente (ViewComponent de tip List)



3. Adăugare elementele componentelor (ViewComponentPart)



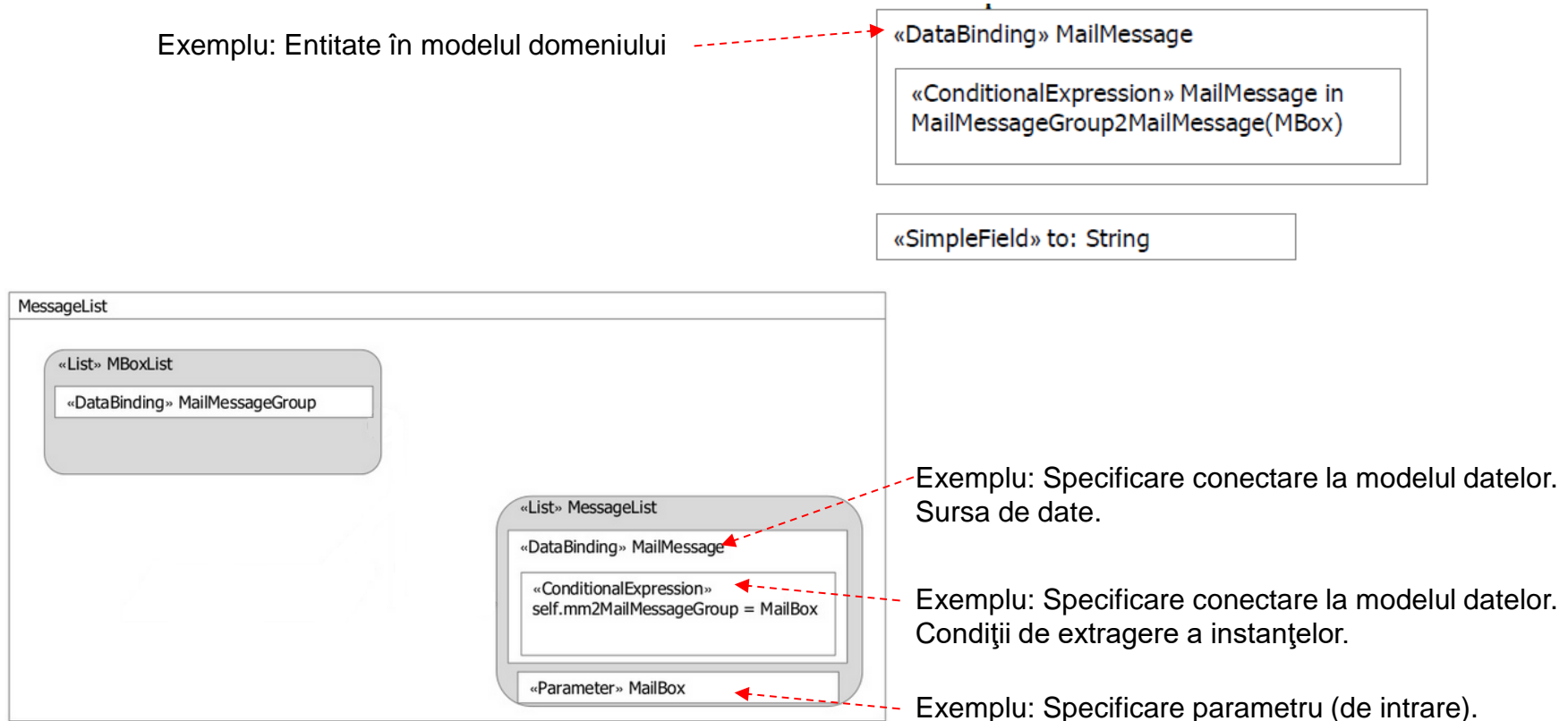
« *TypeName* » [Name][...]

« *SubTypeName* » [Name] [...]

« *SubTypeName* » [Name][...]

IFML - Modelarea legării la modelul domeniului

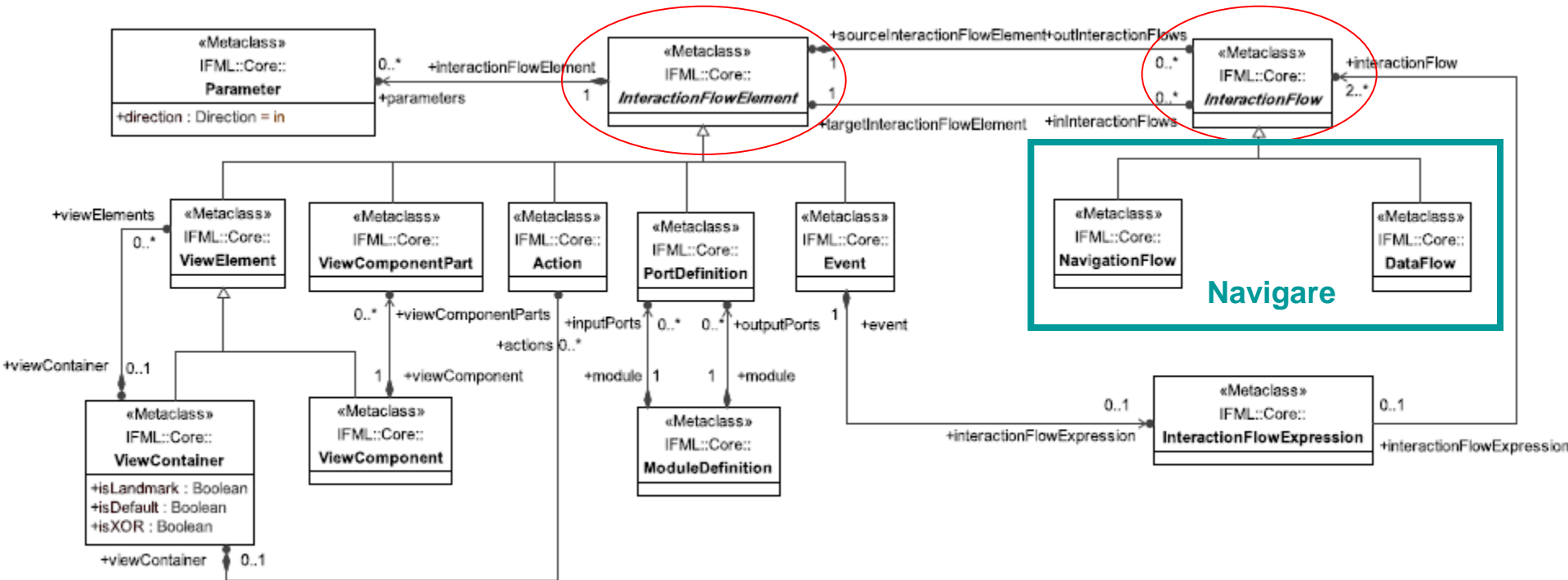
Exemplu: Entitate în modelul domeniului



Stratul de persistență poate fi modelat cu diagrame UML, diagrame ER (folosite în WebRatio), ontologii.

IFML – modelare navigare

Structură conținut vederi
Legare la modelul domeniului
Navigare
Evenimente
Legare la logica business



IFML – modelare navigare

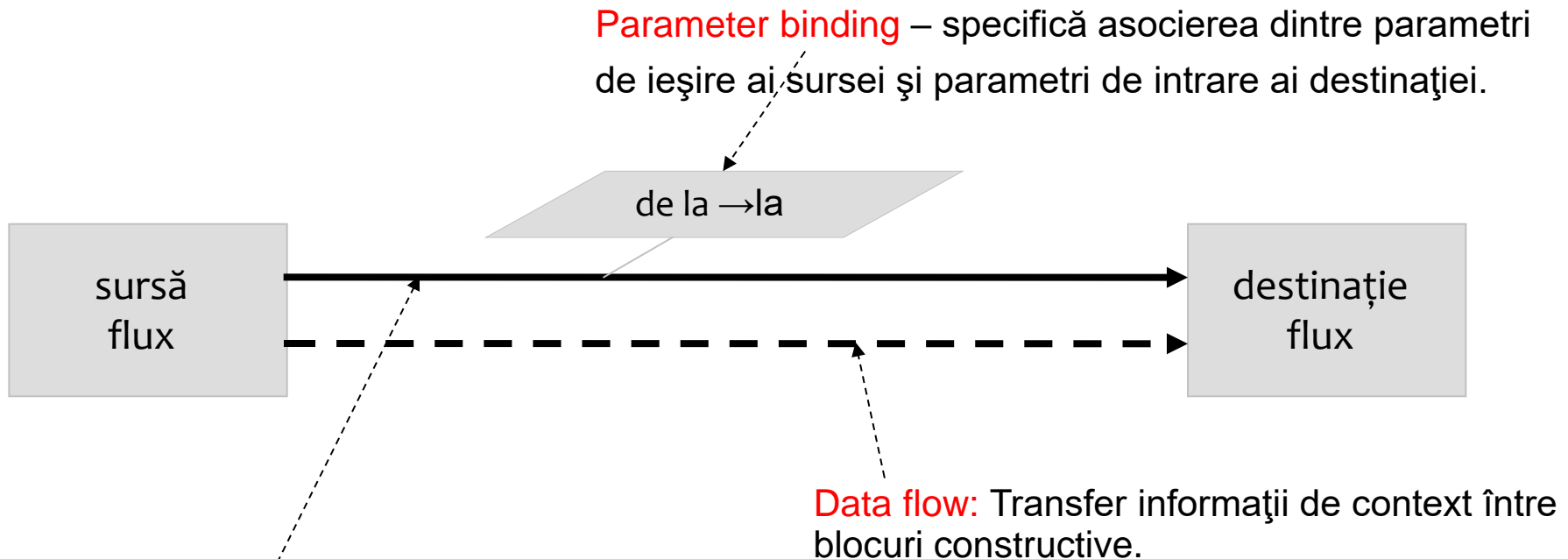
Navigare = topologia legăturilor dintre pagini și unități

Flow = cale de navigare urmată la apariția unui eveniment declanșator

Cale de navigare – secvență de interacțiuni cu utilizatorul care îi permite acestuia să obțină un anumit serviciu oferit de aplicația Web.

Căi de navigare predefinite – pentru ghidarea utilizatorului.

IFML – modelare navigare



Navigation flow : Declanșat de evenimente. (Poate fi însoțit de transfer de parametri.)

Funcții:

- schimbare focus către alt *ViewElement*
- lansare/răspuns *Action*

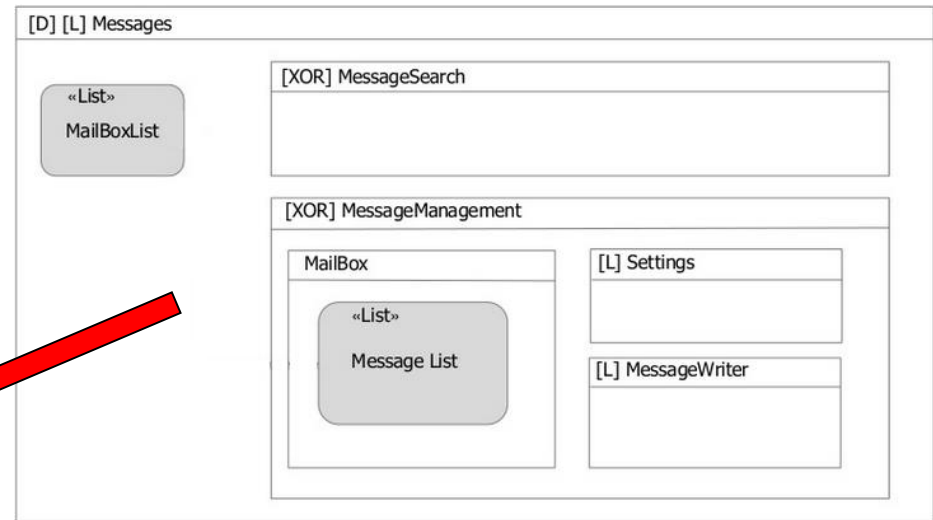
IFML - Modelare navigare

Exemplu

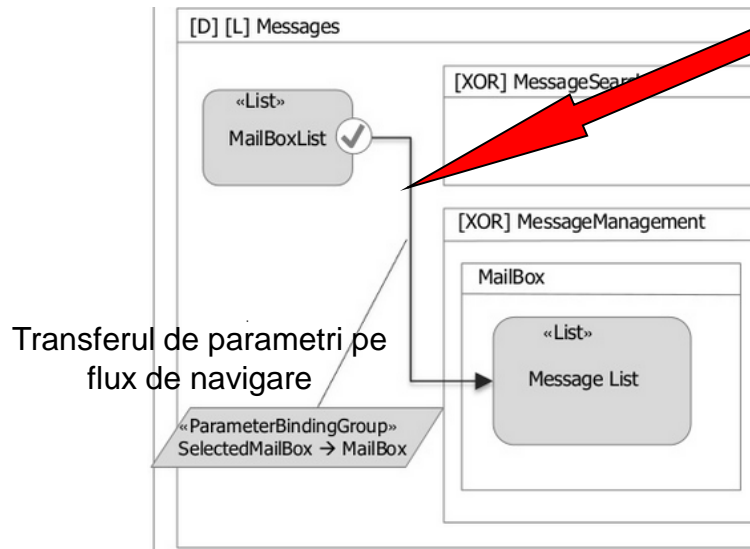
1. Specificare containere (ViewContainer de tip Window)



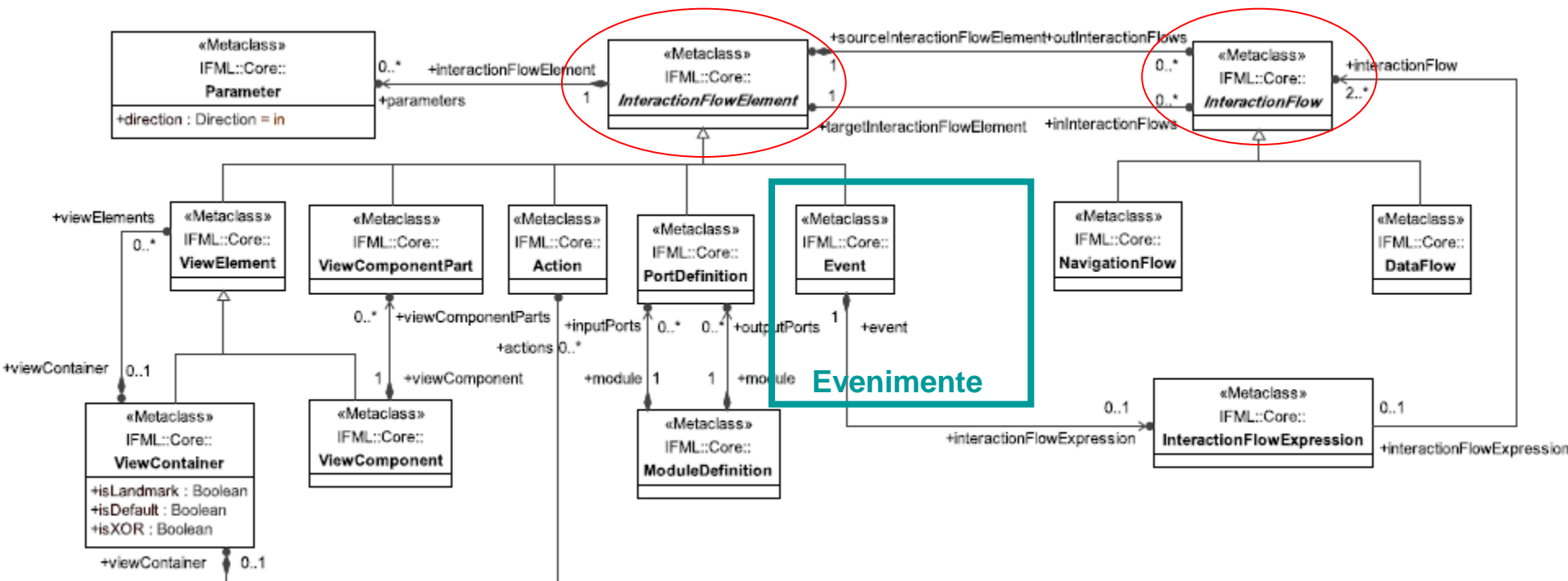
2. Adăugare componente (ViewComponent de tip List)



3. Specificare flux de navigare.



IFML – modelare evenimente



IFML – modelare evenimente

Eveniment : Fenomen a cărui apariție poate afecta starea aplicației.

