Universitatea „Dunărea de Jos” din Galați
Școala doctorală de Inginerie

TEZĂ DE DOCTORAT

CONTRIBUȚII LA REALIZAREA UNUI SUBSISTEM AL CUNOȘTINȚELOR ASUPRA TEHNICILOR ȘI STRATEGIILOR PEDAGOGICE ÎNTR-UN SISTEM DE INSTRUIRE ASISTATĂ DE CALCULATOR

Doctorand,
BULGARU DIANA (ȘTEFĂNESCU)

Conducător științific,
Prof. Univ. Dr. Ing. DUMITRIU LUMINIȚA

Seria I2: Calculatoare și tehnologia informației Nr. 5

GALAŢI
2015

Universitatea „Dunărea de Jos” din Galați
Școala doctorală de Inginerie

TEZĂ DE DOCTORAT

CONTRIBUȚII LA REALIZAREA UNUI SUBSISTEM AL CUNOȘTINȚELOR ASUPRA TEHNICILOR ȘI STRATEGIILOR PEDAGOGICE ÎNTR-UN SISTEM DE INSTRUIRE ASISTATĂ DE CALCULATOR

Doctorand,
BULGARU DIANA (ȘTEFĂNESCU)

Conducător științific, Prof. Univ. Dr. Ing. Dumitriu Luminița

Referenți științifici Prof. Univ. Dr. Ing. Trăușan-Matu Ștefan
Prof. Univ. Dr. Ing. Bădică Cosmin
Conf. Univ. Dr. Ing. Tudorie Cornelia

Seria I2: Calculatoare și tehnologia informației Nr. 5

GALAŢI
2015

Seriile tezelor de doctorat susținute public în UDJG începând cu 1 octombrie 2013 sunt:

Domeniul **ȘTIINȚE INGINEREȘTI**

Seria I 1**: Biotehnologii**
Seria I 2**: Calculatoare și tehnologia informației**
Seria I 3**. Inginerie electrică**
Seria I 4**: Inginerie industrială**
Seria I 5: **Ingineria materialelor**
Seria I 6**: Inginerie mecanică**
Seria I 7**: Ingineria produselor alimentare**
Seria I 8: **Ingineria sistemelor**

Domeniul **ȘTIINȚE ECONOMICE**

Seria E 1: **Economie**
Seria E 2: **Management**

Domeniul **ȘTIINȚE UMANISTE**

Seria U 1: **Filologie- Engleză**
Seria U 2: **Filologie- Română**
Seria U 3**: Istorie**

"Laissez-moi commencer par un mot que tous les hommes, depuis que l'homme existe, ont prononcé: ***merci.****"*

"Permiteti-mi, vă rog, să încep cu un cuvânt pe care orice om, de când omenirea există, l-a pronunțat: ***mulțumesc.****"*

(Octavio PAZ, discursul de la Stockholm, "La quête du présent")

CUVÂNT ÎNAINTE

Într-un moment ca acesta, este greu sa găsesc cuvintele potrivite pentru a mulțumi tuturor celor care mi-au oferit sprijinul, înțelegerea, toleranța, încurajările și ajutorul fără de care nu aș fi putut finaliza această teză.

Mai întâi, îi mulțumesc lui Dumnezeu care mi-a dat puterea să realizez acest demers, mi-a dăruit o familie minunată și a "rânduit" lucrurile astfel încât să fiu înconjurată de prieteni și să întâlnesc oamenii de care aveam nevoie, atunci când aveam nevoie de aceștia.

Primul meu gând de se îndreaptă către D-nul Prof. Dr. Ing. Severin Bumbaru, mentorul nostru pe calea cunoașterii.

Îi mulțumesc îndrumătorului meu, D-nei Prof. Dr. Ing. Luminița Dumitriu, pentru încrederea acordată, pentru discuțiile și sfaturile pertinente, de un înalt profesionalism.

Le mulțumesc membrilor Comisiei pentru efortul Domniilor lor.

