

Universitatea Babeş-Bolyai
Facultatea de Ştiinţe Economice şi Gestiunea Afacerilor
Departamentul de Informatică Economică

**Nouă Metodologie şi Set Complet de Aplicaţii pentru
Conceptualizarea DSL-urilor
-studiu de caz pe dezvoltarea unei aplicaţii de modelare a lanţurilor valorice-
~Rezumatul tezei~**

Coordonator:
Prof. Dr. Ştefan Ioan Niţchi

Doctorand:
Costin Aurelian Răzvan

Cuprinsul tezei

Introducere	
1 Capitolul 1: Definiția formală a conceptului de limbă	
1.1 Introducere	
1.2 Definiții	
1.3 Definiția textuală a conceptului de limbă.....	
1) Setul de atribute și regula de compunere a notațiilor	
2) Setul de notații.....	
3) Setul de relații.....	
4) Setul de contexte	
5) Setul de adverbe	
1.4 Definiția formală a conceptului de limbă.....	
1.5 Conceptualizarea unui limbaj pe baza teoriei grafurilor	
1.6 Abordări asemănătoare.....	
1.7 Concluziile capitoului 1	
2 Capitolul 2: ML^2 Limbaj de modelare a DSL-urilor	
2.1 Introducere	
2.2 ML^2 Sintaxa detaliată.....	
2.3 ML^2 vs. MOF M3	
2.3.1 Analiză comparativă a modelor UML a limbajelor	
2.3.2 Alte abordări menite să îmbunătățească MOF M3	
2.4 Concluziile capitoului 2	
3 Capitolul 3: Aplicație pentru ingineria limbajelor	
3.1 Introducere	
3.2 Descrierea aplicației de inginerie a limbajelor	
3.3 Modulul de reinginerie	
3.3.1 Studiu de caz pe baza reingineriei VLML	
3.4 Modulul de inginerie a limbajelor	
3.5 Modulul de analiză a limbajelor	
3.6 Concluziile capitoului 3	
4 Capitolul 4: SCOR.....	

4.1	Introducere
4.2	Procesele SCOR
4.3	Metrici SCOR.....
4.4	SCOR în practică.....
4.4.1	HP
4.4.2	BMW Motorrad
4.4.3	STATOIL.....
4.5	SCOR vs. VRM.....
4.5.1	Compararea funcționalităților
4.5.2	Compararea arhitecturilor
4.5.3	Compararea proceselor
4.5.4	Compararea sceanriilor de utilizare
4.5.5	Concluziile secțiunii 4.5.....
4.6	Concluziile capitolului 4
5	Capitolul 5: Dezvoltarea unei aplicații de modelare SCOR
5.1	Introducere
5.2	Identificarea soft-urilor de modelare SCOR existente
5.3	Catalog de evaluare
5.4	Cerințele aplicațiilor de modelare SCOR.....
5.5	Analiza soft-urilor
5.5.1	Aris EasyScor
5.5.2	Tibco Business Studio
5.5.3	Cogniti
5.5.4	Metastorm ProVision
5.5.5	Nimbus Control
5.5.6	Xelocity Process Wizard
5.5.7	Software Analysis Overview
5.6	Conceptualizarea limbajului de modelare SCOR.....
5.6.1	Modelul – produs
5.6.2	Modelul – domeniu de aplicare.....
5.6.3	Modelul – regiuni de desfacere.....

5.6.4	Modelul – regiuni de aprovizionare
5.6.5	Modelul – harta geografică
5.6.6	Modelul – detalierea procesului
5.6.7	Modelul – proces de afaceri
5.6.8	Modelul – model organizațional
5.6.9	SCOR - DSL
5.6.10	Workflow de modelare a lanțurilor valorice
5.7	Concluziile capitolului 5
Concluzii și dezvoltări ulterioare	
1	Contribuții
2	Dezvoltări ulterioare
Bibliografie	

Abstract

Această teză de doctorat propune contribuții în trei domenii diferite. În primul rând, pornind de la o definiție inovativă a conceptului de limbaj, bazat pe o simplă ipoteză care afirmă că un cuvânt este definit de un set unic de atribute, autorii construiesc o teorie care explică fiecare element al limbajului, relațiile dintre aceste elemente, și modul în care sunt construite discursurile. Definiția formală a unui limbaj ar putea răspunde la o mulțime de probleme legate de procesarea limbajelor naturale (NLP).

În al doilea rând, pe baza acestei teorii, autorii descriu metodologia de conceptualizare a unui limbaj. În acest sens se propune o aplicație software menită să ajute inginerii de limbaj să descrie noi limbaje (DSL). Aplicația permite utilizatorilor săi să conceptualizeze, să valideze, să analizeze și să compare DSL-urile. Scopul principal al acestei aplicații este de a oferi o mai bună și mai detaliată înțelegere a conceptului de limbaj. Rezultatul folosirii acestui software se dorește a fi un set de date care poate fi ușor transformat și importat în platforme de meta-modelare pentru a genera noi instrumente de modelare. Graficul care este generat în urma procesului de conceptualizare descrie limbajul în detaliu și poate fi utilizat pentru a transmite cerințe, de la persoane responsabile de conceptualizarea limbajelor, către persoanele responsabile cu implementarea noilor DSL-uri în platforme de meta-modelare.

În al treilea rând, teza descrie un studiu de caz bazat pe conceptualizarea unui nou instrument de modelare pentru managementul lanțurilor valorice. Aplicația de modelare va fi construită pe baza unui limbaj de referință în modelarea lanțurilor valorice, în speță, se va folosi, așa numitul limbaj SCOR. Faza de conceptualizare a acestei aplicații se va face folosind aplicația propusă pentru ingineria limbajelor.

Introducere

În această lucrare, propunem abordări inovative în trei domenii diferite de cercetare: economie, informatică și lingvistică. Rezultatele prezentate în teză au fost obținute ca urmare a unei perioade de cercetare care a implicat scalarea nivelurilor de abstractizare.

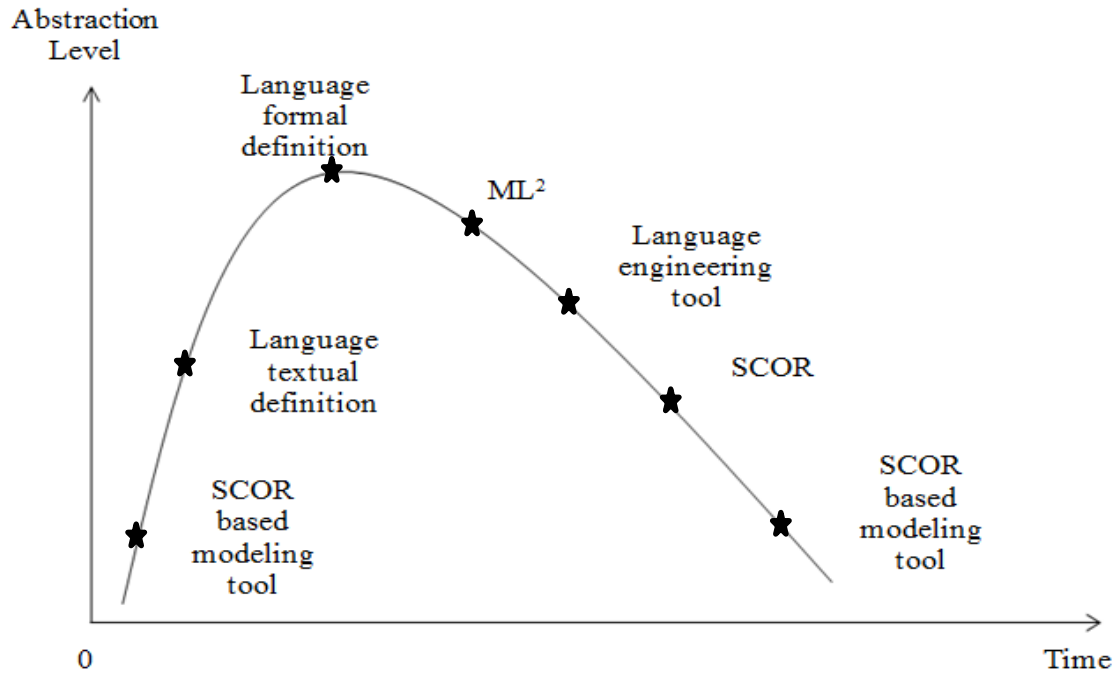
Domeniul economic a fost atins datorită obiectivului principal al cercetării, și anume, acela de a dezvolta o aplicație software de modelare a lanțurilor valorice.

Lingvistica a fost studiată cu scopul de a cataliza o nouă definiție a conceptului de limbaj, care să ne permită să înțelegem mai bine structura limbajelor și, în consecință, să înțelegem modul de conceptualizarea acestora.

Informatica a fost abordată datorită dezvoltării unui software menit să ajute inginerii de limbaj să dezvolte noi limbaje (DSL). Software-ul permite utilizatorilor săi să conceptualizeze, să valideze, să analizeze și să compare DSL-urile. Scopul principal al acestei aplicații este de a oferi o mai bună și mai detaliată înțelegere a conceptului de limbaj. Rezultatul folosirii acestui software se dorește a fi un set de date care poate fi ușor transformat și importat în platforme de meta-modelare pentru a genera noi instrumente de modelare. Graficul care este generat în urma procesului de conceptualizare descrie limbajul în detaliu și poate fi utilizat pentru a transmite cerințe, de la persoane responsabile de conceptualizarea limbajelor, către persoanele responsabile cu implementarea noilor DSL-uri în platforme de meta-modelare.

La momentul în care am pornit pe drumul acestei cercetării, scopul nostru principal a fost acela de a oferi o aplicație software care să permită utilizatorilor săi să modeleze, să analizeze, să simuleze, să optimizeze și să gestioneze lanțurile valorice. Pe drum, am descoperit că, pentru a putea atinge acest obiectiv, este necesară o cercetare mai profundă și mai abstractă a conceptului de limbaj și al modului de conceptualizare a limbajelor. Astfel, am escaladat pe ierarhia nivelelor de abstractizare și am încercat să obținem o înțelegere mai detaliată în ceea ce privește definițiile limbajelor, structura lor, și metodologia de conceptualizare a noilor limbaje. După această escaladare a nivelelor de abstractizare, am dobândit cunoștințele necesare pentru a putea conceptualiza un limbaj de modelare a lanțurilor valorice, iar în final, să folosim acest limbaj pentru a dezvolta aplicația de modelare a lanțurilor valorice, care a fost, de altfel, scopul inițial al cercetării.

Considerăm că, cel mai bine, am putea explica evoluția cercetării noastre folosindu-ne de o diagramă Gauss. În imaginea de mai jos am marcat cu câte o steluță nivelele de abstractizare atinse și studiate. Aceste puncte, s-au transformat, ulterior, în capitolele tezei de față.



La momentul stabilirii obiectivului de cercetare, aveam puține informații despre cum ar trebui să arate și să funcționeze o aplicație software de modelare a lanțurilor valorice. Dacă partea tehnică de care aveam nevoie pentru derularea cercetării a constituit o problemă mai mică, în ceea ce privește bunul mers al cercetării, conceptualizarea limbajului care va fi folosit pentru a modela lanțurile valorice, tehnica de modelare a acestora, precum și algoritmi de simulare și analiză au constituit un domeniu total necunoscut pentru noi. Prin urmare, am fost nevoiți să escaladăm nivelele de abstractizare pentru a putea defini și înțelege conceptul de limbaj. Ipoteza principală de la care am pornit a fost că, dacă am putea reprezenta grafic structura unui limbaj, atunci ar trebui să fie posibil să dezvoltăm un instrument de modelare a limbajelor, care ar putea fi folosit, apoi, pentru a conceptualiza un limbaj de la zero.