Categorii:

CatchingEvent – *capturat* de interacțiunea modelată și folosit ca declanșator de navigare și/sau de transfer parametri între elemente ale modelului.

Produs de:

- gest utilizator (`ViewElementEvent`)
- terminarea unei acțiuni (`ActionEvent`)
- notificare sistem (`SystemEvent`) (ex. expirare timer, întrerupere legătură la BD)

Exemple: `OnSelect`



`OnSubmit`



ThrowingEvent – *generat* de interacțiunea modelată

Poate fi capturat de un `CatchingEvent`

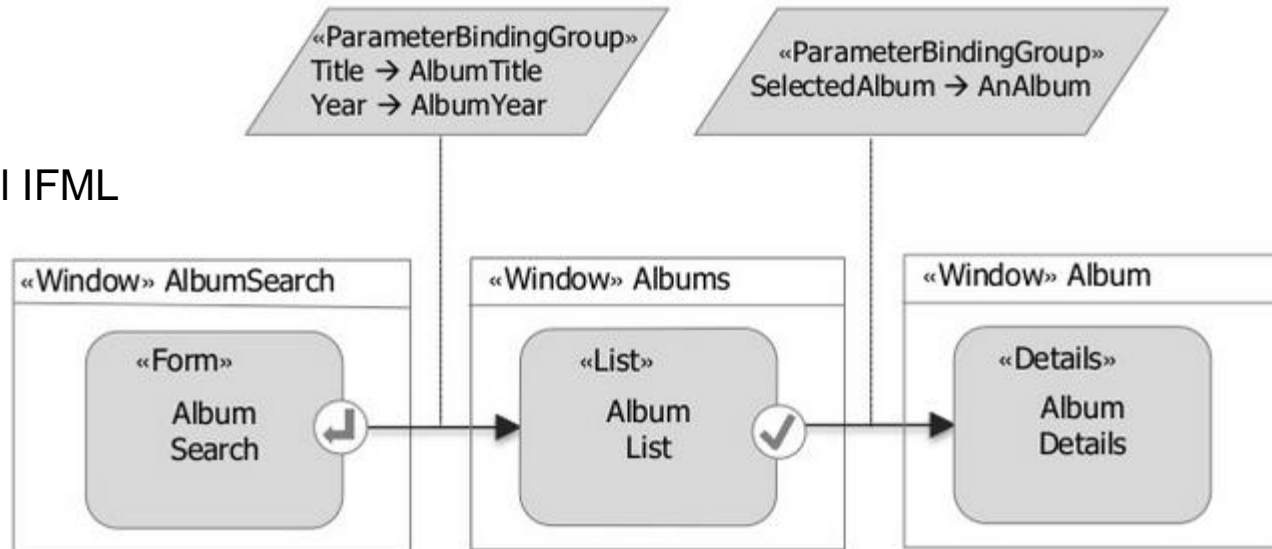


IFML - Modelare evenimente

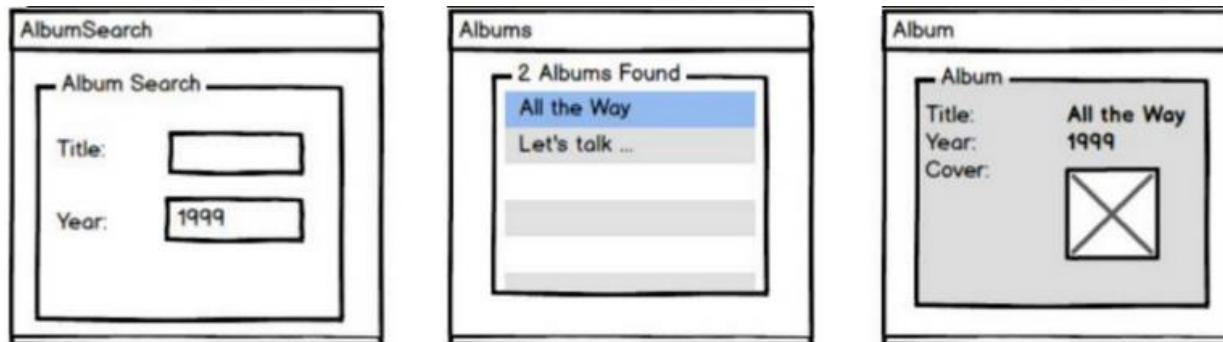
Structură conținut vederi
Legare la modelul domeniului
Navigare
Evenimente
Legare la logica business

Exemplu

Model IFML

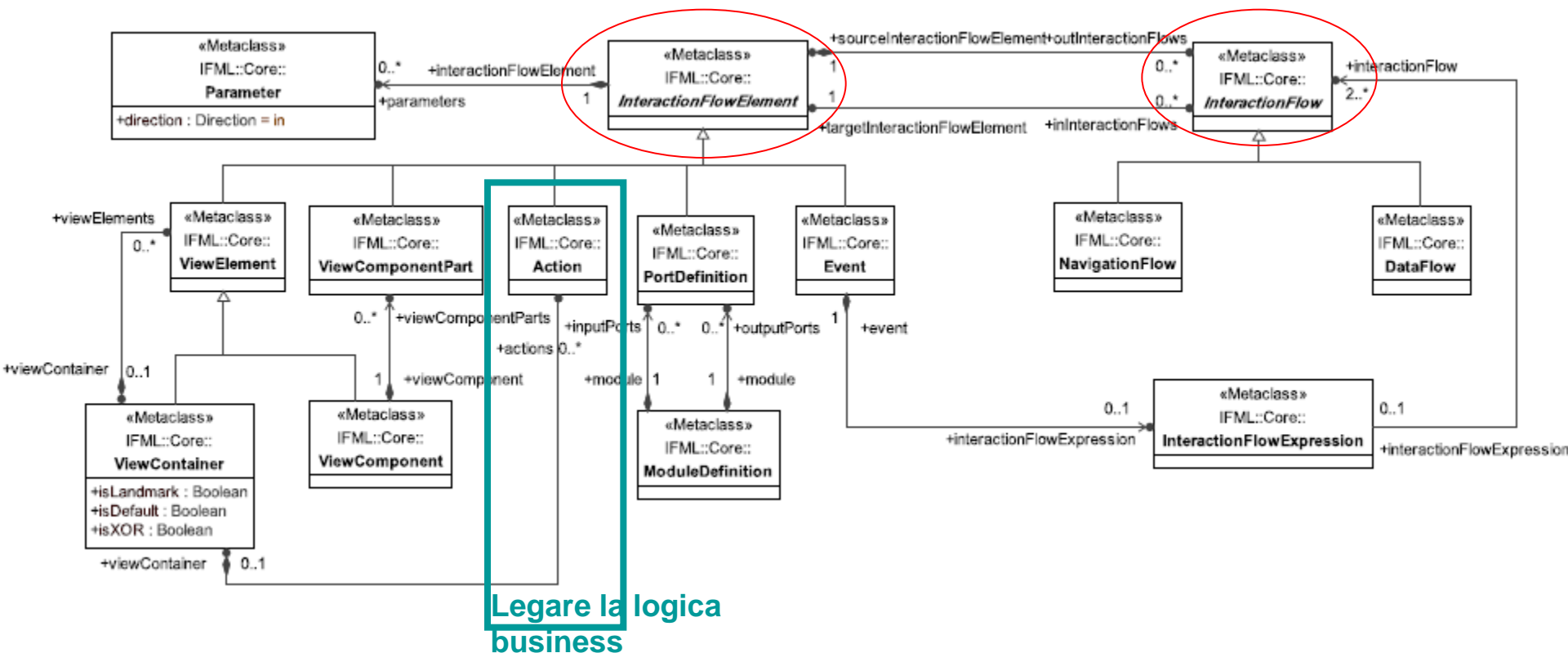


Interfață corespunzătoare în aplicație



IFML – modelare legarea la logica business

Structură conținut vederi
Legare la modelul domeniului
Navigare
Evenimente
Legare la logica business

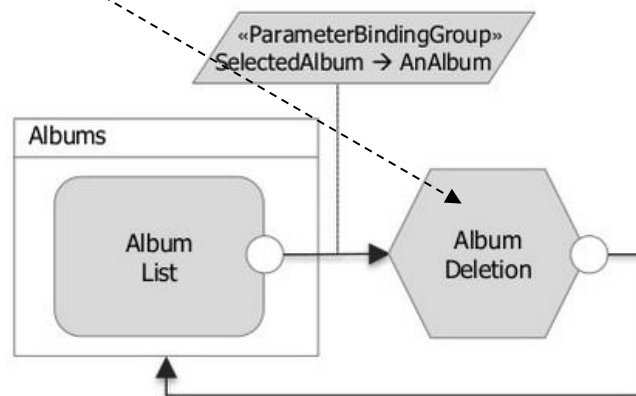


IFML Modelarea legării la logica business

Action

Logică business declanșată de un eveniment.

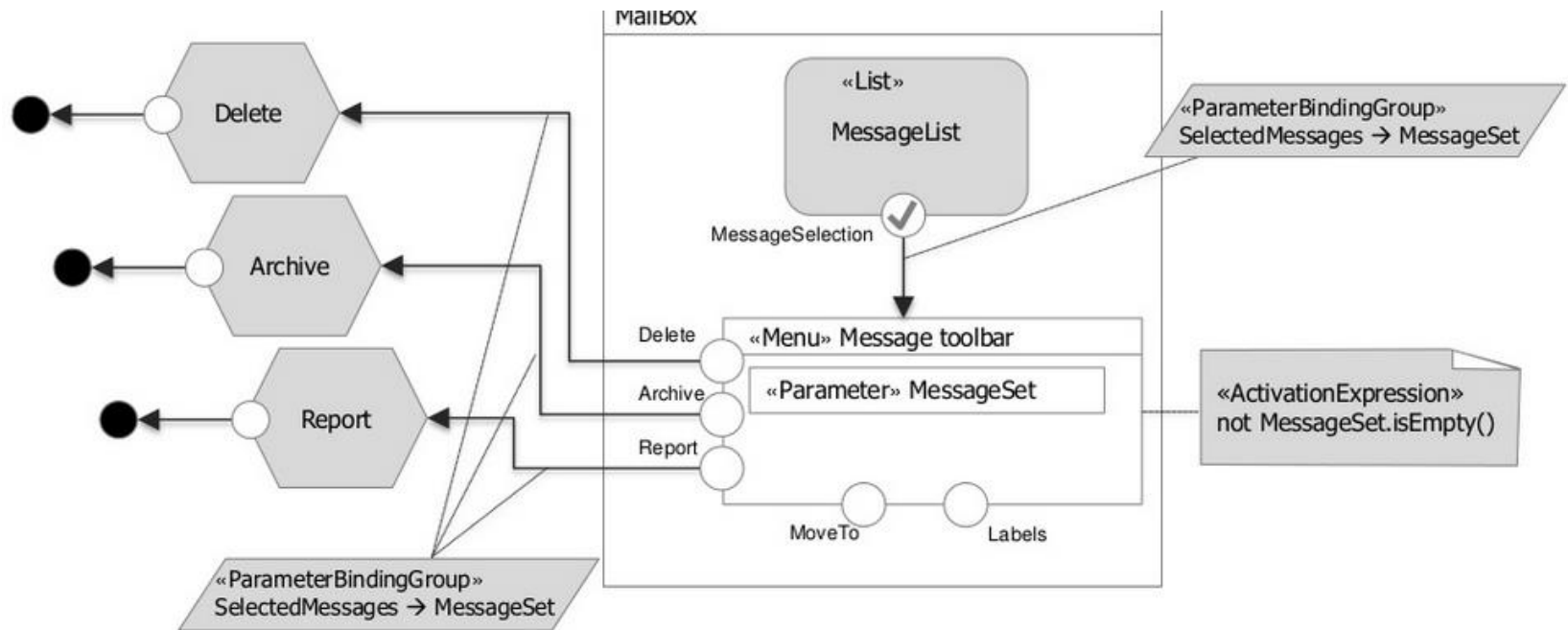
Implicit e situată de partea server, altfel e adnotată cu [Client]



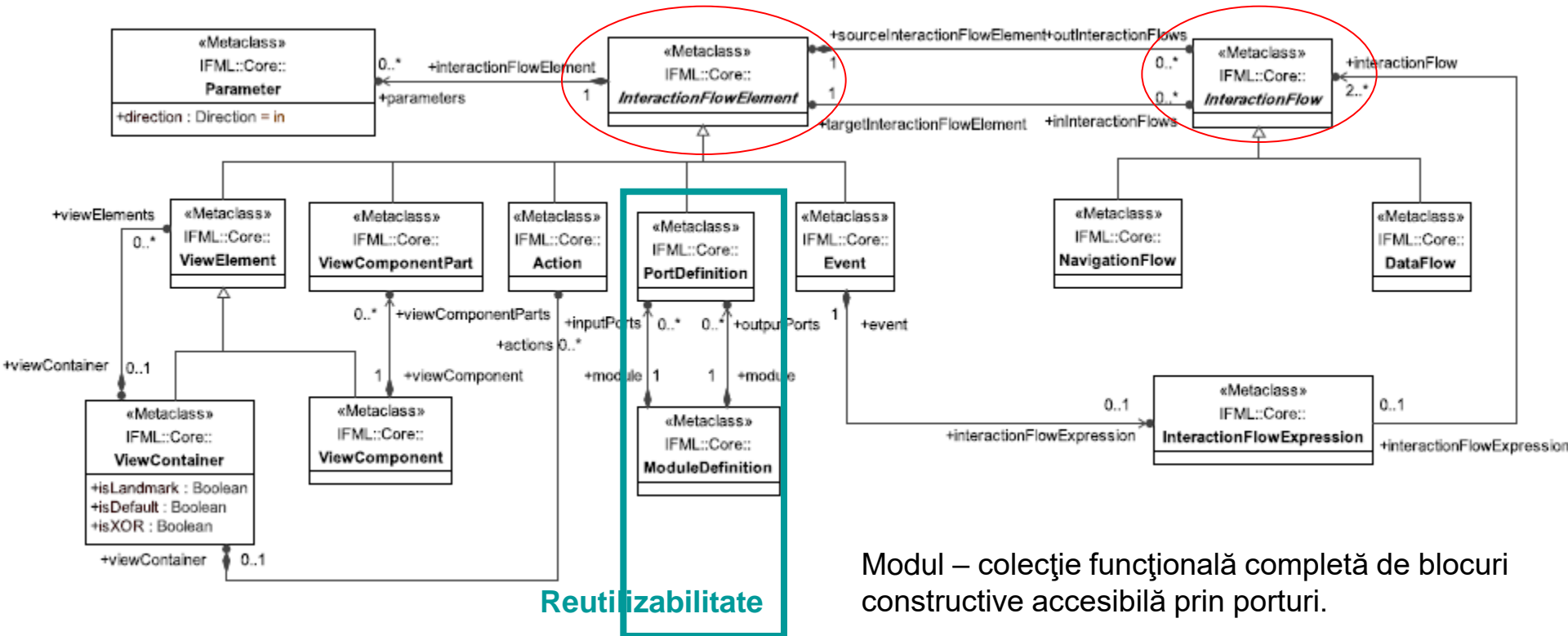
Logica business poate fi modelată cu metode din clase UML, diagrame de activitate, de secvențe, de stări și tranziții.

IFML - Modelare acțiuni

Exemplu



IFML – Reutilizabilitate



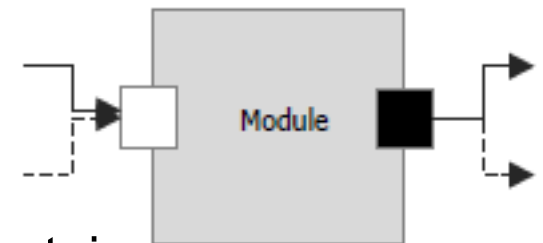
Modul – colecție funcțională completă de blocuri constructive accesibilă prin porturi.

Reutilizabilă sau înlocuibilă păstrând aceiași parametri de intrare și de ieșire.