Le mulțumesc prietenilor și colegilor din catedră.

Și nu în ultimul rând, mulțumesc părinților mei, fiului și soțului, care m-au sprijinit necondiționat și m-au încurajat pe parcursul acestor ani.

GALAȚI, Diana Bulgaru (Ștefănescu)

2015

CUPRINS

1. STADIUL ACTUAL AL CERCETĂRILOR ÎN DOMENIUL INSTRUIRII ASISTATE DE CALCULATOR 1
1.1 INSTRUIREA ASISTATĂ DE CALCULATOR 1
1.1.1 NOȚIUNI PRELIMINARE 1
1.1.2 ABORDĂRI ȘI TEORII CU INFLUENȚĂ ASUPRA PROIECTĂRII PEDAGOGICE 3
1.2 SISTEME DE INSTRUIRE ASISTATĂ DE CALCULATOR 12
1.2.1 EVOLUȚIE ȘI CARACTERISTICI 12
1.2.2 ARHITECTURA MODULARĂ A UNUI SISTEM INTELIGENT DE INSTRUIRE 17
1.2.3 ALTE ASPECTE ALE SISTEMELOR EDUCAȚIONALE 22
1.3 METODE ȘI INSTRUMENTE PENTRU PROIECTAREA SISTEMELOR DE INSTRUIRE 26
1.3.1 SISTEME/MEDII DE DEZVOLTARE DE TIP AUTOR 26
1.3.2 STANDARDE ÎN INSTRUIREA ASISTATĂ 27
1.3.3 LIMBAJE DE MODELARE EDUCAȚIONALĂ 31
1.3.4 METODA MISA ȘI INSTRUMENTE 32
1.4 ONTOLOGIILE – TEORII CADRU PENTRU DEZVOLTAREA SISTEMELOR INTELIGENTE DE INSTRUIRE 32
1.4.1 ONTOLOGII 32
1.4.2 DEFINIREA TERMENULUI DE ONTOLOGIE 33
1.4.3 CLASIFICĂRI ALE ONTOLOGIILOR 35
1.4.4 ONTOLOGIA TASK-URILOR ŞI A METODELOR 39
1.4.5 ROLUL ONTOLOGIILOR ÎN SISTEMELE EDUCAŢIONALE 43
1.5 CONCLUZII 49

2. CONTRIBUŢII METODOLOGICE LA MODELAREA ONTOLOGICĂ A CUNOAȘTERII 51
2.1 STUDIU PRELIMINAR 51
2.1.1 ONTOLOGII ŞI REPREZENTAREA CUNOŞTINŢELOR 51
2.1.2 INGINERIA ONTOLOGICĂ 56
2.2 OBIECTIVE 60
2.3 CONTRIBUȚII PRIVIND O METODĂ INTEGRATĂ DE INGINERIE ONTOLOGICĂ (MIIO) 62
2.4 CONCLUZII ŞI CONTRIBUŢII 65

3. CONTRIBUŢII PRIVIND FORMALISMUL MODELĂRII CONCEPTUALE A DOMENIULUI 67
3.1 OBIECTIVE 67
3.2 CONTRIBUȚII TEORETICE PRIVIND LIMBAJUL DE REPREZENTARE OntL_CG 70
3.2.1 OntL_CG – BAZAT PE PARADIGMA ENTITATE-RELAȚIE 71
3.2.2 LIMBAJUL OntL_CG 71
3.2.3 SEMANTICA FORMALĂ A LIMBAJULUI OntL_CG 74
3.3 CONTRIBUȚII TEORETICE PRIVIND ONTOLOGIILE LIMBAJULUI DE REPREZENTARE OntL_CG 75
3.4 CONTRIBUȚII TEORETICE PRIVIND O METODĂ DE OPERAȚIONALIZARE A UNEI ONTOLOGII A DOMENIULUI 77
3.5 OPERAȚIONALIZAREA BAZATĂ PE MODELUL GRAFURILOR CONCEPTUALE 82
3.6 CONTRIBUȚII APLICATIVE – Sistemul OntL_CG 87
3.6.1 CAZURILE DE UTILIZARE 87
3.6.2 PREZENTAREA GENERALĂ A APLICAȚIEI 88