Primul pas pe care l-am făcut, a fost în direcția definirii textuale a unui limbaj. Ne-am bazat eforturile pe studiul literaturii de specialitate din acest domeniu. Din păcate, fiecare pistă aleasă, ne-a condus către drumurii înfundate, deoarece nu am putut folosi definițiile lingvistice existente, pentru a înțelege structura limbajelor. Astfel, așa cum se poate observa pe graficul de mai sus, ne-am decis să scriem propria teorie despre limbaje. Următorul pas a fost să dezvoltăm o definiție formală a conceptului de limbaj. Acest pas reprezintă cel mai înalt nivel de abstractizare atins. Scopul acestei etape a fost de a identifica mecanismul care stă în spatele unui om sau mașinii, care este capabil/ă să conceptualizeze un limbaj. Întrebările fundamentale de cercetare,

la acest moment au fost: "Cum apare un cuvânt?", "Ce declanșează necesitatea de conceptualizare al unui nou cuvânt?", "Care este sensul real al acestui cuvânt?". Prezumția noastră a fost că un cuvânt descrie și reprezintă un set unic de atribute. Prin atribut, am înțeles o percepție a unui inginer de limbaj. Mai multe aspecte despre această teorie sunt detaliate în Capitolul 1.

După ce am reușit să descriem formal conceptul de limbaj, a trebuit să ne demonstrăm că această definiție este validă. Astfel, a luat naștere limbajul de modelare a limbajelor (limbaj pe care l-am denumit ML2). Practic, am conceptualizat un limbaj auto-descriptiv, care să permită inginerilor de limbaj să descrie noi limbaje (DSL-uri). Limbajul de modelare a fost testat de inginerii de limbaj de la firma BOC din Viena. Rezultatele folosirii acestui limbaj pentru re-ingineria unor limbaje dezvoltate în platforma de meta-modelare ADOxx sunt prezentate în Capitolul 2. Deoarece rezultatele au fost încurajatoare, am decis să dezvoltăm primul modul al aplicației de ingineria limbajelor. Modulul a avut ca scop re-ingineria automată a unui DSL implementat într-o platformă de modelare. Modulul avea menirea de a genera o reprezentare grafică a unui DSL, folosind ML2.

Rezultatele au continuat să fie încurajatoare, astfel, am decis să dezvoltăm alte două module pentru aplicația de inginerie a limbajelor. Astfel, am dezvoltat un modul de analiză a DSL-urilor, cu scopul de a valida un DSL implementat într-o platformă de meta-modelare, și un modul de inginerie, care poate fi folosit pentru a conceptualiza un limbaj, bazat pe o abordare și metodologie comună. Software-ul de modelare a limbajelor este prezentat și detaliat în Capitolul 3.

Ultima parte a cercetării a fost dedicată pentru atingerea obiectivul inițial al cercetării, și anume, dezvoltarea unui instrument de modelare a lanțurilor valorice. Având cunoștințele necesare, și un set complet de instrumente, care ar putea fi folosite pentru a conceptualiza un limbaj, ne-am bazat cercetarea pe o serie de metodologi bine-cunoscute pentru managementul lanțurilor valorice, precum VRM și SCOR. Astfel în capitolele 4 și 5 prezentăm procesul de conceptualizare a aplicației de modelare a lanțurilor valorice.

1 Capitolul 1: Definiția formală a conceptului de limbă

Ferdinand de Saussure a fost primul autor care a făcut distincția între "limbă" și "limbaj", distincție care a fost acceptată la nivel global [3]. El afirmă că un "limbaj" este, de fapt, o

instanță a unei limbi. Din punct de vedere a lui de Saussure limba engleză sau franceză sau română, etc. sunt "limbaje" în timp ce "limba" în sine este conceptul care descrie toate "limbajele". Astfel, definiția unui "limbaj" poate suna astfel "limbajul este o limbă care (...)". Dar, atunci, ce este o limbă? În acest capitol, vom încerca să definim conceptul de limbă. Pornim de la prezumția că limba este reprezentată de un set de cuvinte, ce respectă o structură rigidă și un set de reguli care, dacă este aplicat pe setul de cuvinte, permite crearea discursurilor. Deoarece "limbajul este o limbă care...", definiția conceptului de limbă va fi adevărată pentru orice limbaj.

Propunem următoarea schemă pentru a face distincția clară între limbă, tipuri de limbaj și limbaje:

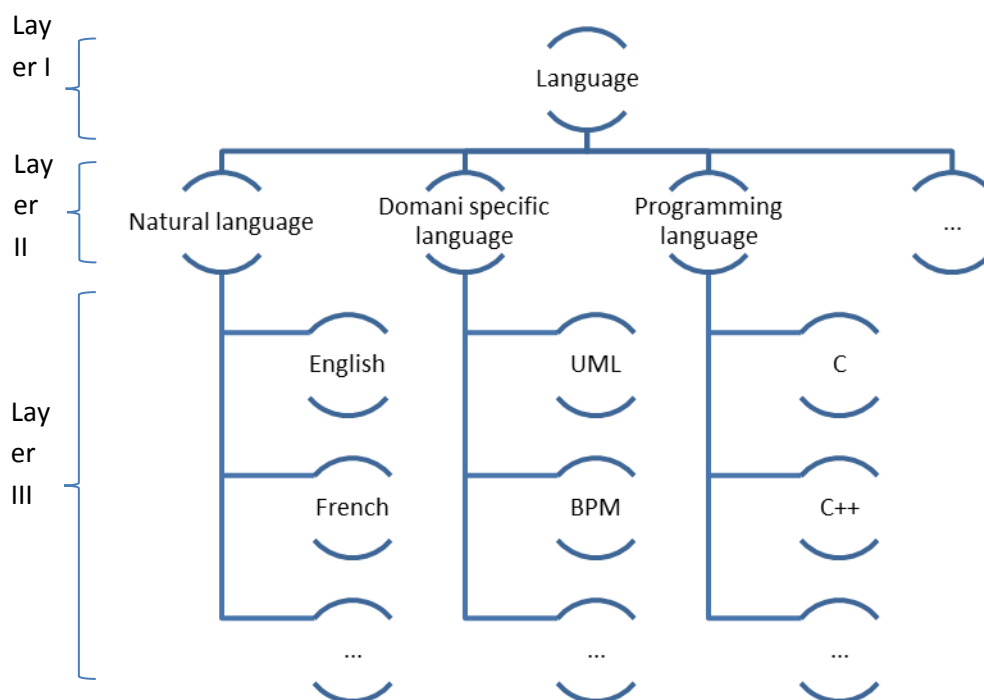


Figura 1 Ierarhia limbajelor

Nivelul 1 cuprinde doar un singur nod: "limba". La acest nivel de abstractizare, conceptul de limbă ar trebui să fie perceput ca un concept care definește toate tipurile de limbaje.

Am definit limbajul în primul paragraf al acestei secțiuni. Acum, am putea extinde această definiție și spune că la nivelul 1 de abstractizare o limbă este definită de un set de elemente de limbaj și un set de reguli care guvernează construcția discursurilor. Această definiție va fi valabilă pentru toate limbile de nivelul 3. Dacă împărțim setul de elemente lingvistice în 5 sub-seturi: un set de atribute, un set de notații, un set de relații, un set de contexte, și un set de

adverbe, putem spune că un set de reguli care guvernează procesul de creare a discursurilor în orice limbă ar cuprinde 5 reguli de bază: regula de compunere a notației, regula de descriere a relației, regula de legare a subiectului, regula de legare a obiectului, și regula de asociere a contextului. Considerăm această definiție ca fiind definiția textuală completă a conceptului de limbă.

1.1 Definiția formală a conceptului de limbă

Definiție: O limbă este reprezentată de o structură de tipul 10-tuple= $\langle A1, N, R, C, A2, \alpha, \omega, s, o, \varepsilon \rangle$ unde:

- $A1$ este un set finit de atribute;
 - Definem “ $U:U$ ” ca fiind “atributul vid”. Atributul vid este un atribut care nu are nici tip de data, nici valoare, astfel, următoarea relație este adevărată: $A1 \cup \{U:U\} = A1$.
 - Atributul care are un tip de dată, dar nu are valoare va fi reprezentat de “Tip de dată”: U .
 - $A1 \cup \{\text{“Tip de dată”}: U\} = A$ unde A conține toate elementele din $A1$ și elementul (“Tip de dată”: U).
- N este un set finit de notații;
 - Definem “ $U:U:\{U:U\}$ ” pentru a reprezenta “notația vidă”. Notația vidă este o notație care nu are denumire, nici reprezentare și este asociată unui set de atribute care cuprinde doar atributul vid.
 - Astfel, se poate defini următoarea relație $N \cup (U: U: \text{Attr}\{U:U\}) = N$.
 - Definem “elementul incipient” ca fiind o notație vidă care primește o denumire. Astfel “Nume”: $U: \text{Attr}\{U:U\}$ este “elementul incipient” al limbajului.
- R este un set finit de relații;
 - R nu are “element incipient”.
- C este un set finit de contexte;
 - C nu are “element incipient”.
- $A2$ este un set finit de adverbe;
 - Definem “ $U:U$ ” pentru a reprezenta “adverbul vid”. Adverbul vid este un adverb care nu are nici tip de data, nici valoare, astfel, următoarea relație este adevărată: $A2 \cup \{U:U\} = A2$.

- Adverbul care are un tip de dată, dar nu are valoare va fi reprezentat de “Tip de dată”: U.
- $A2U\{\text{“Tip de dată”}: U\}=A$ unde A conține toate elementele din A2 și elementul (“Tip de dată”: U).
- “ α ” este o funcție asociată cu “regula de compunere a notației”.Funcția este definită ca, $\alpha: A1^m \rightarrow N$ unde $m \leq c$ (cardinalitatea lui A1) este un întreg. Funcția asociază un set de atribute unei notații.

“ α ” are următoarele proprietăți:

1. α e surjectivă, adică: orice notație are corespondent un set de atribute din A1.
 2. Dacă două sau mai multe notații sunt asociate cu același set de atribute, indiferent de ordinea lor, considerăm aceste notații ca fiind sinonime.
 3. Dacă α este bijectivă, atunci limbajul nu va permite sinonime, iar fiecare set de atribute va fi asociat cu o notație unică. Considerăm un astfel de limbaj un “limbaj complet”.
 4. Dacă α este bijectivă atunci putem defini $\bar{\alpha}$ ca fiind funcția inversă lui α . $\bar{\alpha}$ va avea următoarea proprietate: $\bar{\alpha}=\alpha^{-1}$; $\bar{\alpha}: N \rightarrow A1^m$ este o funcție care asociază o notație unui set de atribute.
- “ ω ” este o funcție asociată cu “regula de descriere a relației”.Funcția este definită ca $\omega: A2^m \rightarrow R$ unde $m \leq c$ (cardinalitatea lui) este un întreg. Funcția asociază un set de adverbe unei relații.

ω are următoarele proprietăți:

1. ω este surjectivă, adică: aceeași relație poate să fie descrisă de seturi diferite de adverbe. Exemplu, relația “fuge” poate fi descrisă de {viteză: 20 km/h} sau de {distanță: 11km}.
 2. O relație care nu are adverbe va fi asociată cu {U:U} setul format din adverbul vid.
- s este o funcție asociată cu “regula de legare a subiectului”. Funcția este definită ca $s: N^*R \rightarrow S$. Funcția asociază o pereche formată din o notație și o relație unui set S. Perechile din setul S identifică care notații pot fi subiectul căror relații.

s are următoarele proprietăți:

1. s este surjectivă, adică fiecare element din S are asociat un element din N^*R .
Pe de altă parte, nu orice element din N^*R poate fi regăsit în S . S conține perechi care au fost specificate de inginerul de limbaj cu scopul de a defini care notații pot fi subiecte a căror relații.
 2. Orice element din R trebuie să aibă minim un posibil subiect.
- o este o funcție asociată cu “regula de legare a obiectului”. Funcția este definită ca $o: N^*R \rightarrow O$. Funcția asociază o pereche formată dintr-o notație și o relație unui set O . Perechile din O au menirea de a arăta care notații pot fi obiecte căror relații.
 o are următoarele proprietăți:
 1. o este surjectivă, adică orice element din O are asociat un element din N^*R .
Pe de altă parte, nu orice element din N^*R se va regăsi în O . O conține perechi specificate de inginerul de limbaj, cu scopul de a arăta care notații pot fi obiecte căror relații.
 2. Pot exista elemente din R care nu au obiect.
 - ε este o funcție asociată cu “regula de asociere a contextului”. Funcția este definită ca $\varepsilon: NUR \rightarrow C$. Funcția asociază notații sau relații la unul sau mai multe contexte pentru a specifica sensul lor.
 ε are următoarele proprietăți:
 1. ε este surjectivă, adică orice context are cel puțin două elemente de limbaj asociate: o notație și o relație. Scopul contextului este de a sugera înțelesul discursurilor.