IFML - Modul

Modul

Porțiune reutilizabilă din interfața utilizator și acțiunile corespunzătoare acesteia.

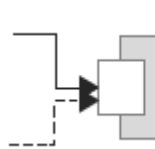


Port

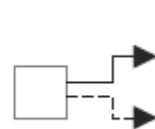
Punct de interacțiune între modul și contextul acestuia

intrare – colectează fluxurile și parametri ce intră în modul

în context

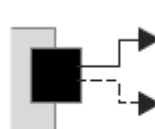


în modul



ieșire – colectează fluxurile și parametri ce ies din modul

în context

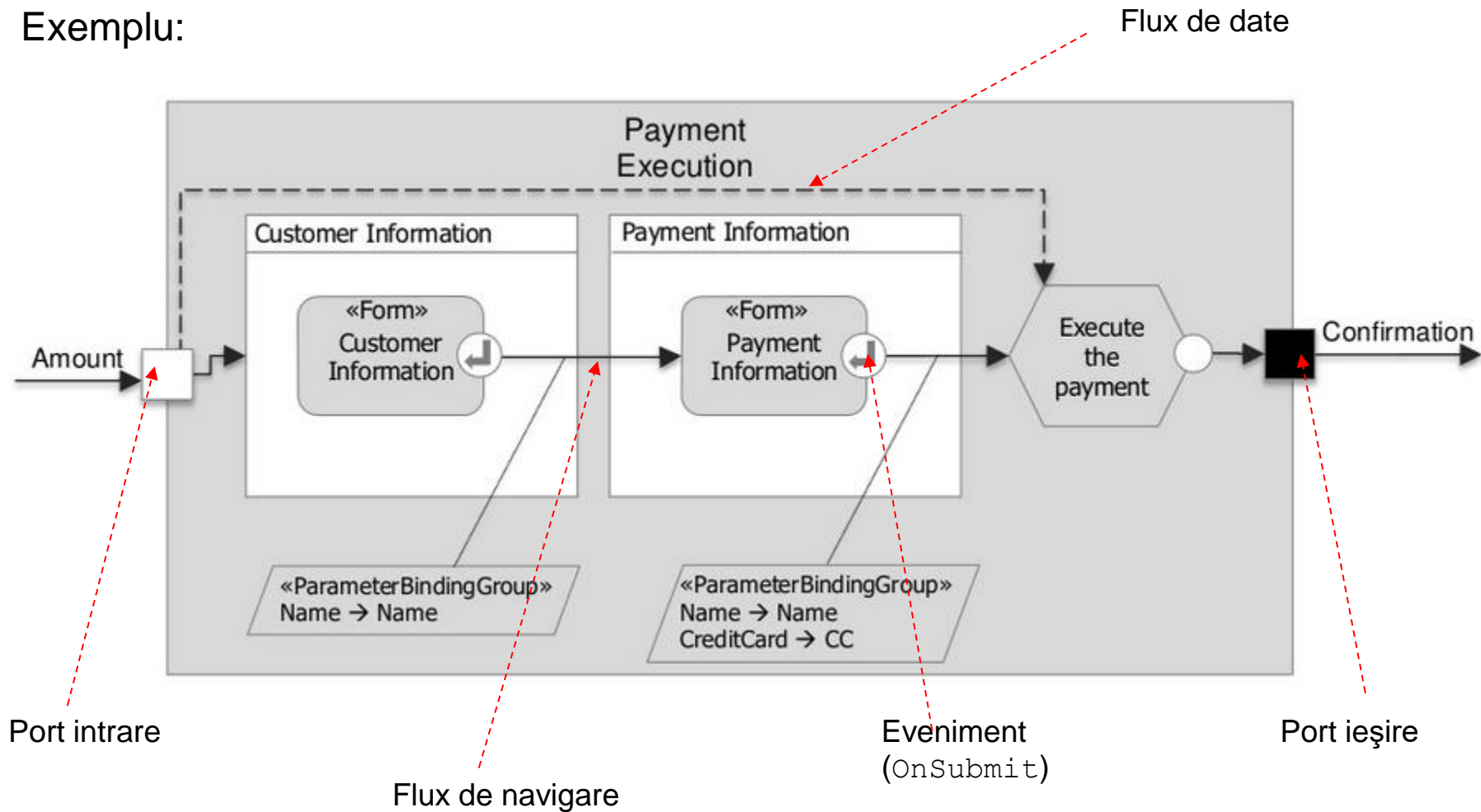


în modul



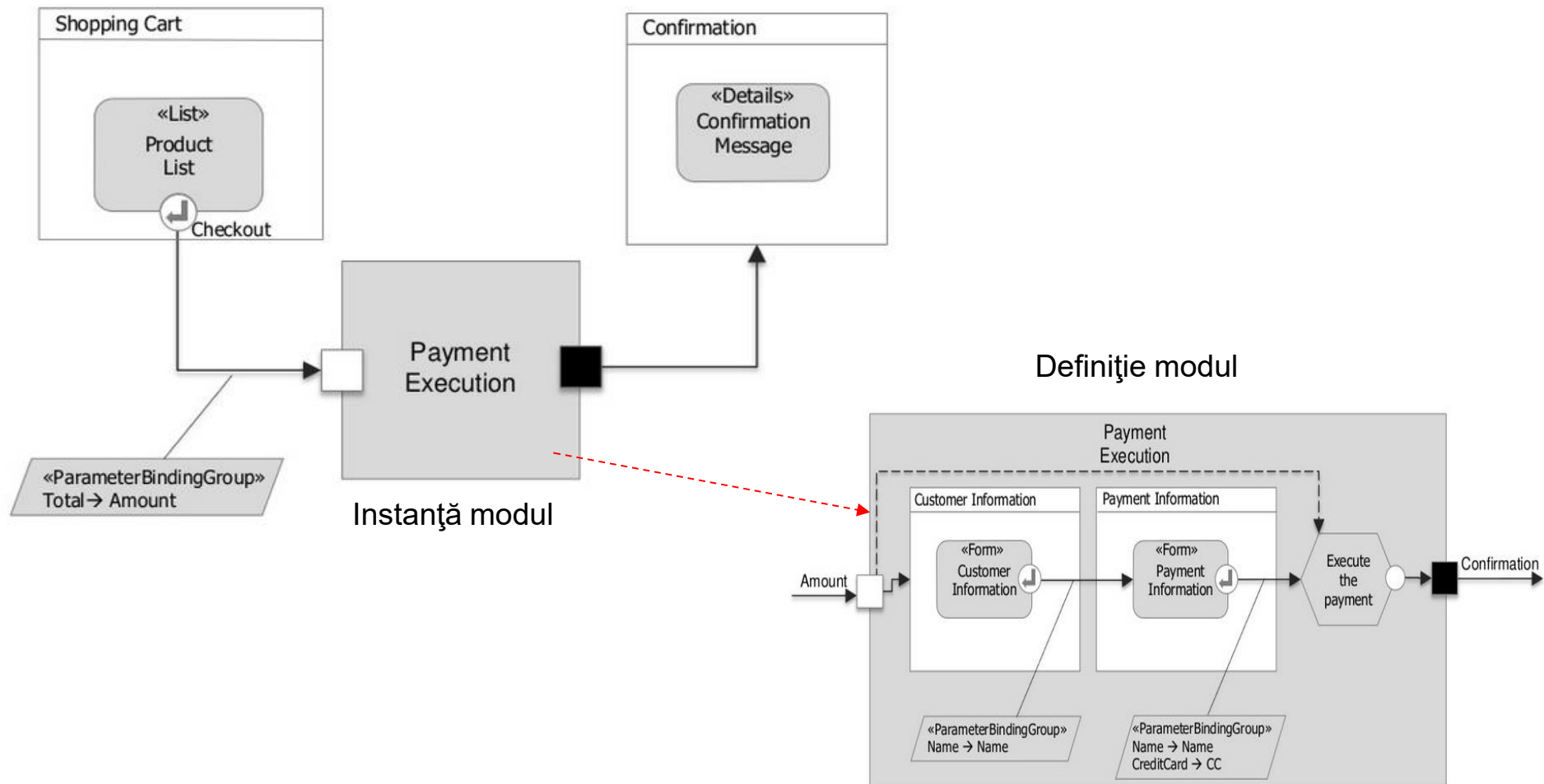
IFML – Definire modul

Exemplu:

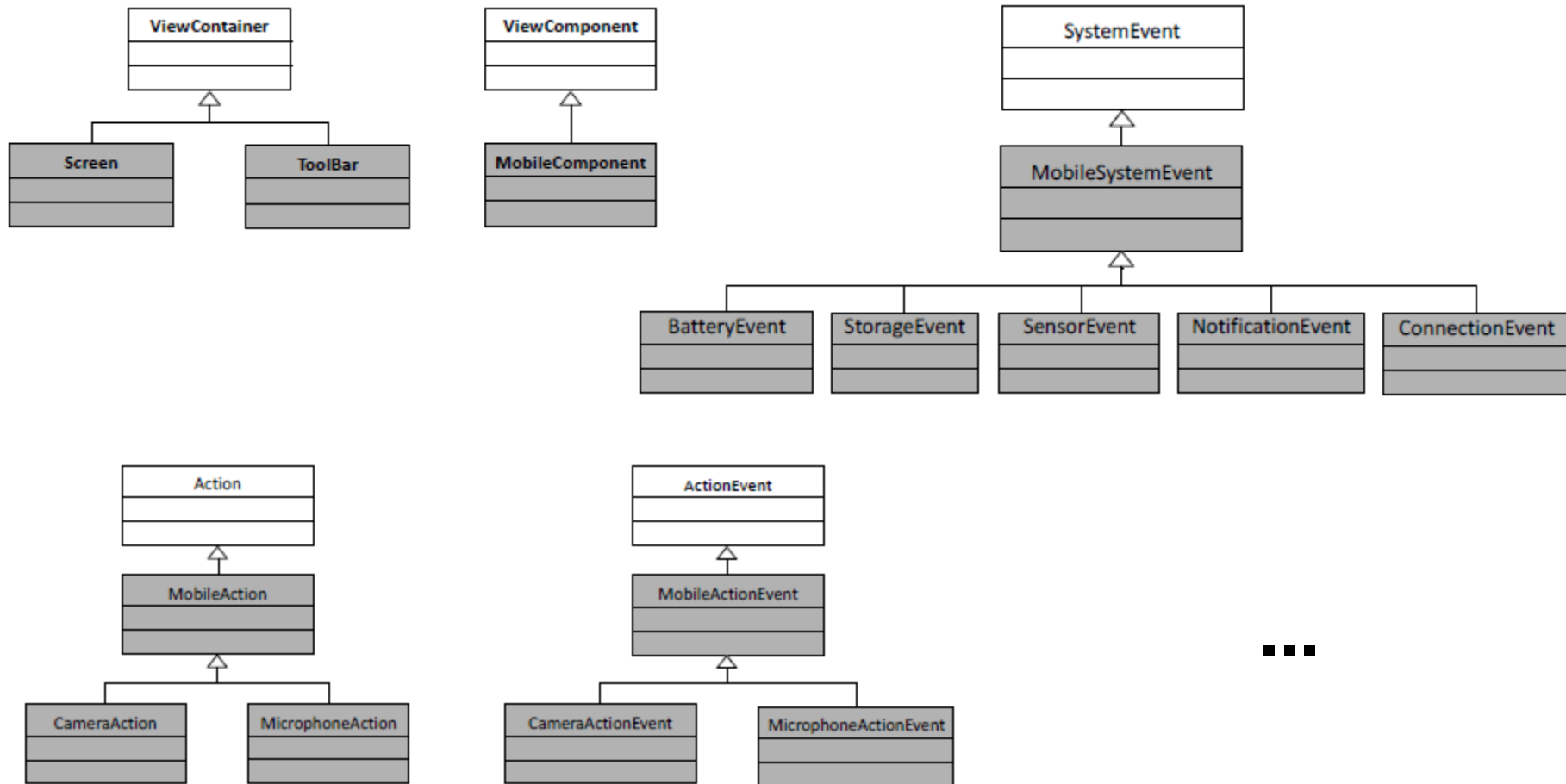


IFML – Utilizare modul

Exemplu:



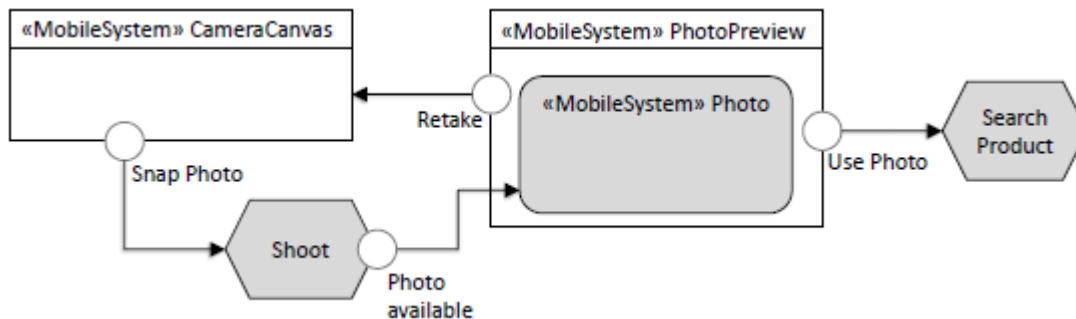
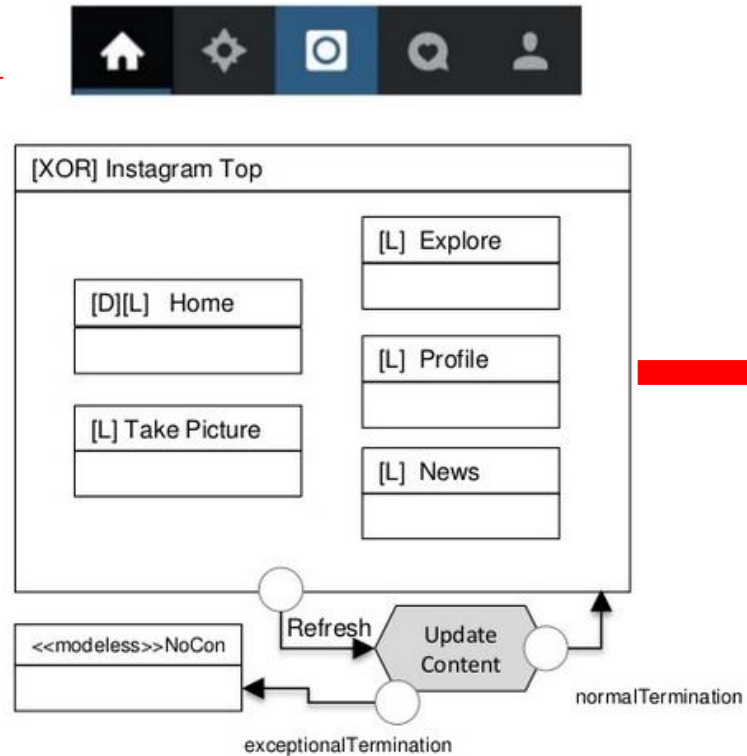
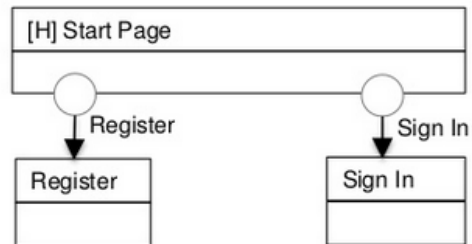
IFML – Modelare aplicații mobile



Brambilla, M., Mauri, A., Umuhoza, E., *Extending The Interaction Flow Modeling Language (IFML) for Model Driven Development of Mobile Applications Front End*, in Mobile Information Systems, 2014