3.6.3 STRUCTURA APLICAȚIEI CLIENT OntL_CG ... 92
3.7 STUDIU DE CAZ ... 115
3.8 CONCLUZII ŞI CONTRIBUŢII ... 118

4. CONTRIBUŢII LA MODELAREA STRATEGIILOR DE INSTRUIRE ADAPTIVE ... 119
4.1 PROPUNEREA UNUI CADRU STRUCTURAT DE MODELARE A CUNOAŞTERII PEDAGOGICE ... 119
4.1.1 TERMENI DE DESCRIERE A CUNOAŞTERII PEDAGOGICE ... 119
4.1.2 STRUCTURAREA CUNOAȘTERII PEDAGOGICE ... 122
4.2 APLICAREA METODEI MIIO PENTRU ONTOLOGIA PROPUSĂ ... 127
4.3 CONTRIBUȚII LA MODELAREA CONCEPTUALĂ A STRATEGIILOR DE INSTRUIRE ADAPTIVE ... 128
4.4 CONTRIBUȚII LA MODELAREA FORMALĂ A STRATEGIILOR DE INSTRUIRE ADAPTIVE 129
4.5 CONCLUZII ȘI CONTRIBUȚII ... 134

5. CONCLUZII, CONTRIBUȚII ȘI DIRECȚII VIITOARE DE CERCETARE ... 135

6. LISTA LUCRĂRILOR PUBLICATE ȘI PREZENTATE ... 139

7. REFERINŢE BIBLIOGRAFICE ... 153

I. ANEXA I – STUDIU DETALIAT AL ELEMENTELOR INGINERIEI ONTOLOGICE ... 1
AI.1. OBIECTIVELE INGINERIEI ONTOLOGICE ... 1
AI.2. MODELE ŞI LIMBAJE DE REPREZENTARE A CUNOŞTINŢELOR UTILIZATE ÎN INGINERIA ONTOLGICĂ ... 1
AI.2.1 Modele bazate pe cadre ... 2
AI.2.2 Modele bazate pe logici de descriere ... 3
AI.2.3 Modele bazate pe grafuri conceptuale ... 3
AI.2.4 Limbaje de reprezentare a ontologiilor pentru web ... 10
AI.3. METODOLOGII, METODE ŞI INSTRUMENTE ÎN INGINERIA ONTOLOGICĂ ... 14
AI.3.1 Metodologii, metode şi instrumente pentru construirea ontologiilor ... 15
AI.3.2 Metodologii, metode şi instrumente pentru evaluarea ontologiilor ... 20
AI.3.3 Alte metodologii, metode şi instrumente utilizate în ingineria ontologică ... 22
AI.4. PRINCIPII DE BAZĂ ÎN INGINERIA ONTOLOGICĂ ... 25

II. ANEXA II - APLICAȚIA OntL_CG ... 1

III. ANEXA III – FORMALISMUL GRAFURILOR CONCEPTUALE FOLOSIT ÎN LUCRARE ... 1
AIII.1. DEFINIREA TERMENILOR FOLOSIȚI ÎN LUCRARE ... 1
AIII.2. JUSTIFICAREA ALEGERII FORMALISMULUI ... 2
AIII.1. SISTEMUL FORMAL AL GRAFURILOR CONCEPTUALE ... 5
AIII.3.1. Definirea formală a suportului și a grafurilor conceptuale ... 5
AIII.3.2. Raționamentul în limbajul de reprezentare a grafurilor conceptuale ... 10
AIII.4. FAMILIA SCG ... 16
AIII.4.1 λ-abstracțiuni ... 16
AIII.4.2 Reguli și constrângeri ... 16
AIII.4.4 Complexitate ... 18