Tot ceea ce operează în conformitate cu aceste specificații este un limbaj. Suma totală de limbajelor aparține clasei “limbă”

2 Capitolul 2: ML^2 Limbaj pentru modelarea limbajelor

Acest capitol prezintă un nou limbaj de modelare a limbajelor, pe care l-am denumit ML^2 . Ideea unui limbaj de conceptualizare și modelare a limbajelor nu este nouă. MOF este un bun exemplu în acest sens. MOF este un limbaj creat pentru a descrie bine-cunoscutul limbaj de modelare UML. Cu timpul, MOF s-a impus ca standard de modelare a limbajelor.

Pe de altă parte, inginerii de limbaje se confruntă cu probleme reale atunci când încep un nou proiect de dezvoltare al unui limbaj. Problema e că, aceștia ar avea nevoie de un limbaj mai

specific decât al treilea nivel al MOF-ului pentru a conceptualiza limbajul. În ceea ce privește această nevoie, noi propunem un limbaj care permite inginerilor de limbaj să schițeze și să modeleze mai ușor limbajul.

Am definit limbajul nostru de limbaj de modelare dezvoltând idei specificate în teoria MOF și bazându-ne pe o perioadă de cercetare care a implicat re-ingineria unor limbaje dezvoltate și implementate în platforma de meta-modelare ADOxx. În ceea ce privește aceste aspecte, cercetarea noastră a constat în încercarea de a modela limbajele existente, utilizând abordarea MOF, ca fiind standardul pentru meta-meta-modelare. Chiar dacă am fost în măsură să prezentăm structura limbajelor, nu am fost în măsură să reprezentăm aspecte privind sintaxa sau gramatica limbajelor. În acest sens am considerat că este nevoie de o nouă abordare a modelării limbajelor.

Pentru a conceptualiza ML² am adoptat o tehnică de tipul top-down, începând cu analiza domeniului modelării limbajelor. Această perioadă ne-a permis să identificăm următoarelor aspecte pe care ar trebui să le abordăm:

1. sintaxa concretă a elementelor de limbaj care nu au nici o reprezentare (clase care oferă cadru inițial). Exemplu: elementul incipient, o notație care nu are atribute sau reprezentare (EI: Attr{U: U});
2. Sintaxa concretă a elementelor de limbaj care au reprezentare, adică clase definite de către inginerul de limbaj, care reprezintă obiecte din lumea reală. Exemplu: oricare dintre notațiile precum C: C: Attr{c:c1};
3. Sintaxa concretă a relațiilor (clase definite de către inginerul de limbaj care reprezintă posibilele acțiuni și interacțiuni ale obiectelor din lumea reală);
4. Sintaxa concretă a contextelor și anume structurile compozite în care combinația de notații și relații au un sens.
5. Gramatica, adică am definit patru tipuri de conectori între clase: “moștenire” (care are același înțeles ca în UML), “face parte” (care este un conector care leagă notații și relații de un context specific), “conector subiect și obiect” (care sunt conectorii între o notație și o relație. Acești conectori denotă și ordinea în care elementele apar într-o propoziție).

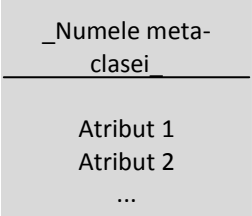
Limbajul de modelare propus de noi se diferențiază de UML, deoarece este menit să abordeze un spectru mai larg de limbaje. Dacă UML se adresează, în principal, limbajelor de programare, care necesită un formalism precis, limbajul nostru permite un formalism mai flexibil, care

permite legarea notațiilor cu așa numitele, relații. De exemplu, dacă relația "egal" trebuie să aibă o semnificație precisă în programare, în alte limbi, putem spune că există o diferență între o relație "egal" și o relație "egal" care are asociat adjectivul "mai mult sau mai puțin".

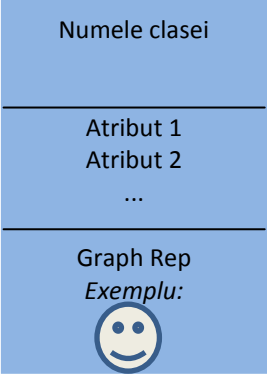
În contrast cu alte abordări asemănătoare, limbajul propus de noi dispune de mijloace mai generale pentru a specifica modelul unui limbaj. Astfel, limbajul nostru ar trebui să fie considerat mai abstract și universal decât UML-ul, dar, în același timp, specializat pe un anumit domeniu, și anume modelarea limbajelor.

2.1 ML² – detalierea sintaxei

În această secțiune vom prezenta în detaliu sintaxa propusă a ML²-ului.

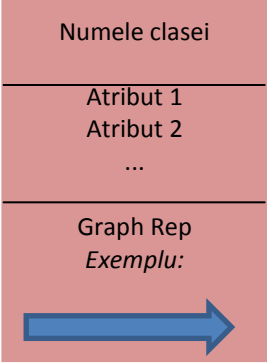
Denumire notație	"Meta Clasă"
Reprezentare grafică	
Descriere	Această notație este folosită pentru a modela cadrul inițial al limbajului. Culoarea gri, sugerează faptul că această clasă este o meta-clasă. O meta-clasă este o notație care nu propune o reprezentare a instanțierii ei.
Tip	Clasă
Relații	Poate avea relații cu: <ul style="list-style-type: none"> - alte "Meta-Clase"; - "Clase definite de inginerul de limbaj"; - "Clase relație".
Conectori	Tip de conectori: <ul style="list-style-type: none"> -moștenire- între 2 "Meta clase", o "Meta clasă" și o "Clasă definite de către inginerul de limbaj"; - conector subiect sau obiect- între o "Meta clasă" și o "Clasă relație".

Tabel 1 "Meta Clasă" - descriere

Nume	"Clasa definită de inginerul de limbaj"
Reprezentare grafică	
Descriere	<p>"Clasa definită de inginerul de limbaj" este folosită pentru a modela obiectele din lumea reală. Propunem, ca o reprezentare grafică a acestei clase, un dreptunghi albastru, care este împărțit în trei secțiuni. Prima secțiune permite inginerului de limbaj să atribuie un nume clasei. Cea de a doua secțiune definește atributele. A treia secțiune prezintă reprezentarea grafică a modului în care această clasă se va instanția în momentul folosirii ei în discurs. Culoarea albastră are scopul de a diferenția această clasă de "Meta Clasă" și "Clasa relație". "Clasa definită de inginerul de limbaj" este folosită pentru a reprezenta notații care au reprezentare în momentul instanțierii ei.</p>
Tip	Clasă
Relații	<p>Poate avea relații cu:</p> <ul style="list-style-type: none"> - "Meta-Clase"; - "Clase relație"; - "Contexte".
Conectori	<p>Tip de conectori:</p> <ul style="list-style-type: none"> -moștenire- între o "Meta Clasă", și o "Clasă definită de inginerul de limbaj"; -conector subiect și obiect- între o "Clasă definită de inginerul de limbaj" și o "Clasă relație"; -aparține la- între o "Clasă definită de inginerul de limbaj" și un


	"Context".
--	------------

Tabel 2 "Clasa definită de inginerul de limbaj" - descriere


Nume	"Clasa relație"
Reprezentare Grafică	
Descriere	"Clasa relație" este utilizată pentru modelarea acțiunilor (verbelor) din lumea reală. După cum se vede mai sus, propunem, ca reprezentare grafică a acestei clase, un dreptunghi roșu, care este împărțit în trei secțiuni. Prima secțiune permite inginerului de limbaj să atribuie un nume relației. Cea de a doua secțiune permite specificare posibilelor "adverbe" ale relației. A treia secțiune prezintă reprezentarea grafică a instanțierii clasei în formarea discursurilor. Culoarea are scopul de a diferenția această clasă de "Meta Clasă" și "Clasa definită de inginerul de limbaj".
Tip	Clasă
Relații	Poate avea relații cu: <ul style="list-style-type: none"> - "Meta-Clase"; - "Clase relație"; - "Contexte".
Conectori	Tip de conectori: <ul style="list-style-type: none"> -conector subiect și obiect- între o "Clasă definită de inginerul de limbaj" și o "Clasă relație";

	-aparține la- între o "Clasă definită de inginerul de limbaj" și un "Context".
--	--

Tabel 3 "Clasa relație" - descriere


Nume	"Context"
Reprezentare grafică	
Descriere	"Contextul" este folosit pentru a modela contextul în care declarațiile au sens.
Tip	Container
Relații	Poate avea relații cu: - "Meta-Clase"; - "Clasa definită de inginerul de limbaj".
Conectori	Tip de conectori: -aparține la- între o "Clasă definită de inginerul de limbaj" și un "Context"; -aparține la- între o "Clasă relație" și un "Context".

Tabe 4 "Context" - descriere

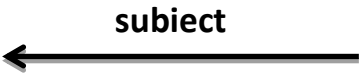
Nume	"moștenire"
Reprezentare grafică	
Descriere	Conectorul "moștenire" este folosit pentru a construi structura inițială a limbii. Acesta arată ierarhia de clasă și modul în care sintaxa definite se mapează pe meta-modelul inițial al limbajului conceptualizat.
Tip	Conector
Clase conectate	Poate conecta: - o "Clasă definită de inginerul de limbaj" la o "Meta clasă"; - o "Clasă definită de inginerul de limbaj" la o altă "Clasă definită"

	de inginerul de limbaj"; - o "Meta clasă" la o altă "Meta clasă";
--	--


Tabel 5 "Moștenire" - descriere

Nume	"aparține la"
Reprezentare grafică	
Descriere	Conectorul "aparține la" este folosit pentru a moda care "Clasă definită de inginerul de limbaj" sau "Clasă relație" poate fi folosită, în care "Context". Acest lucru permite inginer de limbaj să stabilească contextul în care declarațiile pot fi construite folosind doar elemente permise.
Tip	Conector
Clase conectate	Poate conecta: - o "Clasă definită de inginerul de limbaj" la un "Context"; - o "Clasă relație" la un "Context".

Tabel 6 "Aparține la" - descriere

Nume	"conector subiect"
Reprezentare grafică	
Descriere	"Conectorul subiectul" este folosit pentru a modela care "Clasă definită de inginerul de limbaj" este subiectul pentru o "Clasă relație".
Tip	Conector
Clase conectate	Poate conecta: - o "Clasă definită de inginerul de limbaj" la o "Clasă relație";

Tabel 7 "Conector subiect" - descriere

Nume	"conector obiect"
Reprezentare grafică	
Descriere	"Conectorul obiect" este folosit pentru a modela care "Clasă definită de inginerul de limbaj" este obiect pentru o "Clasă relație".

Tip	Conector
Clase conectate	Poate conecta: - o "Clasă relație" la o "Clasă definită de inginerul de limbaj".

Tabel 8 "Conector Obiect" - descriere

3 Capitol 3: Aplicație pentru ingineria limbajelor

În acest capitol vom descrie o aplicație de inginerie a limbajelor. Aplicația propusă se bazează pe ML^2 pentru crearea de reprezentări vizuale ale limbajelor.