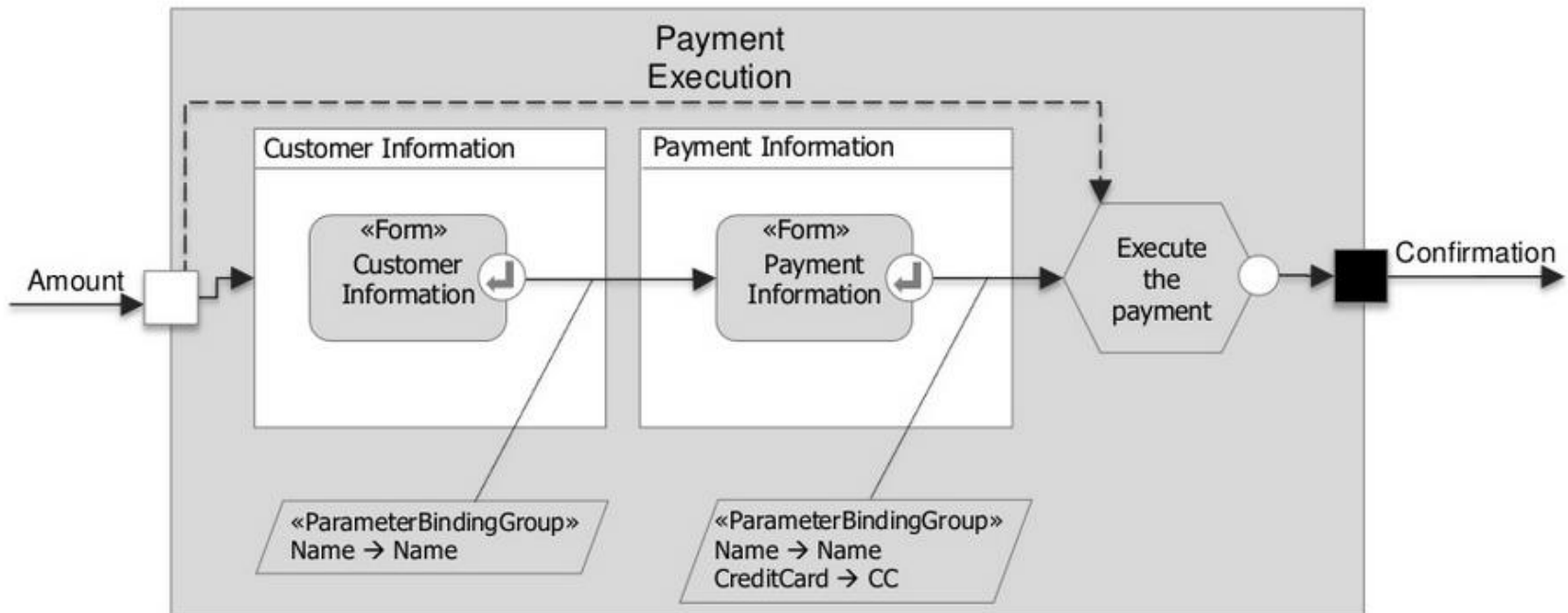
IFML – Modelare aplicații mobile

Exemple



Evaluare formativă

1. Care sunt categoriile de elemente modelate cu IFML ?
2. Care sunt parametrii de intrare ai acțiunii “Execute the payment”, de unde provin aceștia și pe ce tip de flux circulă fiecare ?



PLAN CURS

MDE (Model Driven Engineering) & MDWE (Model Driven Web Engineering)

IFML (Interaction Flow Modeling Language)

WebRatio Platform

Alte tehnologii de modelare și proiectare a aplicațiilor Web

WebRatio –platformă MDWE pentru site-uri web

Suport pentru:

- Modelare aplicații cu IFML și BPMN
- Creare aplicații Web și mobile - prin generare cod din model (model-driven)

Generare automată:

- baza de date relațională (pe baza definirii corespondenței dintre schema conceptuală a datelor și un set de surse de date)
- șabloane pentru pagini (pe baza definirii de foi de stil XSL și asocierii la pagini)
- cod optimizat la client și la server
- conformanță cu HTML5, CSS3 și Java/JSP 2.0+
- Instalare aplicații în web, pe dispozitive mobile, în cloud.
- Integrare cu instrumente pentru managementul ciclului de viață al aplicațiilor: resursele proiectului partajate pe un server de lucru colaborativ și pe un server de administrare versiuni; facilitează colaborarea între echipele de dezvoltare și de operare (DevOps).

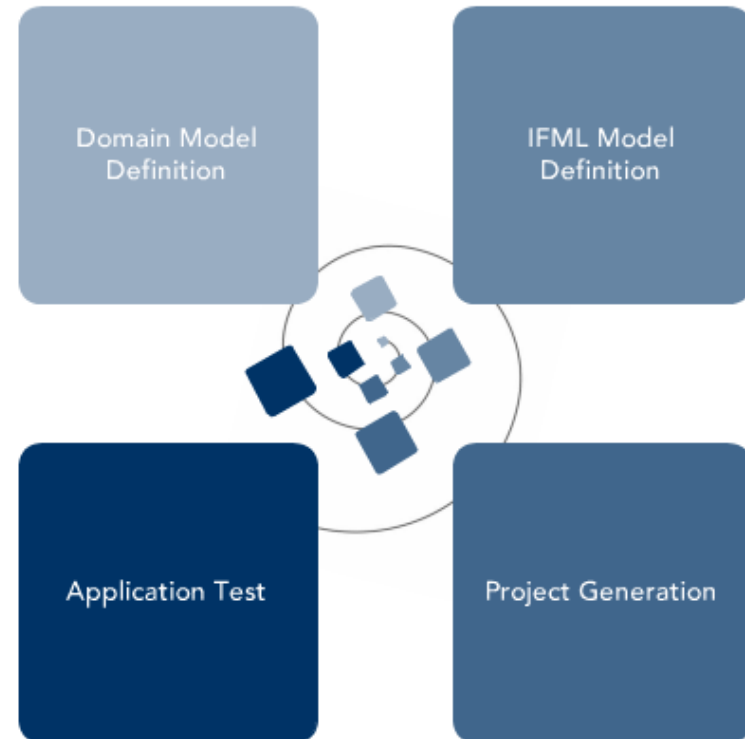
Domenii de utilizare:

- Prototipare rapidă în vederea validării cerințelor
- Creare aplicație executabilă pe arhitecturi standard (ex. JavaEE, Struts, .NET)
- Mentenanță
- Evoluție

WebRatio – procesul de dezvoltare

Fazele procesului ITERATIV de dezvoltare:

1. Definirea modelului domeniului (diagrame de clase sau diagrame ER).
2. Definirea modelului fluxurilor interacțiunilor (diagrame IFML pentru specificarea interacțiunii cu utilizatorul; încapsularea logicii business în *Action Definition*).
3. Generarea proiectului
4. Testarea aplicației



WebRatio – instrument CASE pentru IFML

Arhitectura aplicației generată din model pe bază de:

- șablon MVC,
- componente generice în treapta business
- reguli de prezentare CSS și XSL.

Descrieri interne:

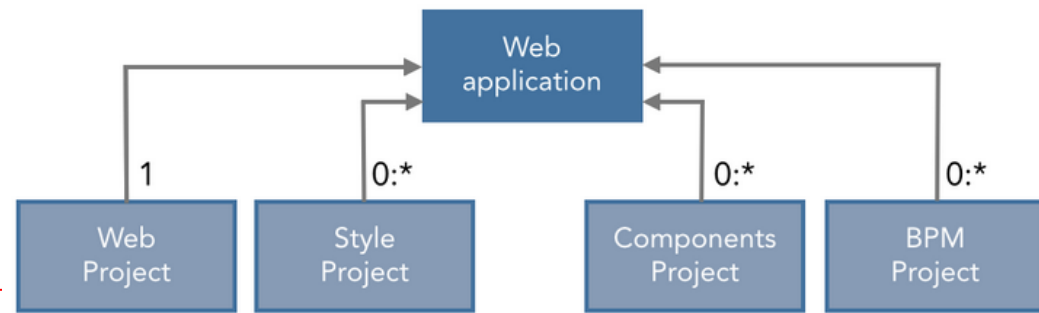
- XML – structuri date și hypertext
- XSL – grafică, layout, validare, elaborare documentație



Permite:

- Extindere limbaj cu unități definite de utilizator
- Extindere funcții WebRatio cu verificări definite de utilizator, generatoare de cod și de documentație, reguli de prezentare.

Tipuri de proiecte pentru aplicații web



Web Project – modelul aplicației (model, view și controller)

- Modelul domeniului
- Modelul Web - aspectele funcționale ale aplicației (pagini, operații, vederi, servicii, ...) definite cu IFML; organizat ierarhic.
- definițiile acțiunilor

Style Project – stilul prezentării (view)

- Șabloane de stil pentru elementele grafice ale aplicației și resurse utilizate de acestea (fișiere CSS, JavaScript, resurse media).

Components Project (controller)

- Setul componentelor definite de utilizator
- Setul regulilor de validare definite de utilizator
- Resursele necesare la runtime (ex. biblioteci externe)

BPM Project

- Procesele business automatizate folosind aplicația; posibil referite din Web Project.

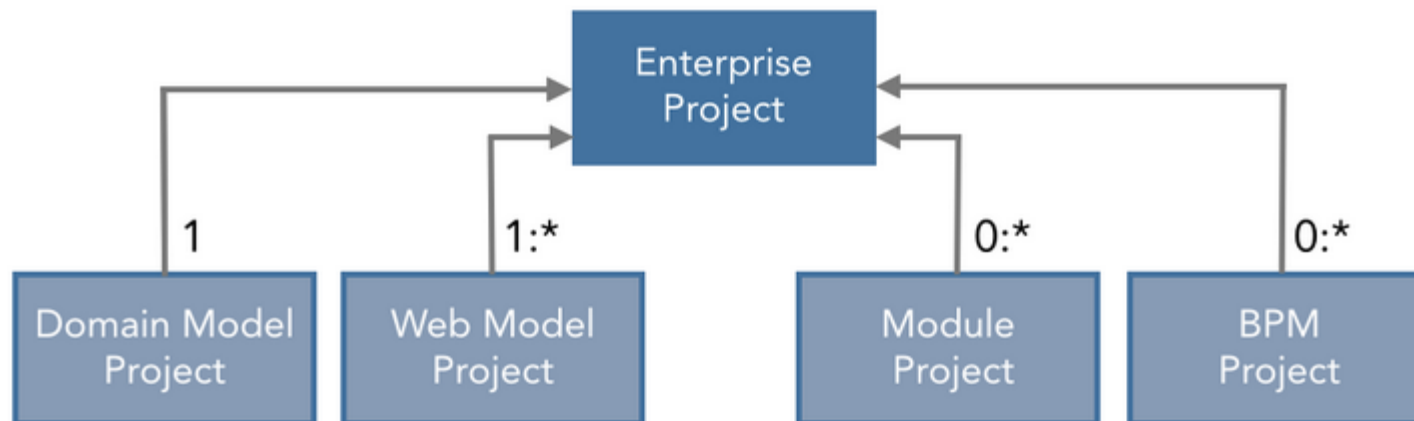
Tipuri de proiecte pentru aplicații enterprise

Enterprise Project – container de proiecte ce partajează același model de domeniu

Domain Model Project – modelul de domeniu, comun tuturor aplicațiilor

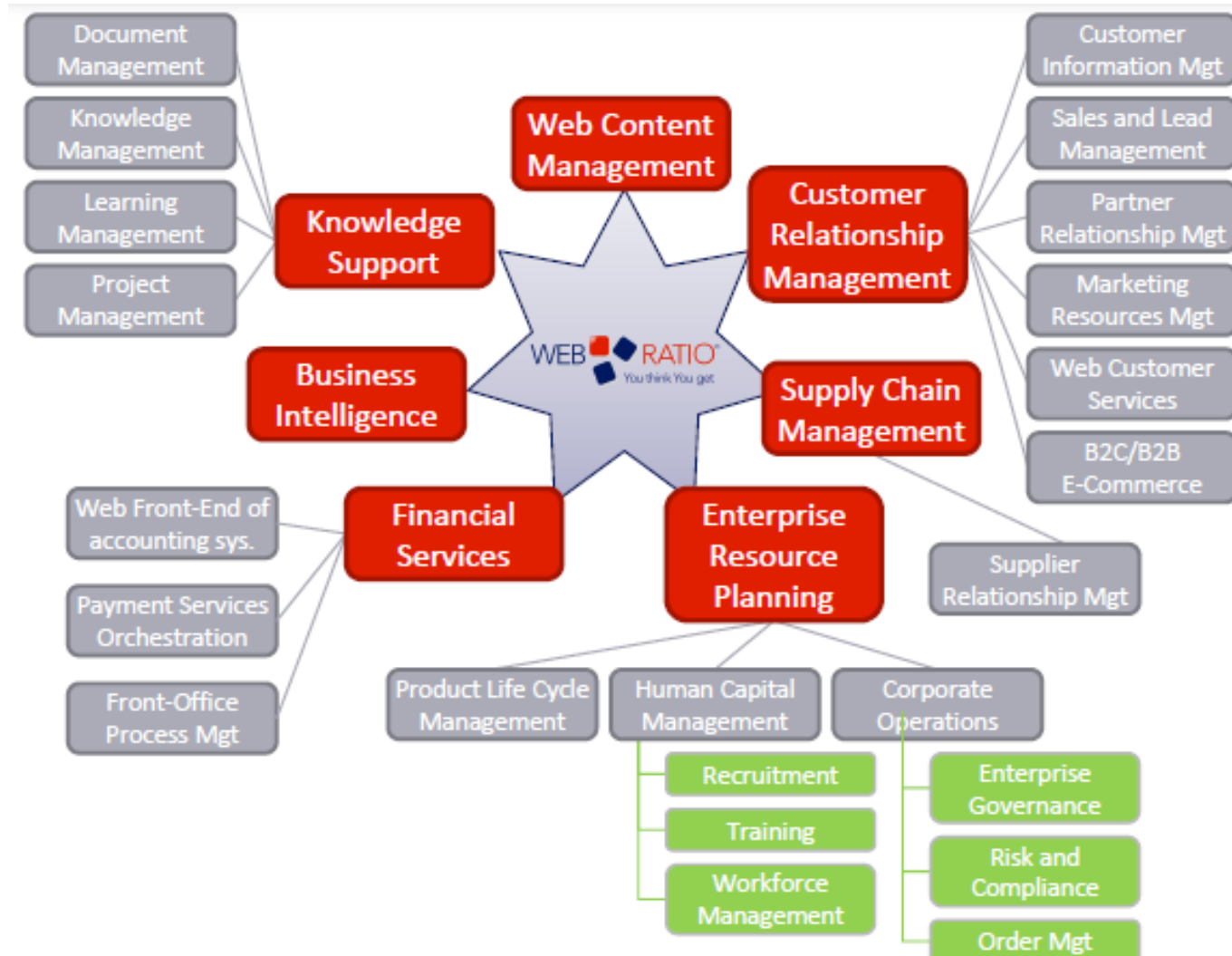
Web Model Project – Modelul Web al aplicației; vederile ce pot să apară în acest proiect : Site View, Service View, Module View.

Module Project – modulele partajate de toate aplicațiile.



WebRatio

Tipuri de aplicații



WebRatio – modelul domeniului

Structura datelor – schemă conceptuală ce cuprinde definiții de *concepte*, *proprietățile* lor, *relațiile* dintre acestea și *operațiile* aplicabile.