Aplicația propune trei module:

1. Un modul de reinginerie a limbajelor;
2. Un modul de analiză a limbajelor;
3. Un modul de descriere (inginerie) a limbajelor.

Aplicația de inginerie a limbajelor a fost dezvoltată pe o arhitectura bazată pe componente. A fost implementată ca o aplicație de sine stătătoare, folosind Java. Componentele de bază sunt: componenta de intrare, componenta de structurare a unui meta-model, componenta de reprezentare, și componenta de ieșire și generare a rezultatului.

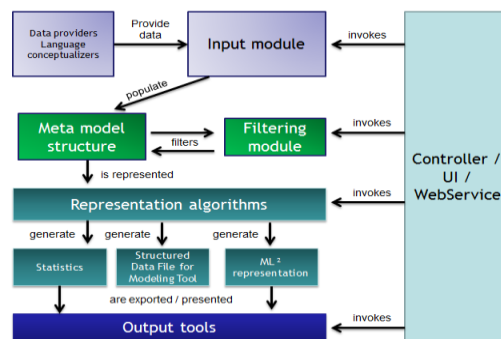


Figura 2 Arhitectura aplicației de ingineria limbajelor

3.1 Modulul de reinginerie

Modulul de reinginerie propune o interfață utilizator foarte simplă. Interfața este formată din 5 panouri. Primele 4 panouri (de intrare, rezultate, imagine, și alege acțiuni) necesită inputuri de la utilizator, în timp ce al cincilea panou este utilizat pentru a afișa o previzualizare a rezultatelor reingineriei. Scenariul de utilizare poate fi descris astfel:

1. În primul panou utilizatorului i se cere să furnizeze tipul de fișier care va fi procesat (fișier ML^2 sau fișier ADOxx), iar apoi să furnizeze fișierul în sine. Fișierul trebuie să conțină descrierea limbajului așa cum este definit de către inginerul de limbaj;

2. În al doilea panou (rezultate), utilizatorul este rugat să aleagă tipul de rezultate așteptate. Există două tipuri de rezultate enumerate: (1) "statistici", rezultat care prezintă limbajul într-un document care conține un tabel în care este descris (documentul arată ca cel folosit pentru definirea limbajului ML² în capitolul 2), (2) "privire de ansamblu", rezultat care asigură reprezentarea grafică a limbajului folosind ML²;
3. În al treilea panou, utilizatorul este rugat să selecteze tipul de fișier în care ar trebui să fie listate rezultatele;
4. În al patrulea panou, utilizatorul poate alege să i se afișeze doar anumite aspecte ale limbajului;
5. După ce utilizatorul introduce toate informațiile necesare, în panoul cinci, se va afișa o previzualizare a rezultatelor, iar documentul rezultat este generat.

3.2 Modulul de inginerie a limbajelor

Am folosit metodologia de inginerie a limbajelor descrisă în Capitolul 1, pentru a permite inginerilor de limbaje să conceptualizeze, cu ușurință, un nou DSL, care ar putea fi apoi importat într-o platformă de meta-modelare pentru a crea un nou instrument de modelare. Ne-am bazat activitatea noastră pe platforma de meta-modelare ADOxx, pentru a demonstra validitatea funcționalității modulului. Astfel, aplicația are menirea de a oferi o structură de date care poate fi importată în ADOxx pentru a crea un nou instrument de modelare. Cu toate acestea, aplicația poate fi extinsă, cu ușurință, pentru a oferi structuri de date caracteristice și altor platforme de meta-modelare care furnizează un mecanism de import DSL.

Scenariul de utilizare a aplicației este simplu și intuitiv. În interfață, utilizatorul adaugă elemente de limbaj în funcție de tipul acestora. Nu există nici o restricție privind etapele în care trebuie adăugate elementele. Este la latitudinea utilizatorului dacă el începe prin definirea notațiilor, a relațiilor, a atributelor, sau a contextelor.

După ce utilizatorul introduce toate elementele lingvistice și consideră că limbajul conceptualizat a fost descris în totalitate, el va acționa butonul de salvare a limbajului. În acest moment nici o validare a limbajului nu este făcută. Pentru a analiza propriul limbaj conceptualizat, utilizatorul poate folosi modulul de analiză a limbajului.

3.3 Modulul de analiză al limbajului

Modulul de analiză al limbajului a fost dezvoltat pentru a rezolva două probleme importante în domeniul ingineriei limbajelor: analiza limbajelor și compararea limbajelor.

În primul rând, modulul folosește formalismul limbajului definit în Capitolul 1 pentru a analiza structura unui limbaj. Rezultatul acestei funcționalități este un document care conține statistici referitoare la limbaj. Documentul conține informații cu privire la numărul de notații, relații, contexte, atribute, și adverbe. Totodată, documentul prezintă atributele care sunt asociate cu notațiile, care adverbe sunt asociate cu care relații, care propoziții sunt valabile și cum ar trebui să fie acestea înțelese, în funcție de contextele alocate. Documentul informează, de asemenea, utilizatorii în cazul în care elemente de limbaj au fost declarate inconsistente, adică elemente care nu aderă la formalismul impus.

În al doilea rând, modulul oferă pârghii pentru compararea a două limbaje. Ideea, aici, este de a oferi utilizatorului o modalitate de a valida limbajul, prin compararea lui cu o cerință inițială. Considerăm limbajul descris folosind ML^2 ca fiind cerința inițială, în timp ce limbajul, care este destinat să fie validat este cel furnizat de o platformă meta-modelare, în care acesta a fost implementat. Cele două limbaje sunt comparate sub forma unor grafuri.

4 Capitolul 4: SCOR

În acest capitol am analizat teoria în domeniul lanțurilor valorice cu intenția de a conceptualiza un nou instrument de modelare a operațiunilor specific unui astfel de lanț. Am ales să ne bazăm cercetarea pe un DSL foarte bine-cunoscut în teoria lanțurilor valorice, și anume, SCOR.

SCOR este un cadru de modelare și analiză, care oferă linii directoare pentru gestiunea lanțurilor valorice. SCOR este produs de Supply Chain Consiliului (SCC), o corporație non-profit, de talie mondială, care conferă calitatea de membru tuturor companiilor interesate în aplicarea și schimbul de practici standardizate în domeniul managementului lanțurilor valorice. În plus, SCOR ajută companiile să adopte cele mai bune practici [38] și, prin urmare, poate fi considerat o abordare de modelare normativă [39].

Modelul SCOR a fost dezvoltat pentru a defini toate activitățile de afaceri asociate cu lanțul valoric [40]. Acesta abordează aspecte precum: toate interacțiunile cu clienții, toate tranzacțiile materiale și toate interacțiunile de piață. Modelul nu încearcă să descrie fiecare proces de afaceri sau de activitate organizațională. Concret, modelul nu abordează: vânzările sau marketingul,

dezvoltarea de produse, activitățile de cercetare și dezvoltare, sau unele elemente de asistență pentru clienți [38], [39].

Cadrul este construit în jurul a două seturi de bază: o bibliotecă standard pentru procesele de modelare a lanțului valoric, care permite companiilor să comunice și să înființeze medii colaborative, și un set de metrice standardizate, care permite managerilor lanțului valoric să identifice problemele acestuia și să stabilească strategii de management pentru eficientizarea și optimizarea lui.

SCOR propune o structură ierarhică pentru descrierea proceselor specifice lanțurilor valorice. Blocul principal al modelului SCOR este format din cinci procese de bază: strategia, furnizorii, producția, livrarea, și returul.



Figura 3 Modelul SCOR [41]

Procesele de nivel 1 sunt compuse din procese de nivelul 2. Acestea din urmă sunt cunoscute sub denumirea de categorii de procese [38]. Scopul acestor procese este de a oferi un cadru standardizat pentru a descrie detaliat tipurile de proces. Tipul de proces Strategie este reprezentat folosind notațiile P2, P3, P4, P5 și reprezintă planificarea furnizorilor, planificarea producției, planificarea livrării și planificarea returului [40]. Un exemplu de model al unui lanț valoric este reprezentat de figura următoare:

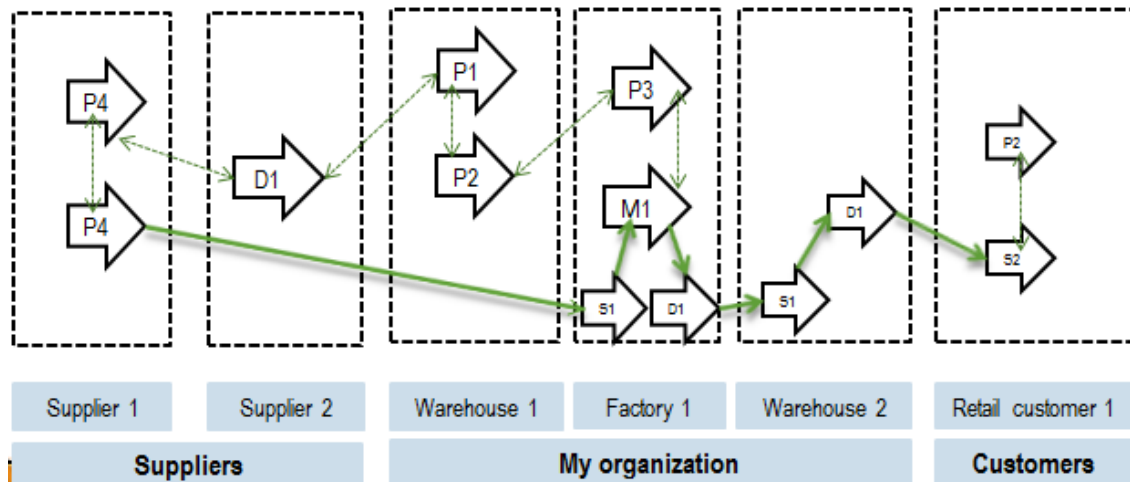


Figure 4 Thread Diagram [42]

Fiecare process de nivelul 2 este detaliat mai departe de procese de nivel 3. Secvența în care aceste procese sunt executate influențează performanța proceselor de nivel 1 și în final, performanța întregului lanț valoric. Procesele de nivel 3 sunt cunoscute sub denumirea de activități [38]. Așadar, dacă modelarea nu continuă cu descrierea proceselor la nivelul 4 atunci, nivelul 3 de procese este considerat ca fiind nivelul activităților standard SCOR.

5 Capitolul 5: Dezvoltarea unei aplicații de modelare SCOR

În acest capitol vom descrie o nouă aplicație pentru modelarea lanțurilor valorice. Aplicația se bazează pe specificațiile SCOR, pe care le-am extins pentru a oferi caracteristici pe care, noi le considerăm obligatorii pentru a putea surprinde un set cât mai complet de aspecte ale lanțurilor valorice.

5.1 Cerințe funcționale

Considerăm că aplicația noastră de modelare ar trebui să adreseze următoarele aspecte ale lanțurilor valorice:

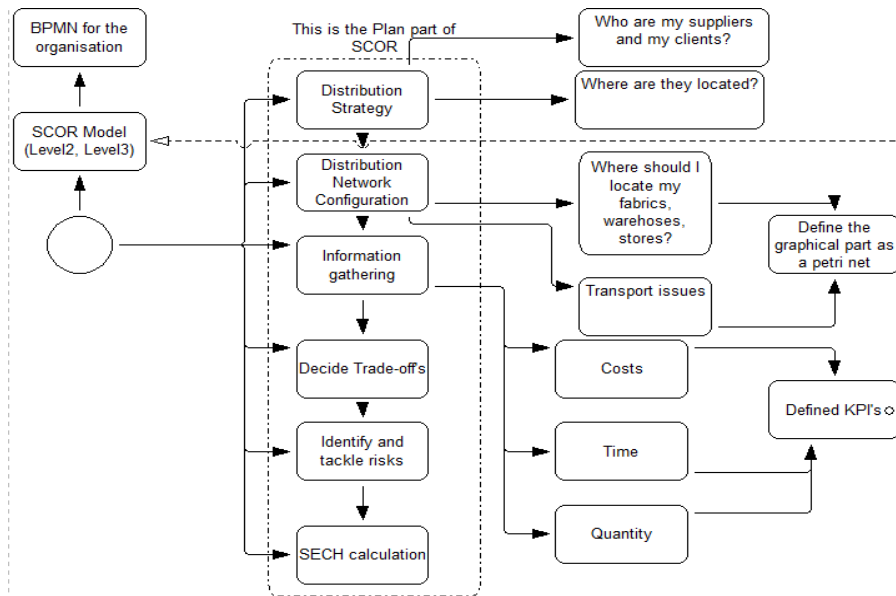


Figura 5 Cerințele aplicației de modelare SCOR

Figura 4 descrie cerințele aplicației și inter-dependința dintre acestea.

5.2 Conceptualizarea limbajului de modelare SCOR

Am început cercetarea noastră în vederea conceptualizării unui nou instrument de modelare SCOR având o abordare de tip top-down. Abordarea top-down este utilizată atunci când inginerul de limbaj pornește de la analiza domeniului, și continuă cu stabilirea contextelor, a tipurilor de modele și, în final, a claselor, relațiilor și a sintaxei folosită.

Așadar, primul pas făcut a fost conceptualizarea contextelor. Atunci când acestea sunt importate într-o platformă de meta-modelare, ele vor deveni tipuri de modele. Pentru că aceste tipuri de modele conțin notații și relații, oferim un exemplu conceptual al modului în care ar trebui să arate fiecare model propus.

5.2.1 Modelul - produs

Modelul - produs este o reprezentare ierarhică a descompunerii unui produs în componente. În literatura de specialitate acest model este cunoscut sub denumirea de listă de materiale (BOM). Pe măsură ce afacerea devine mai receptivă la gusturile consumatorilor și produsele derivate cresc pentru a satisface configurații unice, managementul BOM poate deveni imposibil de gestionat. Pentru a face față cu produse configurabile, în care schimbarea unei mici părți a unui produs poate avea efecte multiple, sunt necesare o serie de tehnici de modelare avansate [73].

Exemplu de model:

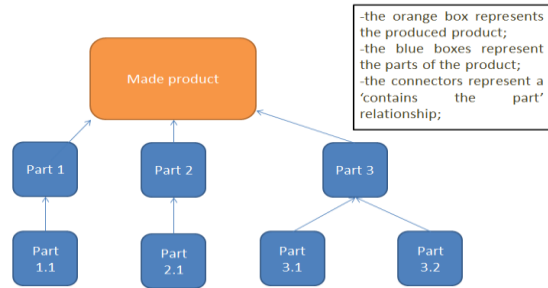


Figura 6 Modelul - prods

Pentru acest tip de model de am dezvoltat contextul Model - produs, în care am definit două notații (produs și componentă), și o relație (format din). Am folosit instrumentul de inginerie a limbajelor pentru a descrie această model iar rezultatul vizual bazat pe ML² este listat în figura 7:

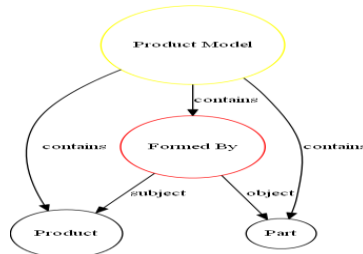


Figura 7 Model Produs - ML²

5.2.2 Modelul – domeniu de aplicare

Modelul - domeniul de aplicare este o imagine de ansamblu a mediului în care acționează o organizație. Acesta prezintă toate interacțiunile și fluxul de materiale sau informații între organizație, și partenerii ei, clienții ei și concurenții ei.

Pentru acest tip de model am creat contextul Domeniul de aplicare, în care am definit două notații (Swimlane și Entitate), și o relație (Flux de materiale/informații). Definirea atributelor și adverbilor face obiectul unei cercetări viitoare.

Modelul propus:

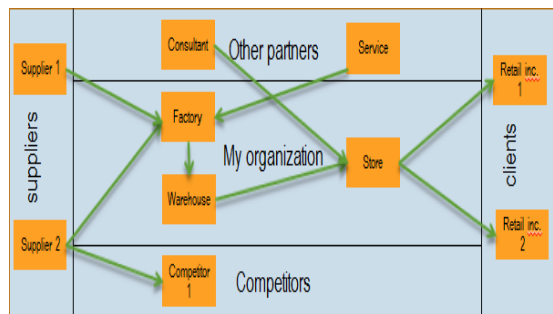


Figura 8 Model – Domeniu de aplicare

Am folosit instrumentul de inginerie a limbajelor pentru a descrie acest model iar rezultatul vizual bazat pe ML^2 este listat în figura:

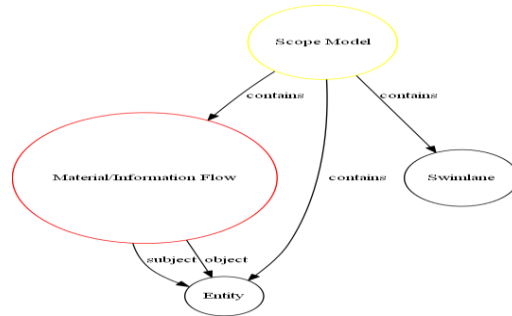


Figure 9 Model – Domeniu de aplicare ML^2

5.2.3 Modelul – regiuni de desfacere

Modelul propus:

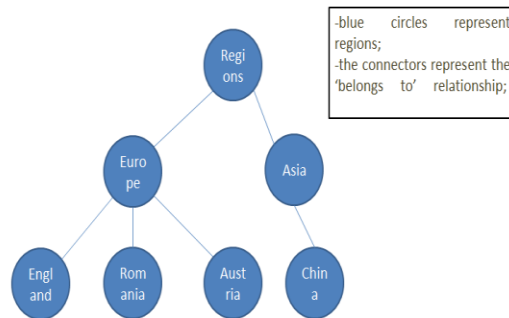


Figure 10 Modelul – regiuni de desfacere

Pentru acest model am definit contextul Regiuni de desfacere, în care am definit o notație (Regiune), și o relație (Detaliază Regiune). Am folosit instrumentul de inginerie a limbajelor pentru a descrie acest model iar rezultatul vizual bazat pe ML^2 este listat în figura:

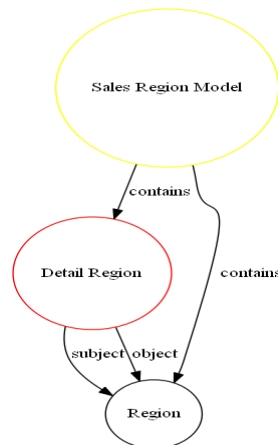


Figure 11 Modelul – regiuni de desfacere ML^2

5.2.4 Modelul – regiuni de aprovizionare

Modelul propus:

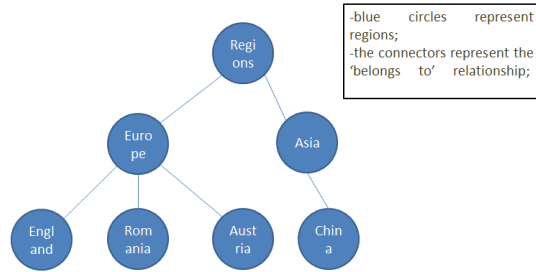


Figure 12 Modelul – regiuni de aprovizionare

Am folosit instrumentul de inginerie a limbajelor pentru a descrie acest model iar rezultatul vizual bazat pe ML^2 este listat în figura:

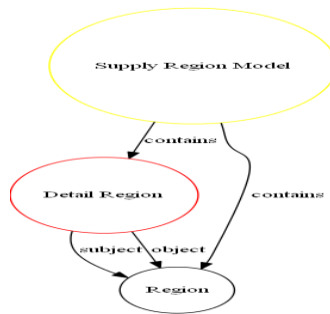


Figure 13 Modelul – regiuni de aprovizionare ML^2

5.2.5 Modelul – harta geografică

Modelul propus:

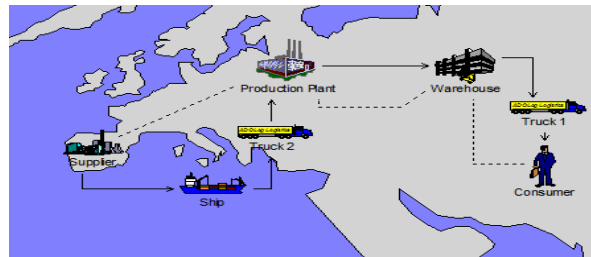


Figure 14 Modelul – harta geografică

Am folosit instrumentul de inginerie a limbajelor pentru a descrie acest model iar rezultatul vizual bazat pe ML^2 este listat în figura:

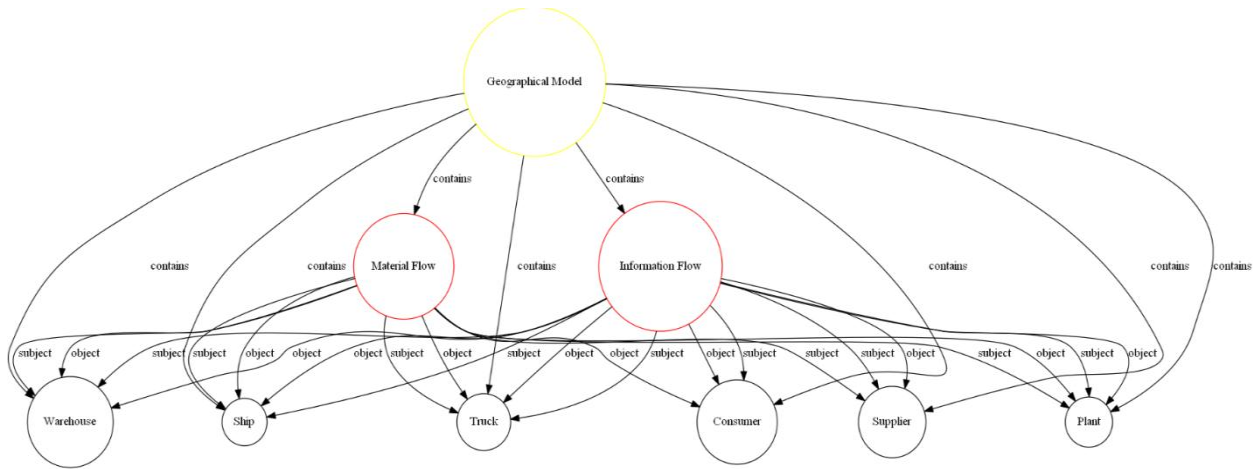


Figure 15 Modelul – harta geografică ML²

5.2.6 Modelul – detalierea procesului

Modelul propus:

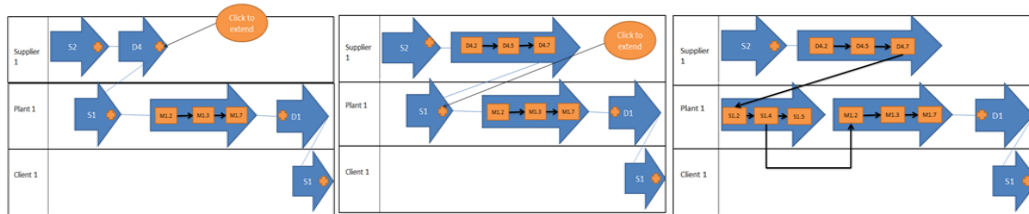


Figure 16 Modelul – detalierea procesului

Am folosit instrumentul de inginerie a limbajelor pentru a descrie acest model iar rezultatul vizual bazat pe ML² este listat în figura:

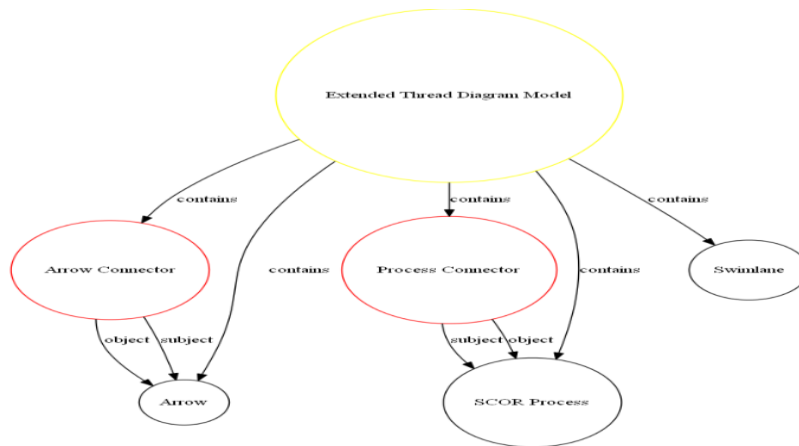


Figure 17 Modelul – detalierea procesului ML²

5.2.7 Modelul – proces de afaceri

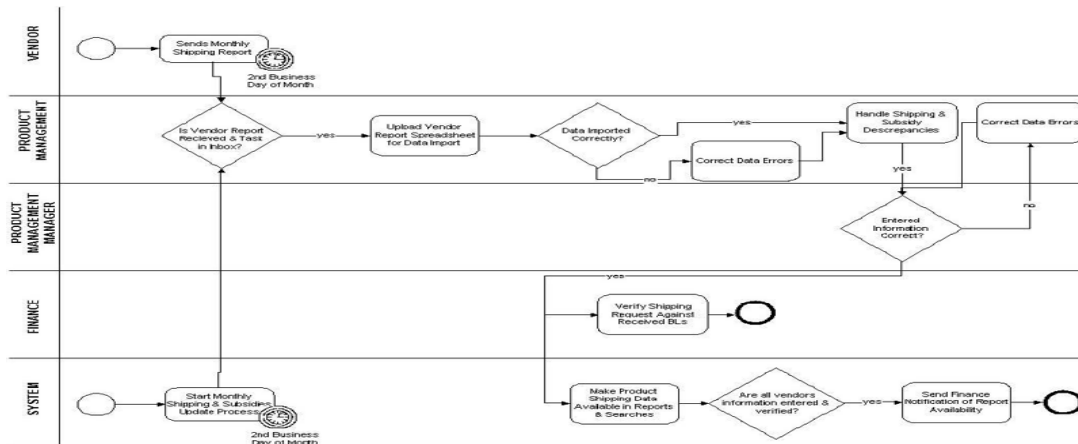


Figure 18 Modelul – proces de afaceri

Am folosit instrumentul de inginerie a limbajelor pentru a descrie acest model iar rezultatul vizual bazat pe ML^2 este listat în figura:

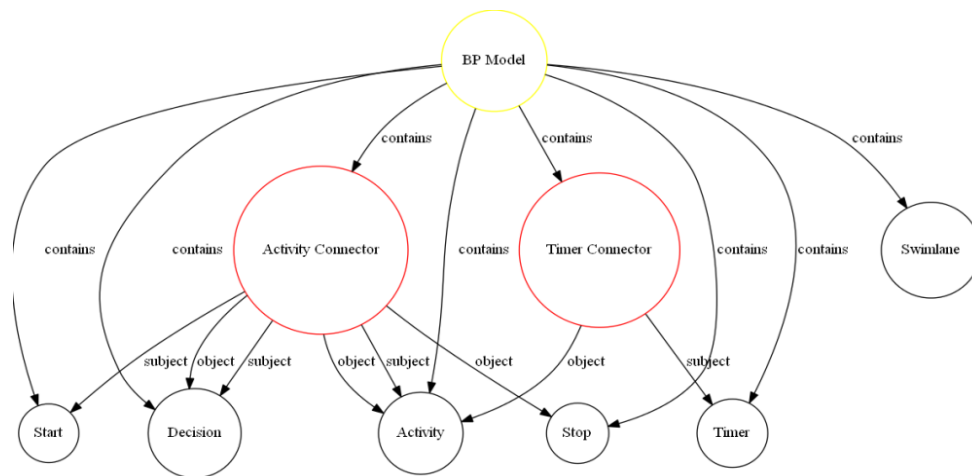


Figure 19 Modelul – Proces de afaceri ML^2

5.2.8 Modelul – model organizațional

Modelul propus:

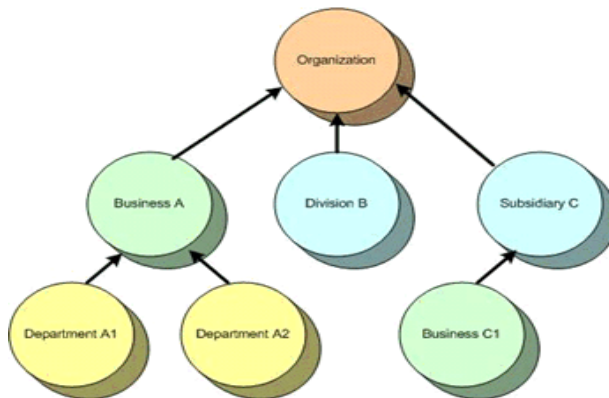


Figure 20 Modelul – model organizațional

Am folosit instrumentul de inginerie a limbajelor pentru a descrie acest model iar rezultatul vizual bazat pe ML^2 este listat în figura:

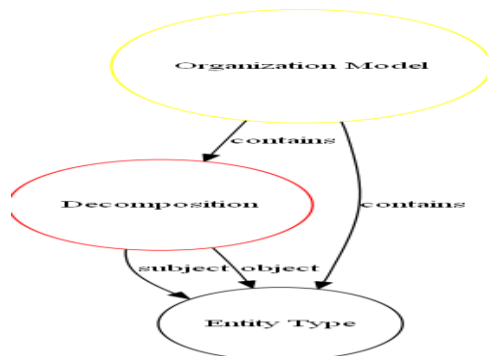


Figure 21 Modelul – Model Organizațional ML^2

5.2.9 Limbajul propus reprezentat în ML^2

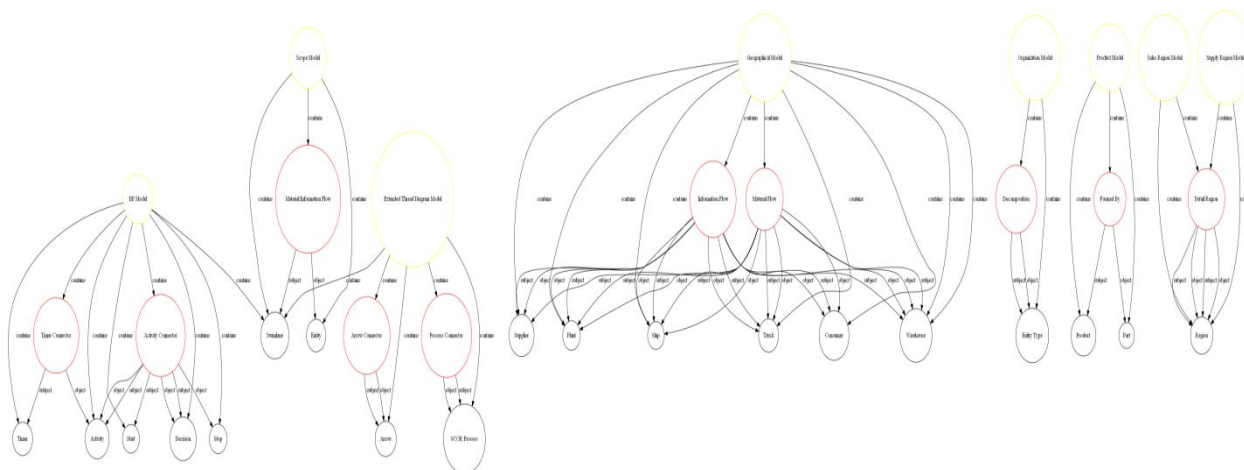


Figure 22 Limbajul propus reprezentat în ML^2

Concluzii și Dezvoltări Ulterioare

1 Contribuții

Scopul acestei cercetări a fost de a dezvolta o metodă de modelare pentru managementul lanțurilor valorice. Dar, de-a lungul perioadei de cercetare, odată cu escaladarea nivelelor de abstractizare în domenii precum managementul lanțurilor valorice, lingvistică, modelare, etc. am obținut și alte rezultate interesante care au fost prezentate și detaliate pe parcursul acestei teze. Astfel, putem spune că această teză de doctorat a prezentat contribuții în trei domenii diferite.

În primul rând, pornind de la o definiție inovativă a conceptului de limbaj, și bazându-ne cercetarea pe o ipoteză simplă care afirmă că un cuvânt este definit de un set unic de atribute, am reușit să construim o teorie care ne-a permis să înțelegem mai bine structura unui limbaj și componentele acestuia.

În al doilea rând, pe baza acestei definiții, am descris metodologia de conceptualizare a unui limbaj. În acest sens am propus o aplicație menită să ajute inginerii de limbaj să descrie noi limbaje. Aplicația permite utilizatorilor săi să conceptualizeze, să reorganizeze, să analizeze și să compare două limbaje. Scopul acestei aplicații este de a oferi o perspectivă mai clară cu privire la structura limbajelor.

În al treilea rând, lucrarea prezintă un studiu de caz în domeniul conceptualizării unei noi aplicații de modelare pentru managementul lanțurilor valorice.

Punctual, am putea spune că:

1. În **Capitolul 1** am reușit să descriem formal conceptul de limbaj. Ipoteza fundamentală a abordării noastre a fost că toate limbile cunoscute au o structură comună. Având la bază o definiție formală a conceptului de limbaj am fost capabili să propunem o metodologie de conceptualizare a limbajelor. Metodologia poate fi utilizată de către inginerii de limbaj pentru a dezvolta noi limbaje într-o manieră mai structurată.
2. În **Capitolul 2** am descris un nou limbaj de modelare al DSL-urilor, menit să servească metodologiei de conceptualizare a limbajelor. Rezultatul acestei secțiuni a fost definirea limbajului ML^2 . ML^2 se aseamănă cu MOF M3, însă, propune o sintaxă îmbogățită, cu scopul de a elimina dezavantajele folosirii MOF în vederea modelării limbajelor.
3. În **Capitolul 3** am folosit ML^2 și definiția formală a conceptului de limbaj, pentru a dezvolta o aplicație menită să ofere utilizatorilor săi toate funcționalitățile necesare

pentru ingineria limbajelor. Aplicația propune trei module: un modul de reinginerie de limbaj, un modul de inginerie de limbaj și un modul de analiză de limbaj.

4. În **Capitolul 4** am analizat teoria lanțurilor valorice, cu intenția de a conceptualiza o nouă aplicație de modelare a proceselor specific acestora. Ne-am bazat cercetarea pe evaluare și dezvoltarea limbajului SCOR. În acest sens, am comparat SCOR cu principalul său concurent VRM și am explicat de ce am ales, în final, să optăm pentru SCOR.
5. În **Capitolul 5** am conceptualizat o nouă metodă de modelare pentru lanțurile valorice folosind aplicația de ingineria a limbajelor descrisă în capitolul 3.

2 Dezvoltări ulterioare

Această teză propune abordări inovative în trei domenii diferite de cercetare: economie, informatică și lingvistică. Aspectele prezentate în această teză sunt rezultatul unei perioade de cercetare care a implicat scalarea în sus și în jos a nivelurilor de abstractizare, fapt care a obligat autorii să studieze porțiuni din toate aceste domenii. Pe parcursul acestor incursiuni în diferite domenii, noi direcții de posibile cercetări au fost gândite de autori:

1. În **domeniul economic** am putea continua cercetare prin analiza platformelor de meta-modelare care ar putea permite implementarea limbajului de modelare SCOR propus în capitolul 5. În acest sens ne-am propus să studiem cum am putea adăuga comportamente dinamice modelelor SCOR. Prin comportament dinamic înțelegem algoritmi de analiză și simulare a modelelor. Țelul principal în această cercetare ar fi identificarea unor noi algoritmi de calcul al metricilor și indicatorilor propuși de standardul SCOR.
2. În **domeniul informaticii** am putea continua cercetarea cu analiza metodei în care am putea implementa algoritmi de calcul identificați. Mai mult, am putea studia ce ar presupune dezvoltarea unor funcționalități care să permită definirea de algoritmi generici.
3. În **domeniul lingvisticii (și al inteligenței artificiale)** am putea continua cercetarea pentru a răspunde la o serie de probleme din sfera prelucrării limbajelor naturale, în speță sinonimii, traduceri, categorisiri de text, validări de text, etc.