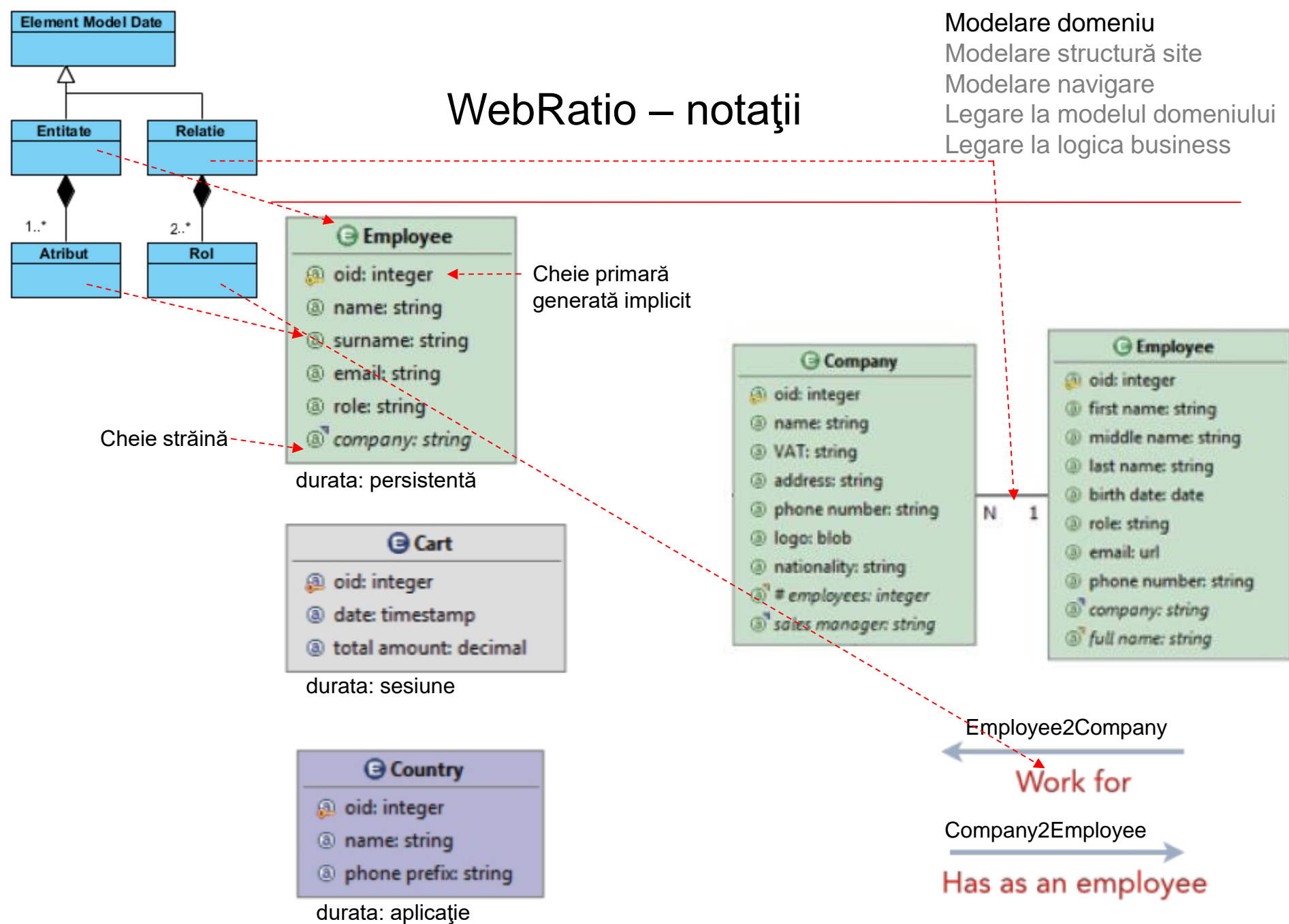
Notatii : diagrama Entity-Relationship, UML (diagrame de clase)

Elementele de modelare E-R:

- *Entitate* – container pentru date
- *Atribut* – descrie o proprietate a unei entități; are tip; categorii: simple, multi-valoare, structurate.
- *Relație* – asociere semantică între entități; are roluri; are multiplicități (cardinalități); poate avea atribute; este N-ară.
- *Ierarhie* generalizare / specializare (ierarhie IS-A)
- *Expresii* OCL sau OQL pentru specificare atribute sau relații derivate (calculate din valorile altor elemente ale schemei datelor).

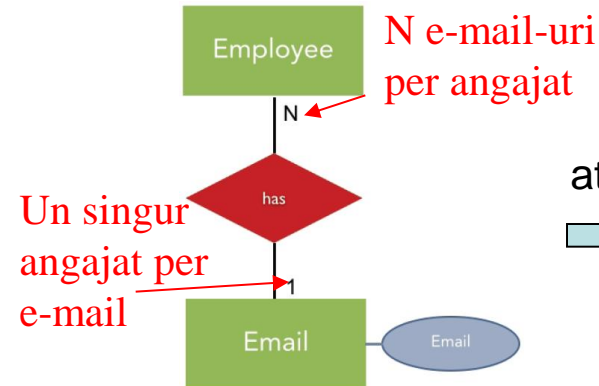
Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business

WebRatio – notații

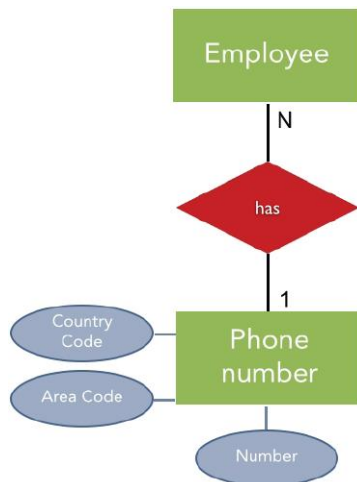
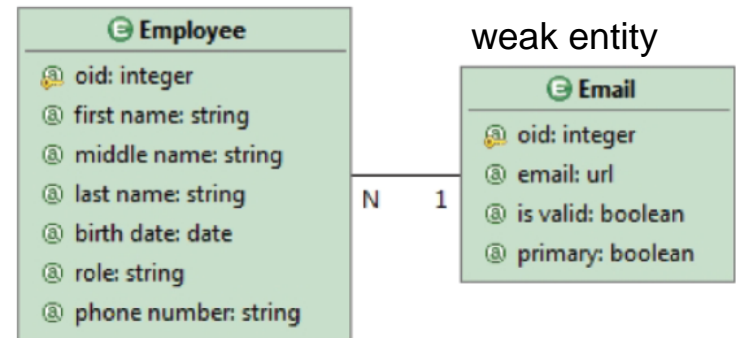


Normalizare schemă BD

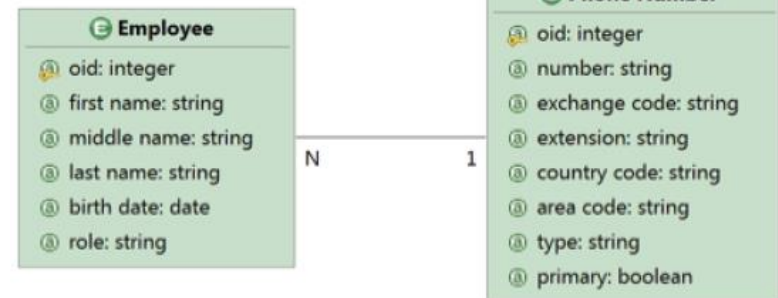
Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business



atribut multivaloare

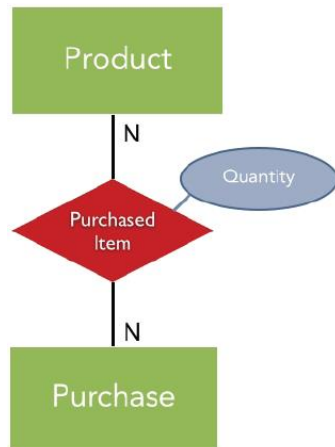


atribut structurat

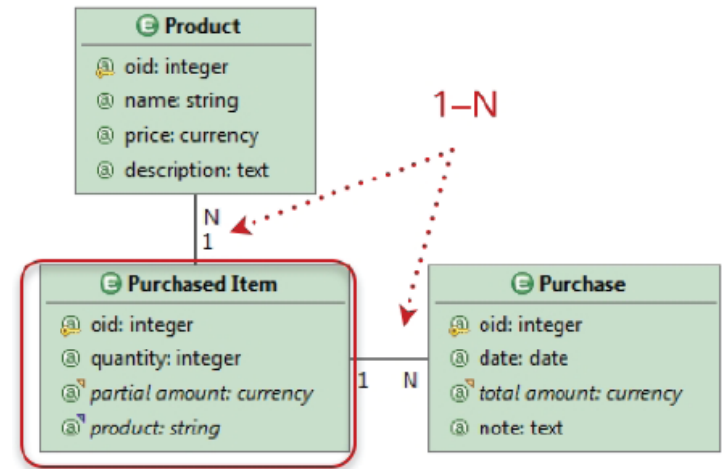


Normalizare schemă BD

Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business

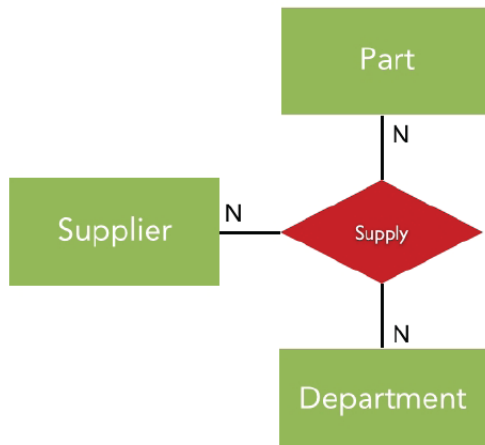


relație cu atribute



weak entity

weak entity



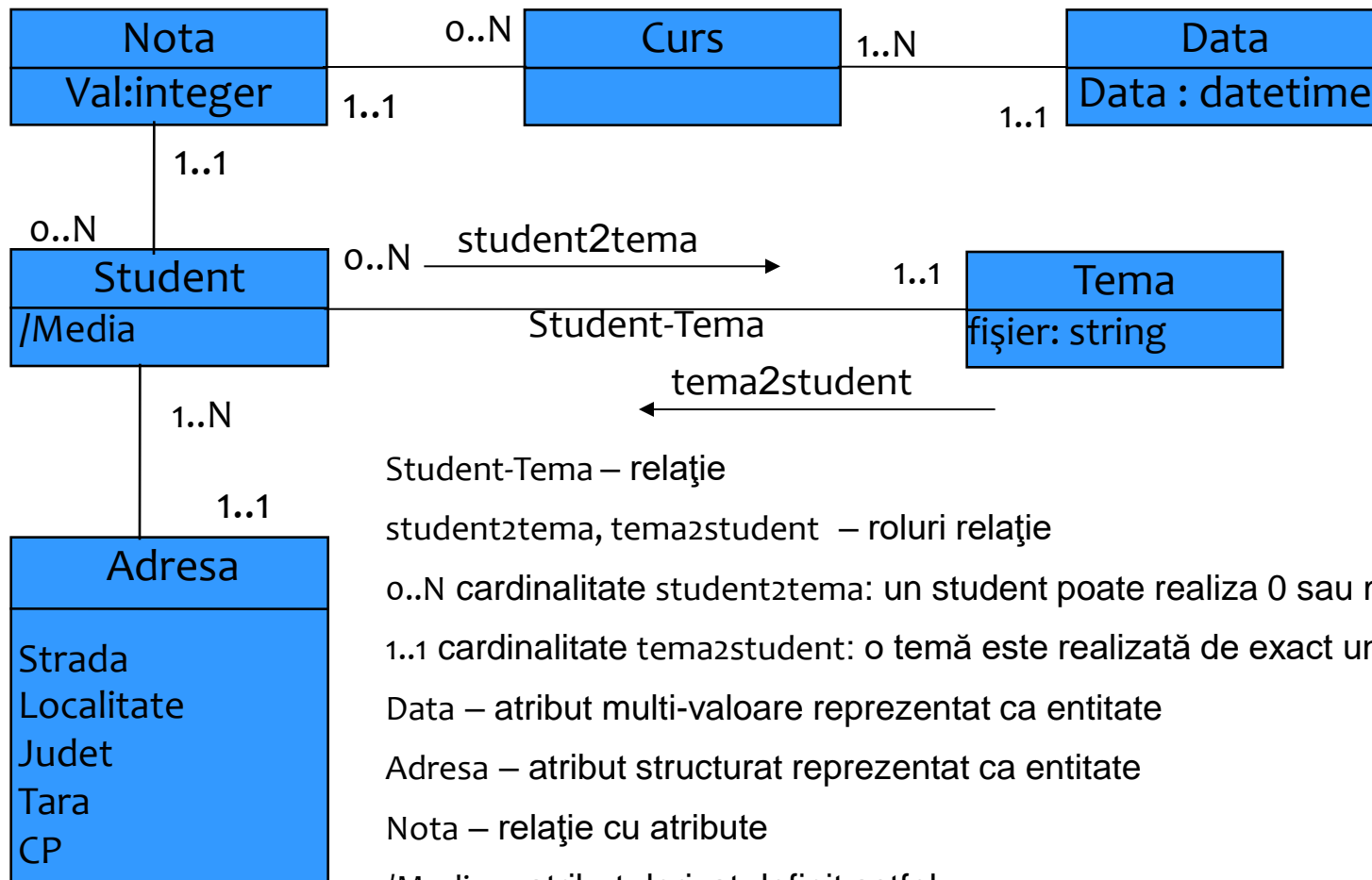
relație N-ară



Transformare relație în
entitate și conectarea ei prin
N relații binare.

Model domeniu - exemplu

Extras din model domeniu:



Student-Tema – relație

student2tema, tema2student – roluri relație

o..N cardinalitate student2tema: un student poate realiza 0 sau mai multe teme

1..1 cardinalitate tema2student: o temă este realizată de exact un student

Data – atribut multi-valoare reprezentat ca entitate

Adresa – atribut structurat reprezentat ca entitate

Nota – relație cu attribute

/Media – atribut derivat definit astfel:

`/Media = Sum(Self.Student2Nota.Val) / Count(Self.Student2Nota)`

WebRatio – sincronizarea modelului domeniului cu stratul de persistență

Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business

1. Conectare la baza de date
2. Sincronizarea modelului domeniului cu obiectele din baza de date (operația Synchronization)
 - forward engineering – creare BD din modelul domeniului
 - reverse engineering – crearea modelului domeniului din tabelele BD

Pentru stratul de persistență WebRatio utilizează Hibernate.



Se generează un set de fișiere ce pot fi folosite de Hibernate pentru a administra entitățile.

i.e. Pentru fiecare entitate din modelul domeniului se generează un fișier XML (mapare Hibernate) și o clasă Java.

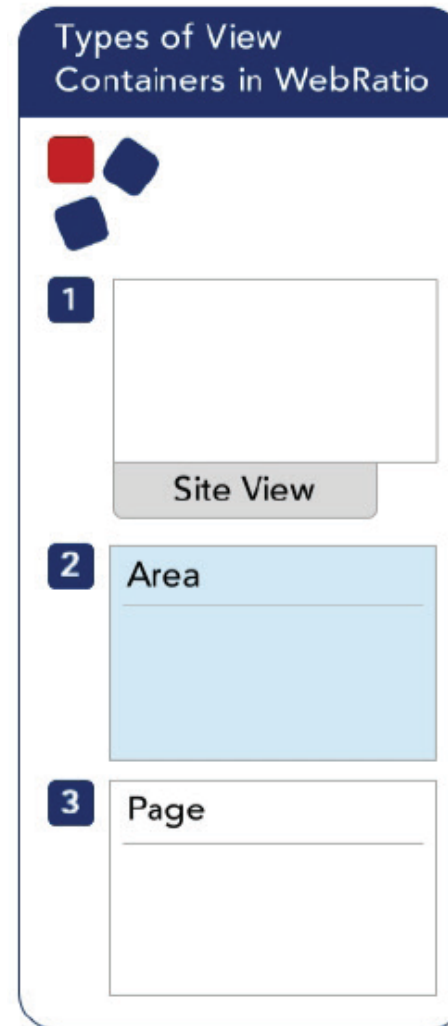
WebRatio – modelare structură site

IFML ViewContainer – element de interfață ce include elemente care afișează conținut sau oferă suport pentru interacțiuni.

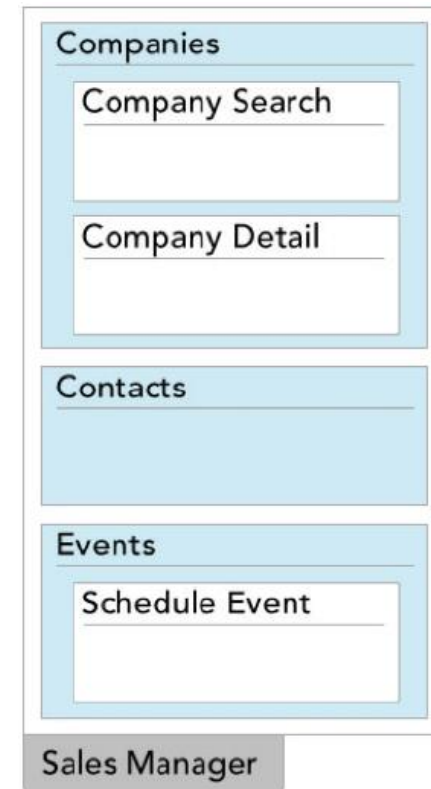
Agregat de *ViewContainer*, *ViewComponent* și *Action*.