Bibliografie

1. **Pateman, Trevor.** <http://www.selectedworks.co.uk>.
<http://www.selectedworks.co.uk/whatisalanguage.html>. [Online] 2012. [Cited: 3 20, 2012.]
<http://www.selectedworks.co.uk/whatisalanguage.html>.
2. **Noe, Alva.** <http://www.npr.org/>. <http://www.npr.org/blogs/13.7/2011/04/04/135014573/what-is-a-language-when-easy-questions-demand-tough-answers>. [Online] 2012. [Cited: 2 20, 2012.]
3. *Syntactic Structures*. **Chomsky, Noam.** Hague : the Hague: Motion, 1957.
4. *In Search of a Basic Principle for Model Driven Engineering*. **Bézivin, J.** 21-24, s.l. : Cepis UPGRADE, 2004, Vol. The European Journal for the Informatics Professional V(2).
5. *A Framework to Formalise the MDE Foundations*. **Thirioux, Xavier, et al.** Zurich : Suisse : International Workshop on Towers of Models, 2007, Vol. TOWERS 2007.
6. **Meta-Object-Facility.** http://en.wikipedia.org/wiki/Meta-Object_Facility.
<http://en.wikipedia.org>. [Online] 2012. [Cited: January 16, 2012.]
http://en.wikipedia.org/wiki/Meta-Object_Facility.
7. *"Heuristics-Based Query Processing for Large RDF Graphs Using Cloud Computing"*.
Husain M, McGlothlin J., Masud M., Khan L., Thuraisingham B.M. s.l. : Knowledge and Data Engineering, IEEE Transactions, vol. 23, pages 1312-1327,, 2011.
8. *"Heuristics-Based Query Processing for Large RDF Graphs Using Cloud Computing"*.
Husain, M., et al. s.l. : Knowledge and Data Engineering, IEEE Transactions, 2011, Vols. vol. 23, pages 1312-1327.
9. *Model-driven Language Engineering: the ASMETA case study*. **Gargantini, Angelo, Riccobene, Elvinia and Scandurra, P.** s.l. : Software Engineering Advances, 2008. The Third International Conference on Software Engineering Advances.
10. *Formal Grammars and Languages*. **Tao Jiang, Ming Li.** s.l. : Chapman & Hall/CRC ©2010, 2010. Algorithms and theory of computation handbook.
11. **umbc.** [http://www.csee.umbc.edu/portal/help/theory/lang_def.shtml]. www.csee.umbc.edu. [Online] umbc. [Cited: 9 22, 2012.]
[http://www.csee.umbc.edu/portal/help/theory/lang_def.shtml].
12. *Formal language theory for natural language processing*. **Wintner, Shuly.** Philadelphia : ACL'02 Workshop, 2002. ACL'02 Workshop.

13. *"Model-based DSL Frameworks"*. **I. Kurtev, J. B'ezivin, F. Jouault and P. Valduriez**. s.l. : Companion to the 21st Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA, 2006.
14. *Exploratory Testing*. **Kaner, Cem**. Orlando, FL : Florida Institute of Technology, 2006. Quality Assurance Institute Worldwide Annual Software Testing Conference.
15. **BOC**. www.openmodels.at . *ADOxx platform description*. [Online] BOC GmbH. [Cited: 2 12, 2012.]
16. *Towards a comparative analysis of meta-meta-models*. **Heiko Kern, Alex Hummel, Stephane Kuhne**. New York : ACM New York, 2011. SPLASH '11 Workshops Proceedings of the compilation of the co-located workshops on DSM'11, TMC'11, AGERE!'11, AOOPEs'11, NEAT'11, & VMIL'11.
17. *The Generic Modeling Environment*. **A. Ledeczi, M. Maroti, A. Bakay, G. Karsai, J. Garrett, C. Thomason, G. Nordstrom, J. Sprinkle, P. Volgyesi**. s.l. : IEEE, 2001. IEEE International Workshop on Intelligent Signal Processing.
18. **Eclipse**. <http://www.eclipse.org/modeling/emf/>. <http://www.eclipse.org/modeling/emf/>. [Online] [Cited: 3 12, 2012.] <http://www.eclipse.org/modeling/emf/>.
19. **Meta-Object-Facility**. http://en.wikipedia.org/wiki/Meta-Object_Facility. www.wikipedia.org. [Online] 2012. [Cited: 2 20, 2012.] http://en.wikipedia.org/wiki/Meta-Object_Facility.
20. *i* on ADOxx®: A Case Study*. **Margit Schwab, Dimitris Karagiannis, Alexander Bergmayr**. Hammamet, Tunisia : s.n., 2010. iStar2010 - Proceedings of the 4th International i* Workshop.
21. **OMG**. *OMG, UML 2.0 Infrastructure Specification*, www.omg.org. s.l. : OMG, 2012.
22. *"Abstract Syntax from Concrete Syntax"*. **Wile, David S**. Boston, MA, USA; : ICSE 1997, 1997.
23. *"A metamodel for the notation of graphical modeling languages"* . **Xiao He, Zhiyi Ma, Weizhong Shao, Ge Li**. s.l. : 31st Annual International Computer Software and Applications Conference (COMPSAC), 2007;.
24. **Babbie, Earl R**. *The Practice of Social Research*. s.l. : Cengage Learning, 2010. ISBN 0495598410, p. 14-18.
25. *"ARIS: Business Process Modeling"*. **Scheer, A.-W**. s.l. : Springer, 2000.

26. *"Three Factors in Language Design"*. **Chomsky, Noam**. s.l. : Linguistics Inquiry 36, no.1, 2005.
27. *"The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference"*. **Backus, J.W.** Zurich : Proceedings of the International Conference on Information Processing, UNESCO. pp. 125–132, 1959;.
28. **Heiko Kern, Alex Hummel, Stephane Kuhne**. "Towards a comparative analysis of meta-meta-models". <http://www.dsmforum.org/events/DSM11/Papers/kern.pdf>. [Online] [Cited: 4 15, 2012.] <http://www.dsmforum.org/events/DSM11/Papers/kern.pdf>.
29. *"Domain-Specific Modeling: Enabling Full Code Generation"*. **S. Kelly, J.-P. Tolvanen**. s.l. : John Wiley& Son, Inc., 2008.
30. *"Meaning as use: a functional view of semantics and pragmatics"*. **Mwihaki, Alice**. s.l. : SW AHILI Forum 11, pg. 127-139, 2004;.
31. **S. Cook, G. Jones, S. Kent, and A. C. Wills**. *"Domain Specific Development with Visual Studio DSL Tools (Microsoft .Net Development)"*. s.l. : Addison-Wesley Longman, 2007.
32. *UIMA: an architectural approach to unstructured information processing in the corporate research environment*. **David Ferrucci, Adam Lally**. DAVID FERRUCCI and ADAM LALLY (2004). UIMA: an architectural approach to unstructured information processing in the corporate research enviro10 , 2004, Vol. Natural Language Processing. pp 327-348. doi:10.1017/S1351324904003523. .
33. **graphviz**. <http://www.graphviz.org/>. <http://www.graphviz.org/>. [Online] [Cited: 09 22, 2011.] <http://www.graphviz.org/>.
34. **Glinz, Martin**. *VLML Presentation*. Vienna : s.n., 2011.
35. *On the Theoretical Foundation of Meta-Modelling in Graphically Extended BNF and First Order Logic*. **Zhu, Hong**. Taipei, Taiwan : Theoretical Aspects of Software Engineering (TASE), pages 95-104, 2010. 4th IEEE Symposium.
36. *The Design of a Conceptual Framework and Technical Infrastructure for Model Management Language Engineering*. **Richard F. Paige, Dimitrios S. Kolovos, Louis M. Rose, Nicholas Drivalos, Fiona A.C. Polack**. s.l. : 14th IEEE International Conference on Engineering of Complex Computer Systems, 2009.

37. *Model-Driven Engineering of a General Policy Modeling Language*. **Nima Kaviani, Dragan Gasevic, Milan Milanovic, Marek Hatala, Bardia Mohabbati**. s.l. : IEEE Workshop on Policies for Distributed Systems and Networks, 2008.
38. **SCC**. *Supply Chain Operations Reference Model Version 10*. s.l. : Supply Chain Council, 2011.
39. *Systemic Assessment of SCOR for Modeling Supply Chains*. **Kasi, Vijay**. 2005. Proceedings of the 36th Hawaii International Conference on System Sciences.
40. *An Introduction to the Supply Chain Council's SCOR Methodology*. **Hermon, P.** 2003. Business Process Trends.
41. **SCC**. <http://scor-software-model-framework.softsia.com/>. <http://scor-software-model-framework.softsia.com/>. [Online] [Cited: 6 12, 2012.]
42. **Supply Chain Council**. <http://supply-chain.org/f/SCOR%2090%20Overview%20Booklet.pdf>. <http://supply-chain.org>. [Online] 2009. [Cited: 9 12, 2011.]
43. *Supply Chain Management*. **Zigiaris, Sotiris**. 2000. BPR Hellas SA .
44. **Supply Chain Council**. www.supply-chain.org. www.supply-chain.org. [Online] 2009. [Cited: 7 27, 2012.]
45. *HP's global supply chain optimization*. **Becker, Mark**. Singapore : s.n., 2011. presentation, http://supply-chain.org/f/M_Bakker_SCC_SupplyChainWorld_v1.pdf.
46. **Motorrad, BMW**. <http://supply-chain.org/f/12%20-%20Stadelhofer%20-%20Innovative%20Supplier%20Management%20Systems%20at%20BMW%20Motorcycles.pdf>. www.supply-chain.org. [Online] Supply Chain Council, 10 25, 2011. [Cited: 10 22, 2012.]
47. **Brun, Stein Erland**. "Supplying consumables to offshore oil and gas plants", http://supply-chain.org/f/STATOIL_SCC.pdf. [Online] 10 11-12, 2003. [Cited: 10 25, 2012.]
48. *High Performance through Process Excellence*. **Kirchmer, M.** 2011, Springer-Verlag Berlin Heidelberg.
49. *Business process reference models: Survey and classification*. **Fettke, P., Loos, P. and Zwicker, J.** 2006, BPM 2005 Workshops, Bussler at al. (Eds.), LNCS 3812, Springer, pp. pp. 469-483.
50. **Blog**. www.supplychainadvice.wordpress.com. [Online] 01 25, 2010. [Cited: 11 20, 2010.] supplychainadvice.wordpress.com/2010/01/25/scor-vs-the-value-reference-model-frm/.