Subtipuri în WebRatio:

- **Site View** – porțiune autoconținută a unui model al fluxurilor interacțiunilor care suportă cazurile de utilizare ale unui anumit actor; corespunde unei anumite secțiuni a aplicației Web, adresabilă cu URL.
- **Area** – compusă din pagini și arii incluse și destinată pentru publicare sau administrare de conținut omogen.
- **Page** – prezintă utilizatorului componente view.



Exemplu



WebRatio – Proprietăți

Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business

Valabile pentru containere și acțiuni:

Home – oferite implicit la lansarea aplicației.

Landmark – vizibilitate globală la nivel de area sau site view.

Default – oferite implicit la accesarea ariei ce le include.

Protected – accesibilitate pe bază de autentificare.

Secure – furnizare prin conexiune securizată;
proprietatea este moștenită de fiecare element
conținut de obiectul ce are această proprietate.

	Home
	Landmark
	Default
	Protected
	Secure
	Localized

Valabilă pentru containere:

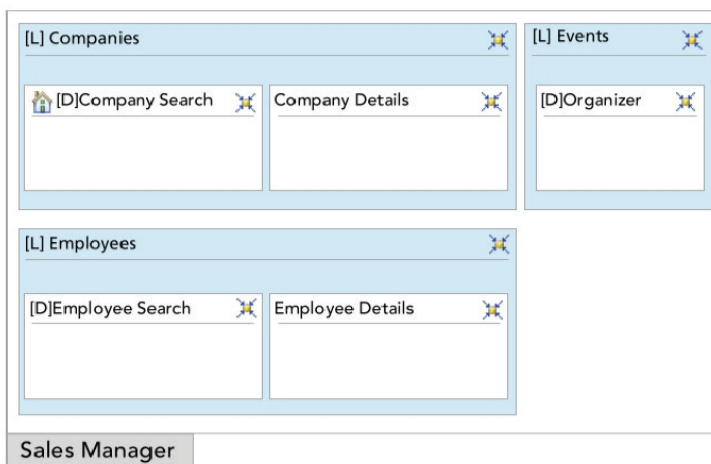
Localized – conținutul poate fi văzut în diferite
limbi, funcție de localizarea utilizatorului.

WebRatio – Exemplu

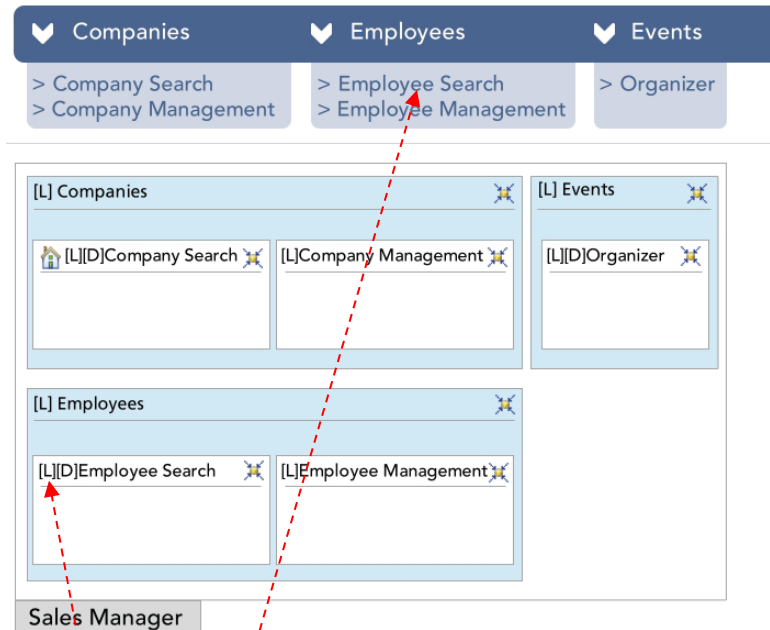
Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business

How it
looks on
the web

Companies | Employees | Events



(a)



(b)

[L] → *Landmark* – vizibilitate globală la nivel de area sau site view.

WebRatio – View Component

ViewComponent :

element de interfață care afișează conținut
și/sau acceptă intrări.

ViewComponentPart – element din cadrul unei *ViewComponent*,

poate

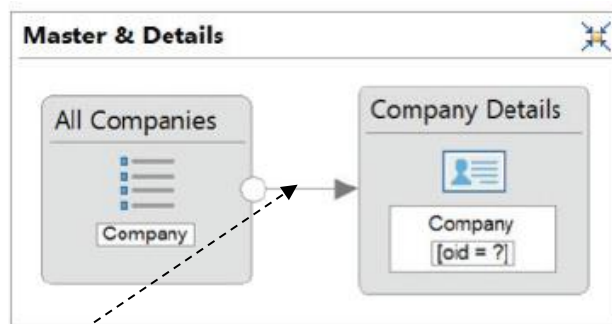
- genera evenimente
- avea fluxuri de intrare și de ieșire
- conține alte *ViewComponentPart*



WebRatio – Modelare navigare

Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business

Exemplu : aplicarea șablonului Master&Details



←..... IFML model in WebRatio

Navigation flow

Transferă implicit ID-ul
obiectului selectat.

Generated
Web Application Page→

Company Master And Details

Companies List

logo	VAT	name	nationality	
Logo.png	AAA000	WebRatio s.r.l.	Italian	Details
	YYY000	Yellow Company	American	Details

Company Details

address	Piazzale Cadorna 10
name	WebRatio s.r.l.
logo	Logo.png
VAT	AAA000
phone number	0123456789
nationality	Italian

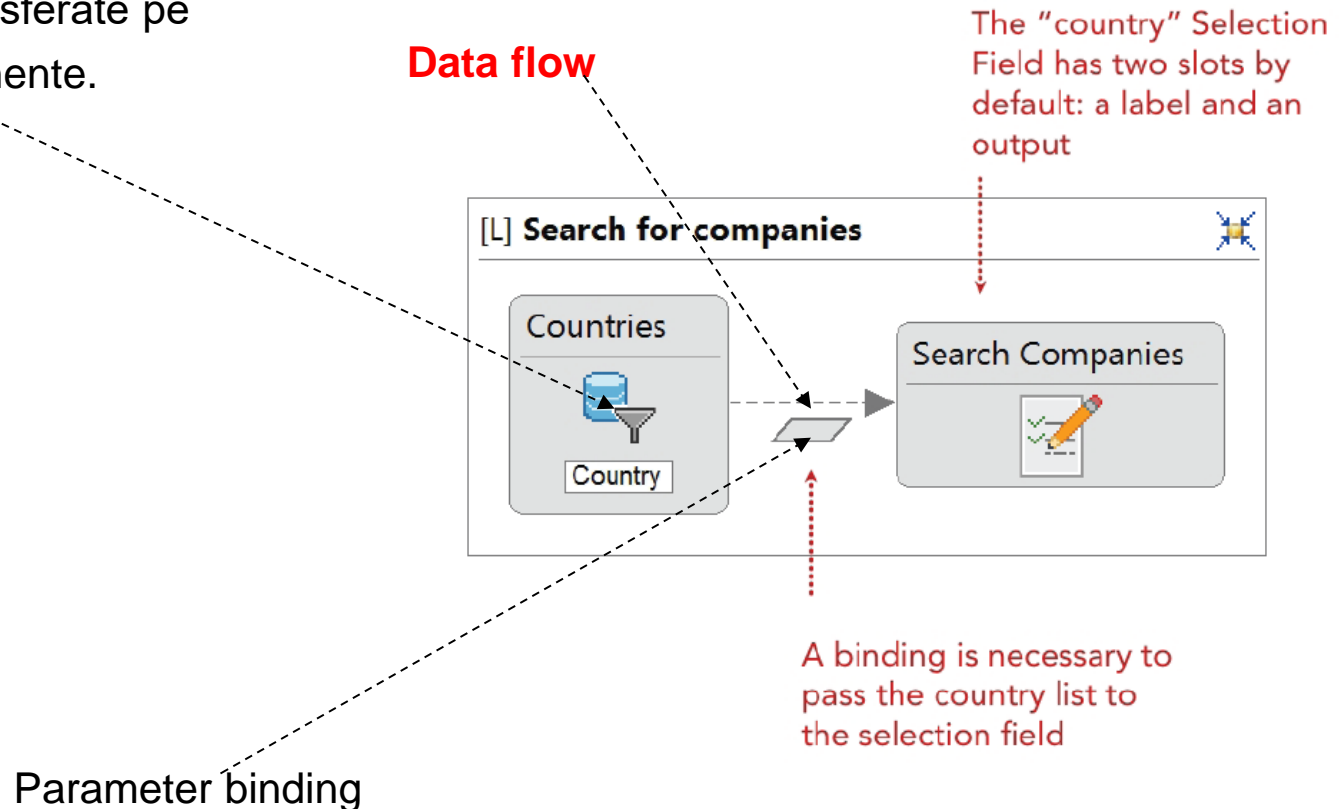
Variantе ale șablonului:

- selecție implicită
- selecție detalii pe mai multe nivele

WebRatio – modelare navigare

Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business

Componentă selector – extrage, conform definițiilor din DataBinding, informații din modelul domeniului fără a le afișa, pentru a fi transferate pe flux de date altei componente.



WEbRatio –

Modelarea legării la modelul domeniului

Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business

Modelarea legării la stratul de persistență



Data Binding

- ✓ Entity
- ✓ Display Attribute
- ✓ Conditional Expression
 - **Key** Condition
 - **Attributes** Condition
 - **Relationship Role** Condition

Proprietățile conexiunii cu elemente din modelul domeniului

- referință la instanță din modelul domeniului
- attributele afișate ale instanței
- predicat pentru selecția instanțelor
 - condiție pe cheia primară
 - condiție pe valoare atribut
 - condiție pe rol într-o relație (cheie străină)

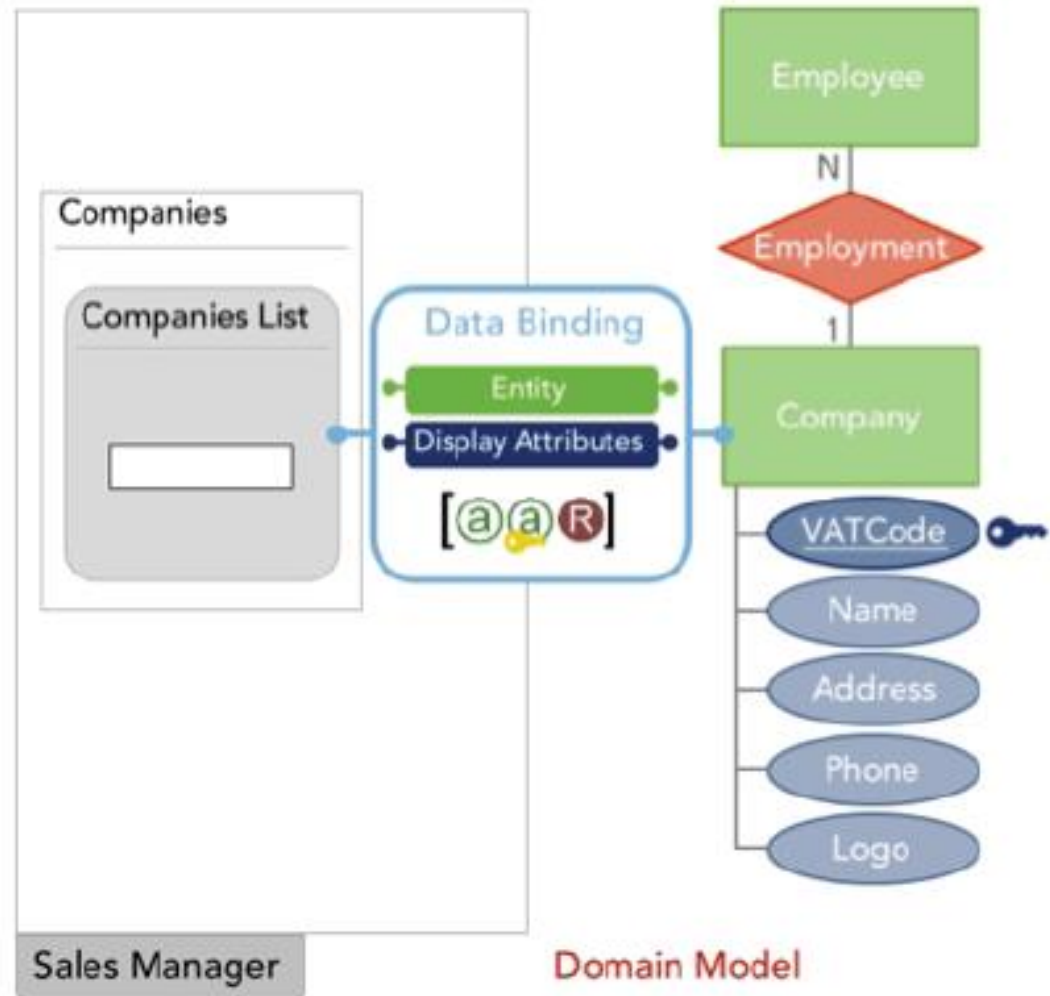
WebRatio – Modelarea legării la modelul domeniului

Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business



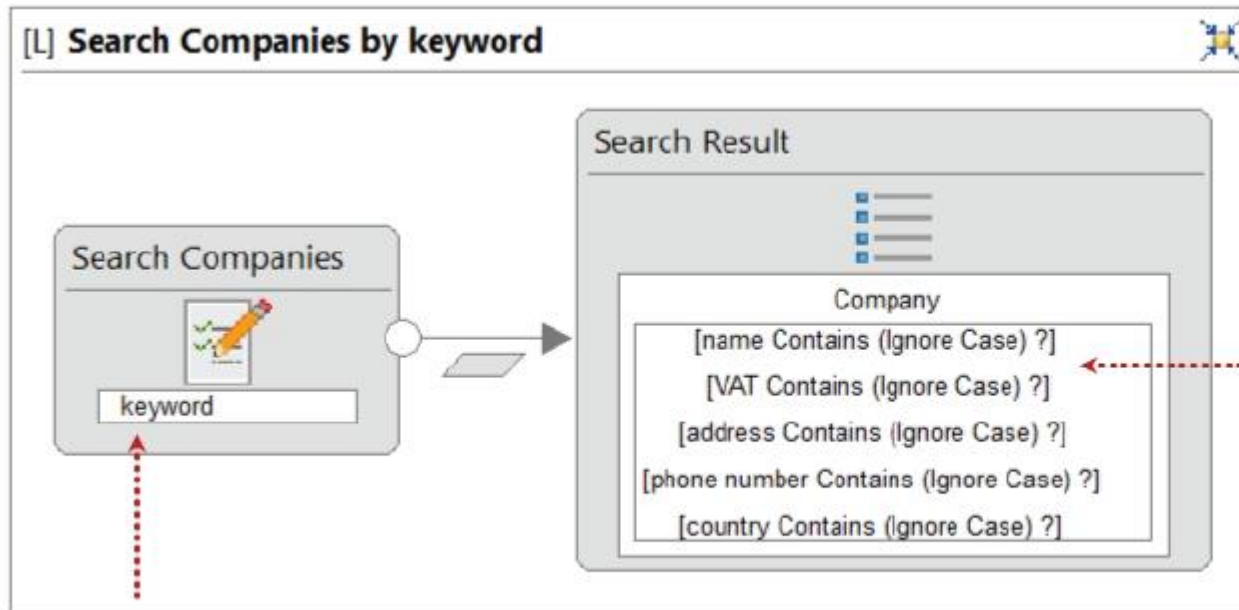
Data Binding

- ✓ Entity
- ✓ Display Attribute
- ✓ Conditional Expression
 - **Key** Condition
 - **Attributes** Condition
 - **Relationship Role** Condition



WebRatio – Exemplu

Exemplu conectare la stratul de persistență



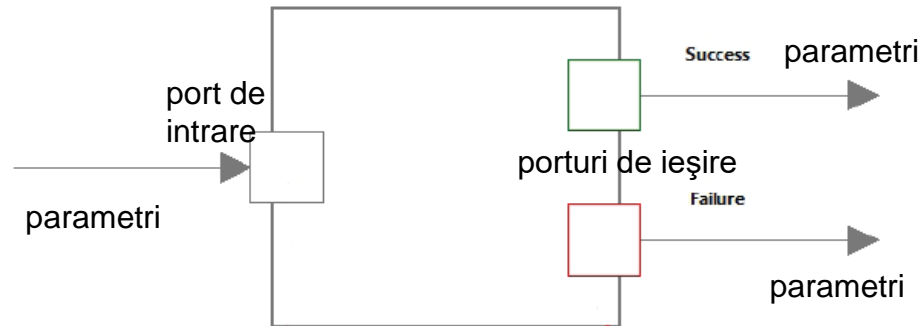
Unique search field

The Conditional Expression is built with the **OR** operator between conditions

WebRatio – Acțiuni

Action Definition:

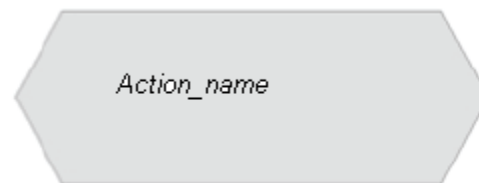
- logică business declanșată de un eveniment și definită cu **operații**
- referită de o acțiune din modelul IFML
- modelată în Module Definitions View



Action – instanțiere a unei Action Definition

Poate fi plasată

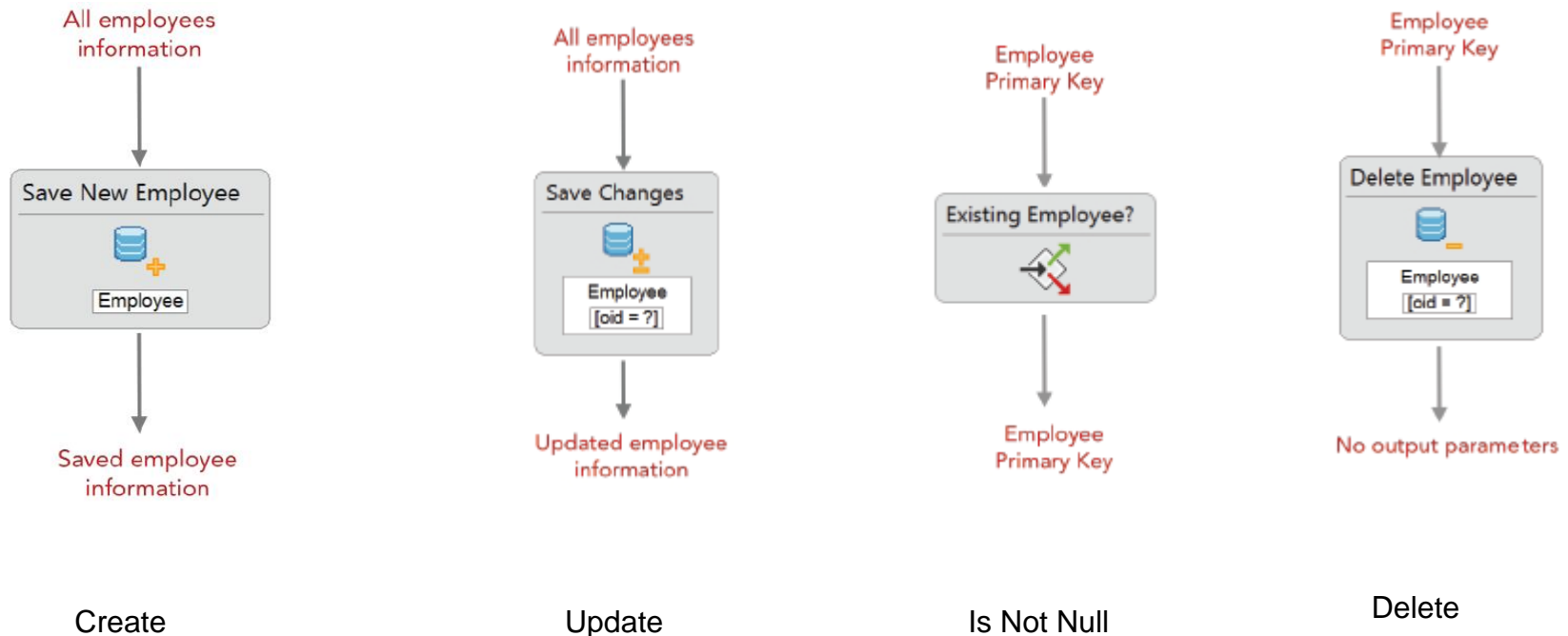
- în modelul IFML în afara paginilor
- în cadrul unei Action Definition.



WebRatio – Operații predefinite

Operație – componentă ce implementează un comportament din logica business.

Exemple operații predefinite în WebML:



WebRatio – Fluxuri de ieşire

Tipuri de flux de navigare de la acţiune/operaţie:

- OK Flow – realizare cu succes a operaţiei/acţiunii



- KO Flow – eşuarea operaţiei/acţiunii



Obs. Conectări permise

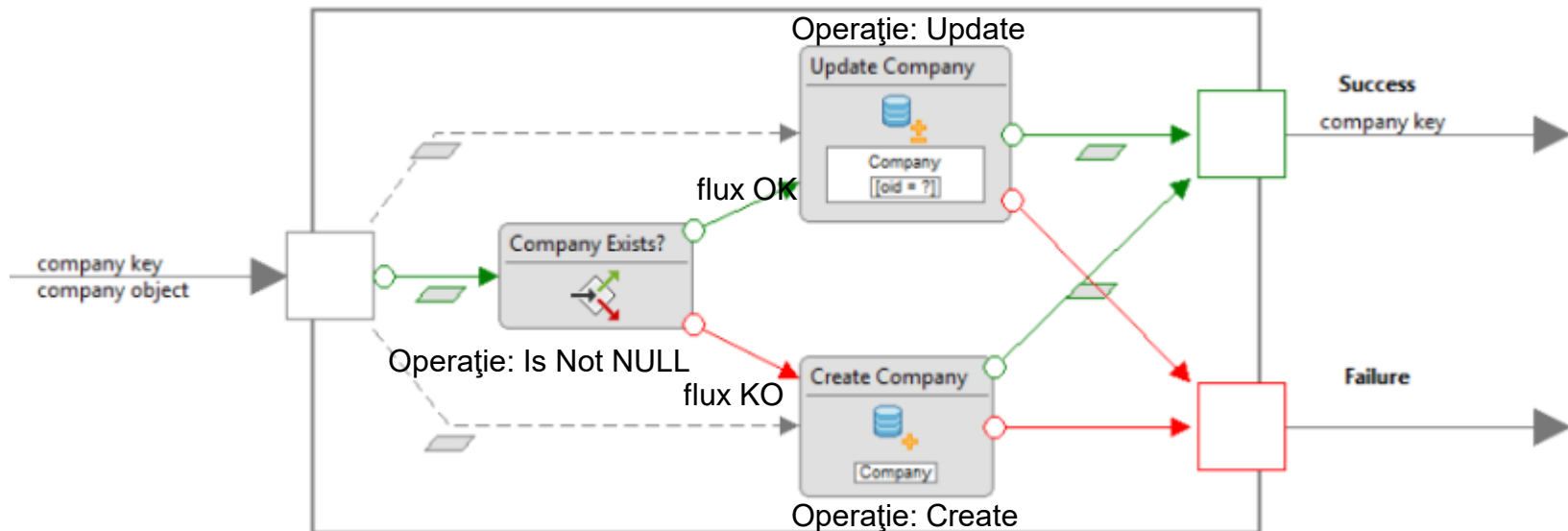
- de la operaţie la operaţie sau la port de ieşire
- de la acţiune la orice element IFML

WebRatio – Definire modul

Module Definition – container cu porțiuni reutilizabile de model de flux de interacțiuni și de logică business.

(Definire de module sau/și de acțiuni).

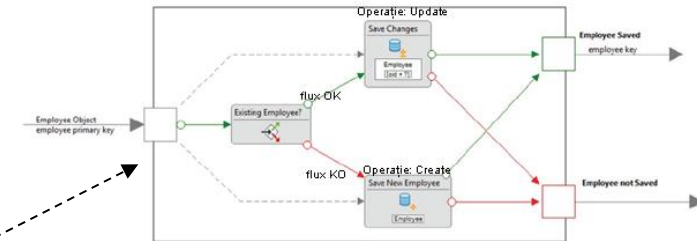
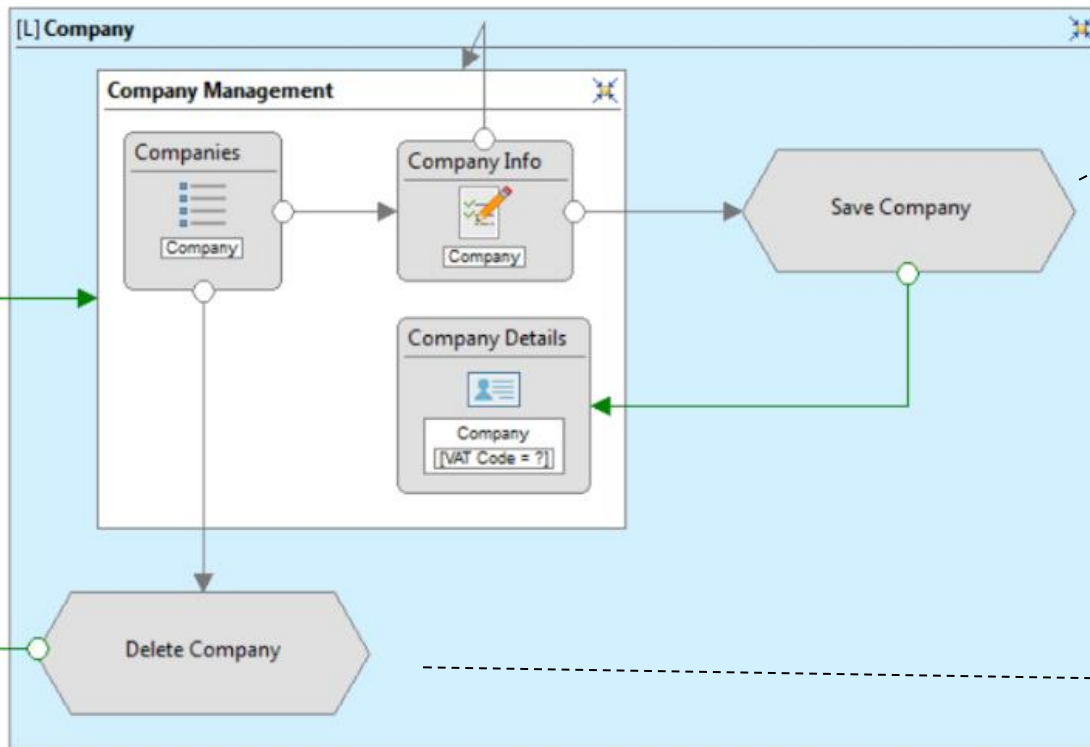
Exemplu: Set de operații definite în cadrul unei Action Definition și legate printr-un set de fluxuri OK și KO. Definiția acțiunii “Save Company”.



WebRatio – Utilizare modul

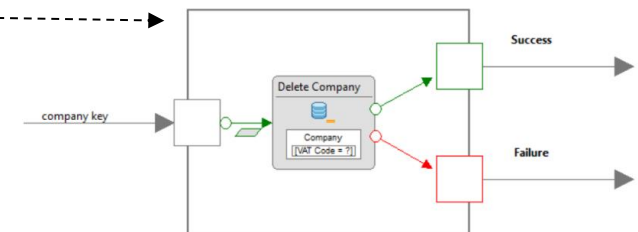
Modelare domeniu
Modelare structură site
Modelare navigare
Legare la modelul domeniului
Legare la logica business

Exemplu: Șablon pentru administrare conținut (operații CRUD).



acțiune definită în modul
(sau folosind UML: metode din clase,
diagrame activitate, de secvențe, de stări și
tranziții)

acțiune definită în modul



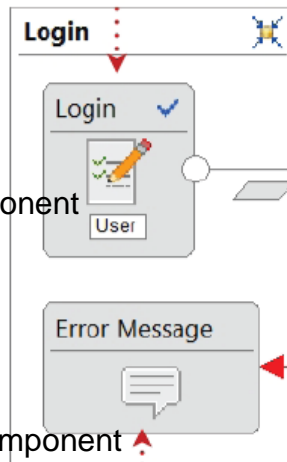
WebRatio – control acces și personalizare

Login – șablon pentru garantarea accesului utilizatorului la resurse protejate ale aplicației.

Model IFML : The form usually has three fields:

- username
- password
- remember me

Page View Container



Form View Component

Message View Component

The message shown when the login fails

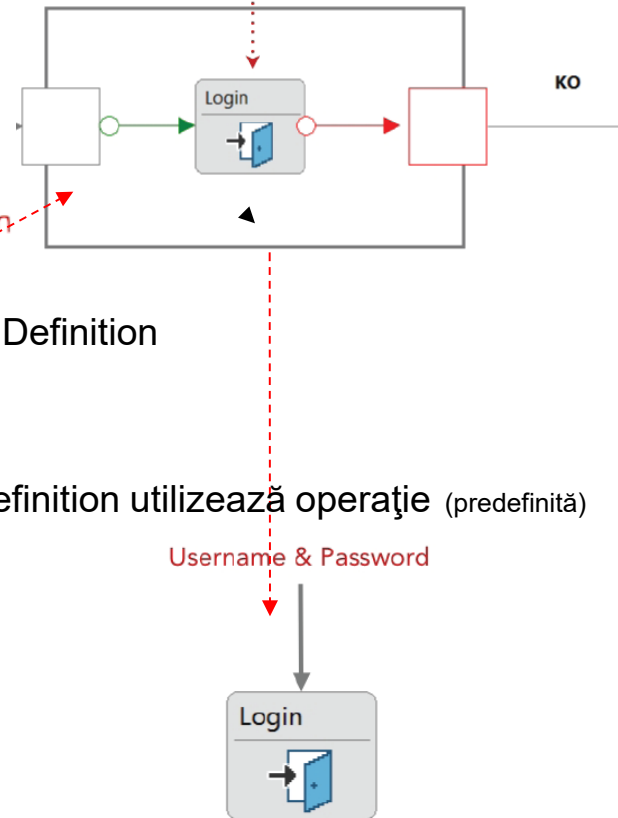
The action performing the login

Action referă Action Definition

This is the operation that performs the user login

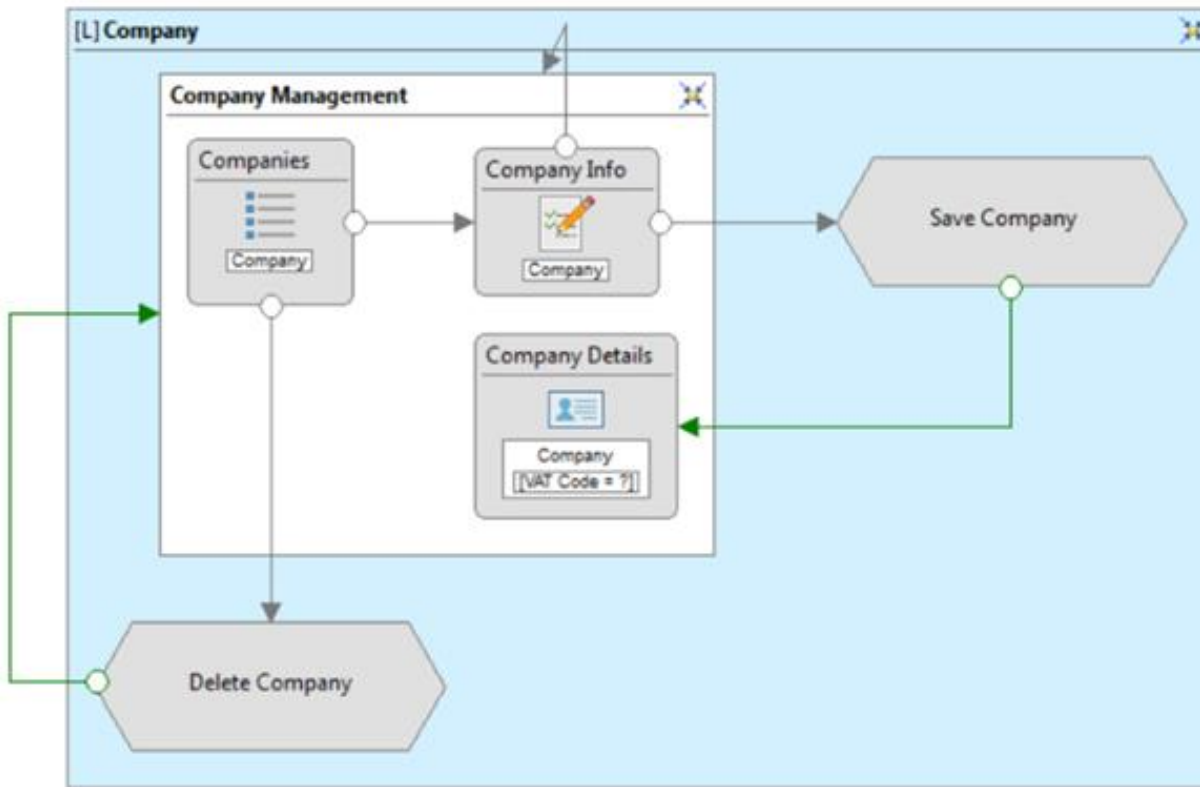
Action Definition utilizează operație (predefinită)