51. *Value Chains Versus Supply Chains*. **Feller A., Shunk D, Callarman T.** s.l. : <http://www.ceibs.edu/knowledge/papers/images/20060317/2847.pdf>, 2006. BPTrends.
52. *Business process reference models: Survey and classification*. **Fettke P., Loos P., Zwicker J.** s.l. : Bussler at al. (Eds.), LNCS 3812, Springer, 2006. BPW 2005 Workshops. pp. 469-483.
53. **Bolstoff P., Rosenbaum R.** *Supply Chain Excellence -3rd Edition*. s.l. : AMACOM, 2011.
54. **Value-Chain Group.** *Business process library*. <http://www.value-chain.org/business-process-library/public/>.
55. *Enterprise and process architecture patterns*. **Barros O, Julio C.** 2011, Business Process Management Journal, Vol. 17, No. 4, pp. 598-618.
56. **Value Chain Group.** *Value-Chain Group's and Education and Training* . [http://www.value-chain.org/education and training/](http://www.value-chain.org/education%20and%20training/).
57. **Harmon, P.** *Business Process Change, Second Edition: A Guide for Business Managers and BPM and Six Sigma Professionals*. s.l. : The MK/OMG Press, 2007.
58. **value chain org.** www.value-chain.org. [Online] [Cited: 02 08, 2012.] <http://www.value-chain.org/businee-process-library/public/>.
59. *IEEE Standard for Software Reviews*. s.l. : IEEE Std. 1028-1997, 1994.
60. *APICS Dictionary*. **James F. Cox, John H., Jr. Blackstone, James F. III Cox.** 2005. ISBN-10: 1558221913.
61. **Zigiaris, Sotiris.** *Supply Chain Management*. s.l. : BPR Hellas SA, 2000.
62. *Determining Criteria for Software Components: Lessons Learned*. **Juan Pablo Carvalho, Xavier Franch, Carmen Quer.** s.l. : IEEE Software, 2007.
63. **Wikipedia Org.** http://en.wikipedia.org/wiki/ISO/IEC_9126. [Online] [Cited: 10 22, 2012.] http://en.wikipedia.org/wiki/ISO/IEC_9126.
64. **ISO/IEC.** *ISO/IEC 9126-1, White-Paper First edition*. s.l. : ISO/IEC, 2001-06-15.
65. *Modeling Software Measuring Data*. **B. A. Kitchenham, R. T. Hughes, S.G. Linkman.** 9, pp.:788-804, s.l. : IEEE Trans Software Eng., 2001, Vol. 27.
66. **Tavaf, Saeedeh Jadid.** *Quality Evaluation in transformation of event logs into visual representations*. Gothenburg, Sweden : Master Book in Software Engineering and Management,, 2009. ISSN 1651-4769.
67. **Wal-Mart.** *Wal-Mart's Sustainability Index and Supply Chain Green Standards*. 2008.
68. **IDS Scheer.** *ARIS Platform Products Version: 7.02*.

69. **TIBCO**. http://developer.tibco.com/business_studio/default.jsp. [Online] [Cited: 10 24, 2012.] http://developer.tibco.com/business_studio/default.jsp.
70. **Metastorm ProVision**. <http://www.metastorm.com>. [Online] Product Overview Metastorm ProVision. [Cited: 11 22, 2011.]
71. **Nimbus**. *Business Process Management within the Value Chain, Nimbus whitepaper*; .
72. **ProcessWizard**. <http://supply-chain.org/scor-software-processwizard>. [Online] [Cited: 10 23, 2012.] <http://supply-chain.org/scor-software-processwizard>.
73. **Hvam, L.** *A procedure for building product models. Robotics and Computer-Integrated Manufacturing*. 1999. pp. 77-87.
74. *Abstract Syntax from Concrete Syntax*. **Wile, David S.** Boston, MA, USA; ; ICSE, 1997.
75. "Model-Driven Engineering". **Schmidt, D.C.** s.l. : IEEE Computer 39 , 2006, Vol. 2.
76. **Pateman, Trevor**. *Language in Mind and Language in Society*. 1987.
77. **OMG**. www.omg.org UML 2.0 Infrastructure Specification. www.omg.org. [Online] OMG. [Cited: 3 17, 2012.] www.omg.org;
78. —. *OMG, UML 2.0 OCL Specification, www.omg.org*; . s.l. : OMG, 2012.
79. *Elements of General Linguistics*. **Martinet, André**. London : (1960). Elements of General Linguistics. Tr. Elisabeth Palmer Studies in General Linguistics, 1960, Vol. 1.
80. **Lyons, John**. *Language and Linguistics*. s.l. : Cambridge University Press, 1981.
81. **Lockerby, Patrick**. http://www.science20.com/chatter_box/blog/what_language. <http://www.science20.com/>. [Online] 5 19, 2009. [Cited: 06 19, 2012.] http://www.science20.com/chatter_box/blog/what_language.
82. **Lightfoot, David**. Introduction by David W. Lightfoot. [book auth.] Noam Chomsky. *Semantic Structures*. Berlin : Walter de Gruyter GmbH & Co. KG, 1957.
83. *Wikipedia-based Semantic Interpretation*. **Gabrilovich, Evgeniy and Shaul, Markovitch**. s.l. : urnal of Arti- cial Intelligence Research, 2009, Vol. 34. 0738-4602.
84. *WordNet: An Electronic Lexical Database*. . **Fellbaum, C.** Cambridge : MIT Press, 1998.
85. *Metamodelling Platforms* . **Dimitris Karagiannis, Harald Kühn**. Dexa, Aix-en-Provence, France : In Bauknecht, K., Min Tjoa, A., Quirchmayer, G. (Eds.): Proceedings of the Third International Conference EC-Web 2002, LNCS 2455, Springer, Berlin, 2002.

86. **dictionary.reference.com**. <http://dictionary.reference.com/browse/language>.
<http://dictionary.reference.com>. [Online] [Cited: 06 19, 2012.]
<http://dictionary.reference.com/browse/language>.
87. **Dictionaries, Oxford**. <http://oxforddictionaries.com/definition/language>.
<http://oxforddictionaries.com/>. [Online] 06 19, 2012. [Cited: 06 19, 2012.]
<http://oxforddictionaries.com/definition/language>.
88. *Indexing by latent semantic analysis*. **Deerwester, S., et al.** 6, s.l. : Journal of the American Society for Information Science, 1990, Vol. 41.
89. *Lingvistica*. **Cohn, Anthony**. 1999, xxx, pp. 13-29.
90. **Bruce, Bower**. Talking back in time; prehistoric origins of language attract new data and debate - language evolution. *Science News on Bnet (Technology Industry)*. *CBS Interactive News Service*. . 29 September, 2000.
91. *Modern Information Retrieval*. **Baeza-Yates, R. and Ribeiro-Neto, B.** New York : Addison Wesley, 1999.
92. **ADONIS**. <http://www.adonis-community.com/>. <http://www.adonis-community.com/>.
[Online] BOC, 2011. [Cited: 11 22, 2011.] <http://www.adonis-community.com/>.
93. *Joint Use of SCOR and VRM*. **Costin, Răzvan Aurelian, Kirikova, Marite and Buchmann, Robert**. 2012, in Perspectives in business informatics research : 11th international conference, BIR 2012 Proceedings, Nizhny Novgorod, Russia, ed. Springer, ISBN: 3-642-33280-3,.
94. **value-chain.org**. www.value-chain.org/. [Online] 10 08, 2012. [Cited: 10 08, 2012.]
www.value-chain.org/bptf/buildingblocks/vrm/.
95. *Value Chains, Value Streams, Value Nets, and Value Delivery Chains*. **W., Brown G.** s.l. :
<http://www.bptrends.com/publicationfiles/FOUR%2004-009-ART-Value%20Chains-Brown.pdf>,
2009. BPTrends.

Declarații:

96. **Costin Aurelian Răzvan**, Țolea Eniko. „E-collaboration - The new economic world”, in *Journal of Information Systems & Operations Management*, Vol. 4, Nr. 1, pg. 142-149, ISSN 1843-4711, CNCSIS B+, Mai 2010;
97. **Costin Aurelian Răzvan**, „E-collaborative system for management of virtual organizations”, in *International Conference Managerial Challenges of the Contemporary Society Proceedings*, Cluj-Napoca, 4-5 June, 2010;

98. **Costin Aurelian Răzvan**, Țolea Eniko, „Deployment features for managing virtual organizations” in *Informatica Economica Journal*, September 2010;
99. **Costin Aurelian Răzvan**, Țolea Eniko, „Ontology for an E-learning model”. In *ICVL 2010: Proceedings of The 5th International Conference on Virtual Learning*, ISSN 1844-8933, Nr. 2689/2010, Editura Universității din București, pg. 135;
100. **Costin Aurelian Răzvan**, Rusu Lucia, Simona Kleinhempel, Sergiu Jecan, „A Collaborative Model For Virtual Enterprise” in *Journal of Information Systems & Operations Management*, Vol. 4, Nr. 2, pg. 20-32, ISSN 1843-4711, CNCSIS B+, December 2010;
101. **Costin Aurelian Răzvan**, Lucia Rusu, Marius Podean, Szasz Szabolcs, „A Hierarchical Model For Medical Registrations” in *Journal of Information Systems & Operations Management*, Vol. 4, Nr. 2, pg. 20-32, ISSN 1843-4711, CNCSIS B+, December 2010;
102. **Costin Aurelian Răzvan**, Țolea Eniko, „The Design of a Distributed Database for Doctoral Studies Management”. In *Informatica Economica Journal*, Vol. 14, No. 4/2010, pg. 139-146, ISSN 1453-1305, CNCSIS B+, December 2010;
103. **Costin Aurelian Răzvan**, Țolea Eniko, „Software Development Business Moving towards a Unified Collaborative System in IT. SaaS may be the New Orientation.”. *Analele Universității din Oradea*. TOM XIX nr. 2, pg. 398-404, ISSN 1582-5450, CNCSIS B+, December 2010;
104. **Costin Aurelian Răzvan**, Țolea Eniko, Marius Podean, „Integrating SaaS Modules Using XML Technologies for Collaborative Project”, in *CEE Symposium 2011 Proceedings*, 28-30 April, 2011;
105. **Costin Aurelian Răzvan**, Jecan Sergiu, Țăranu Mirela, „Integrated Management System in College Admission”, in *IECS Proceedings*, nr. 18, Thomson Reuters-ISI Proceedings, 19 Mai 2011;
106. **Costin Aurelian Răzvan**, Marius Podean, Dan Bența, „On Supporting Creative Interaction in Collaborative Systems: A Content Oriented Approach”, in *IEEE 13th Conference on Commerce and Enterprise Computing Proceedings*, ISBN: 978-0-7695-4535-6, <http://doi.ieeecomputersociety.org/10.1109/CEC.2011.65>, pg. 400-406, October 2011;
107. **Costin Aurelian Răzvan**, Lucia Rusu, Țolea Eniko, Sergiu Jecan, „Integrated Platform for Virtual Enterprise”, in *TELFOR proceedings*, ISBN:9781457714986, Publisher: Institute of Electrical and Electronics Engineers (IEEE), November 2011;

108. **Costin Aurelian Răzvan**, Țolea Eniko, „Collaborative E-Learning Model”, in Proceedings of the 6th International Conference on E-Learning ICVL, pg. 248-253, October 2011;
109. **Costin Aurelian Răzvan**, Lucia Rusu, Simona Marțiș, Marius Podea, „An Ehr Document Model of Chronic Patients” in Journal of Information Systems & Operations Management, ISSN: 1843-4711, nr. 5, pg. 264-275, December 2011;
109. **Costin Aurelian Răzvan**, Jecan Sergiu, Maria Iulia Laba, „Document Management in Context of Collaborative Systems”, in Journal of Information Systems & Operations Management, ISSN: 1843-4711, nr. 5, pg. 308-317, December 2011;
110. **Costin Aurelian Răzvan**, „Using blackboard-like software for coupling SaaS”, in Procedia – Computer Science Journal, ISSN: 1877-0509, Mai 2012;
111. **Costin Aurelian Răzvan**, Marite Kirikova, Robert Buchmann, „Joint Use of SCOR and VRM”, in Perspectives in business informatics research : 11th international conference, BIR 2012 Proceedings, Nizhny Novgorod, Russia, ed. Springer, ISBN: 3-642-33280-3, 978-3-642-33280-7, September 2012;
112. **Costin Aurelian Răzvan**, „Dsl Reengineering Tool Development”, in The 12th International Conference on Informatics in Economy Proceedings, Bucharest, Romania, April 2013;
113. **Costin Aurelian Răzvan**, „New Modeling Language for Business DSLs”, in AWERProcedia Information Technology & Computer Science, accepted for publication, Antalya, Turkey, April 2013.