Username & Password



Evaluare formativă

1. Ce elemente lipsesc din acest extras dintr-un model IFML ?



<https://forms.gle/d7PGJyZ8QmT6nK777>

PLAN CURS

MDE (Model Driven Engineering) & MDWE (Model Driven Web Engineering)

IFML (Interaction Flow Modeling Language)

WebRatio Platform

Alte tehnologii de modelare și proiectare a aplicațiilor Web

UWE

UWE

<http://uwe.pst.ifi.lmu.de/>

UML-based Web Engineering

- Limbaj de modelare : profil UML (bazat pe mecanismele de extensie oferite de UML)
- Metamodel
- Proces de dezvoltare
- Instrumente : MagicUWE, plug-in pentru MagicDraw

UWE

UWE

UML-based Web Engineering

Modelele analizei:

- Cerințele funcționale: cazuri de utilizare, fluxuri de activități
- Date : modele de domeniu

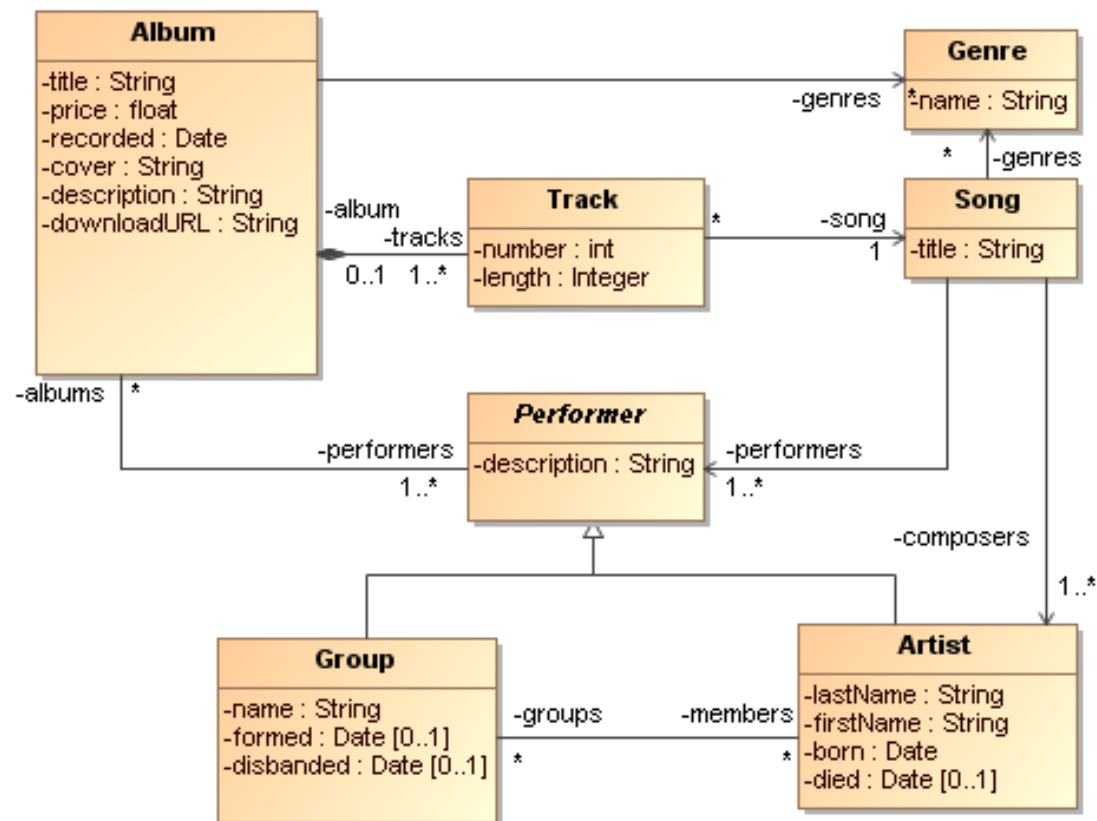
Modelele proiectării:

- Conținut
- Utilizator
- Navigare
- Prezentare
- Proces

UWE

UWE

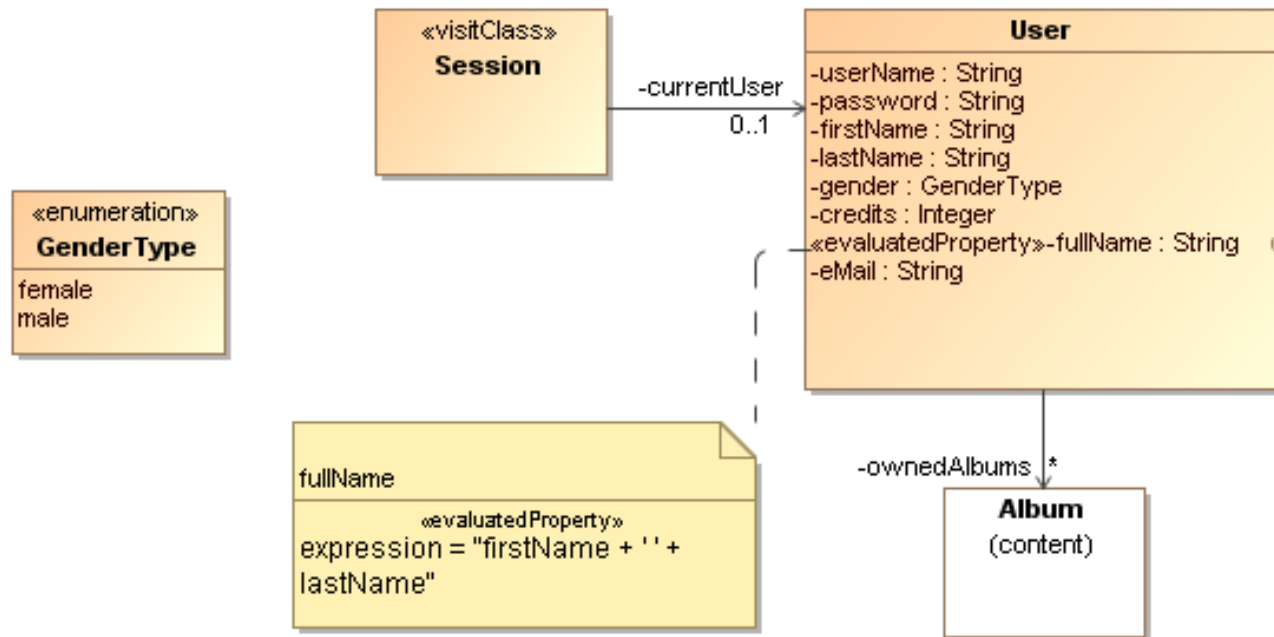
Modelul conținutului – diagrama de clase



UWE

UWE

Modelul utilizator – diagrama de clase; reprezentare sesiune (`<<visitClass>>`) și informații utilizator

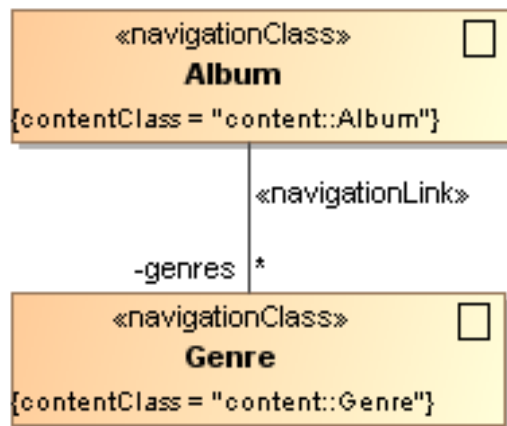


UWE

UWE

Modelul navigare

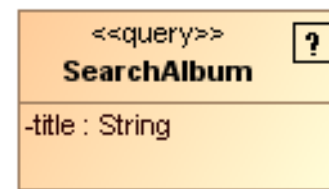
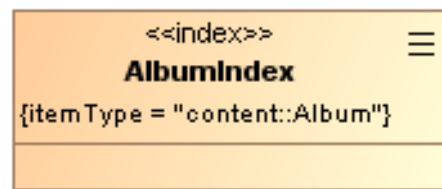
- nod hypertext (relativ la o clasă conținut) (<<navigationClass>>)
- hyperlink (<<navigationLink>>)



Pagini speciale



Primitive de acces în structura de navigare



UWE

UWE

Modelul prezentare

- Elemente *structurale*

- Container – elemente de interfață asociate cu o clasă de navigare sau de proces

`<<presentationGroup>>`

- Clasă de prezentare

`<<presentationPage>>`

- Container – clase prezentate alternativ

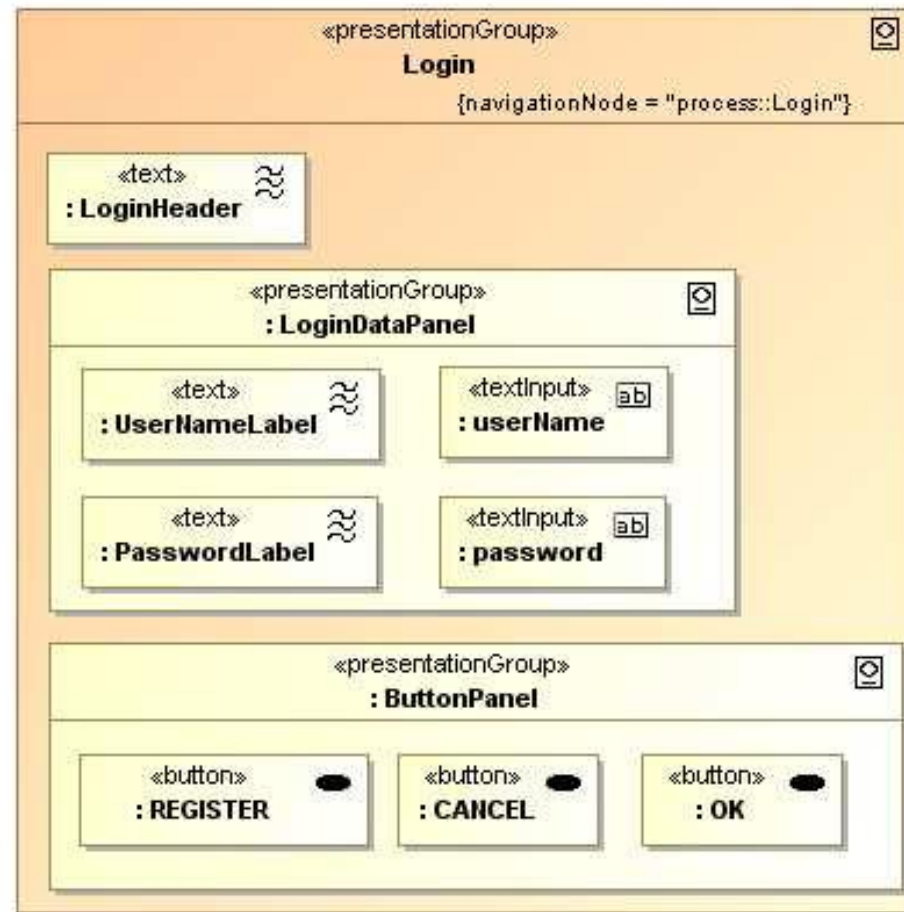
`<<presentationAlternatives>>`

- Elemente *interfață*

`<<anchor>>` `<<button>>` `<<text>>`

`<<image>>` `<<textinput>>`

`<<selection>>` ...



UWE

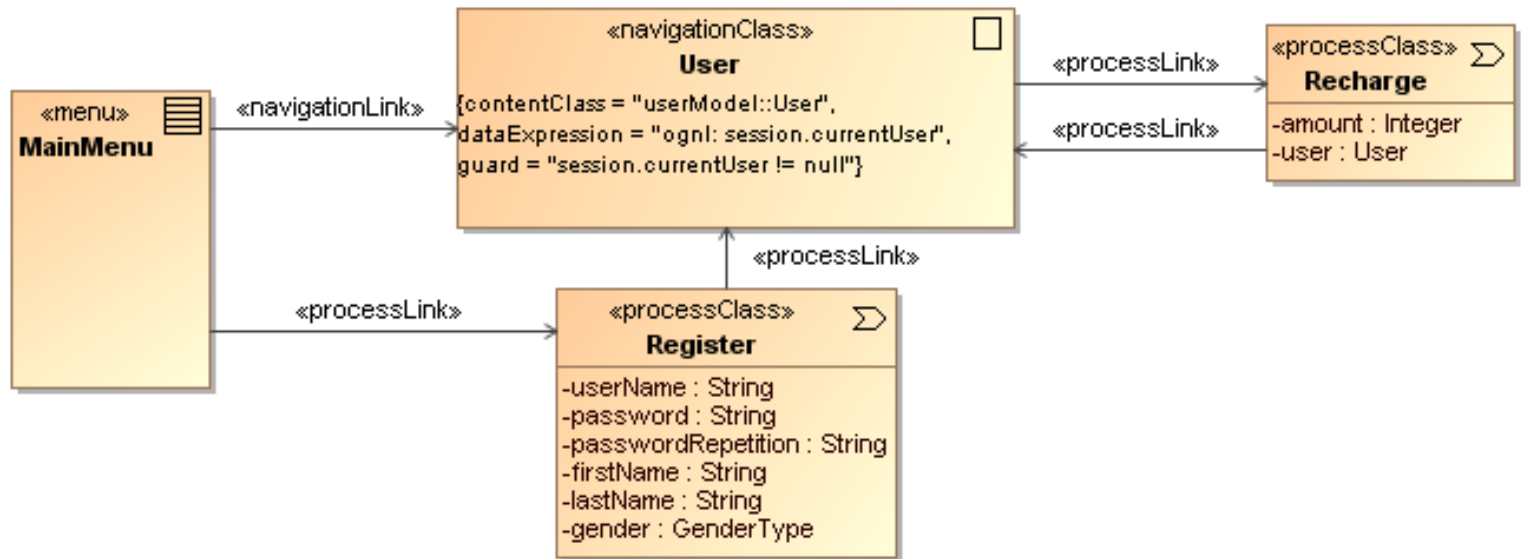
UWE

Modelul proces – reprezentarea aspectelor dinamice ale aplicației web

<<processClass>> : Procesul de ghidare a utilizatorului în aplicație

<<processLink>> :

Asociere între <<navigationClass>> (nod hypertext) și <<processClass>>



Bibliografie

Ian Sommerville, **Software Engineering** 10th edition, Ed. Addison Wesley, 2016

Capitolul 5, secțiunea 5

www.ifml.org

<https://www.omg.org/spec/IFML/1.0/PDF>

<https://issues.omg.org/issues/spec/IFML/1.0b2/fixed>

<https://my.webratio.com/learn/learningobject/interaction-flow-modeling-language-modeldriven-ui-engineering-of-web-and-mobile-apps-with-ifml-v-72?link=ln231x&inu1k.current.att1u=341&history=h1&fllbck=.sv2&cbck=wrReq22076>

www.webratio.com

După creare cont :

www.webratio.com/learn