CONTENTS

1. STATE OF THE ART 1
1.1 COMPUTER ASSISTED EDUCATION 1
1.1.1 INTRODUCTION 1
1.1.2 APPROACHES AND THEORIES WITH IMPACT ON PEDAGOGICAL DESIGN 3
1.2 COMPUTER ASSISTED INSTRUCTION SYSTEMS 12
1.2.1 EVOLUTION AND CHARACTERISTICS 12
1.2.2 MODULE-BASED ARCHITECTURE 17
1.2.3 OTHER ASPECTS REGARDING EDUCATIONAL SYSTEMS 22
1.3 METHODS AND TOOLS FOR INSTRUCTIONAL SYSTEMS DESIGN 26
1.3.1 AUTHORING SYSTEMS 26
1.3.2 STANDARDS IN COMPUTER ASSISTED INSTRUCTION 27
1.3.3 EDUCATIONAL MODELING LANGUAGES 31
1.3.4 MISA METHOD AND TOOLS 32
1.4 ONTOLOGIES – FRAMEWORK THEORIES FOR INTELLIGENT EDUCATIONAL SYSTEMS DEVELOPMENT 32
1.4.1 ONTOLOGIES 32
1.4.2 COMPREHENSIVE DEFINITIONS OF ONTOLOGY 33
1.4.3 CLASSIFICATIONS OF ONTOLOGIES 35
1.4.4 TASK-METHOD ONTOLOGIES 39
1.4.5 ROLES OF ONTOLOGIES IN EDUCATIONAL SYSTEMS 43
1.5 CONCLUSIONS I 49

2. METHODOLOGICAL CONTRIBUTIONS TO KNOWLEDGE ONTOLOGICAL MODELING 51
2.1 PRELIMINARY STUDY 51
2.1.1 ONTOLOGIES AND KNOWLEDGE REPRESENTATION 51
2.1.2 ONTOLOGICAL ENGINEERING 56
2.2 OBJECTIVES 60
2.3 CONTRIBUTIONS TO AN INTEGRATED METHOD FOR ONTOLOGICAL ENGINEERING (MIIO) 62
2.4 CONCLUSIONS AND CONTRIBUTIONS 65

3. CONTRIBUTIONS TO THE FORMALISM FOR DOMAIN CONCEPTUAL MODELING 67
3.1 OBJECTIVE 67
3.2 THEORETICAL CONTRIBUTIONS TO THE REPRESENTATION LANGUAGE OntL_CG 70
3.2.1 OntL_CG – BASED ON ENTITY-RELATION PARADIGM 71
3.2.2 OntL_CG LANGUAGE 71
3.2.3 FORMAL SEMANTICS OF THE OntL_CG LANGUAGE 74
3.3 THEORETICAL CONTRIBUTIONS TO ONTOLOGIES IN OntL_CG REPRESENTATION LANGUAGE 75
3.4 THEORETICAL CONTRIBUTION REGARDING AN OPERATIONALIZATION METHOD OF A DOMAIN ONTOLOGY 77
3.5 OPERATIONALISATION BASED ON CONCEPTUAL GRAPH MODEL 82
3.6 APPLICATIVE CONTRIBUTIONS – OntL_CG SOFTWARE APPPLICATION 87
3.6.1 UML USE CASES 87
3.6.2 GENERAL VIEW OF THE APPLICATION 88

3.6.3 STRUCTURE OF THE CLIENT-APPLICATION OntL_CG ... 92
3.7 CASE STUDY ... 115
3.8 CONCLUSIONS AND CONTRIBUTIONS ... 118

4. CONTRIBUTIONS TO ADAPTIVE PEDAGOGICAL STRATEGIES MODELING ... 119
4.1 PURPOSE FOR A STRUCTURED FRAMEWORK FOR PEDAGOGICAL KNOWLEDGE MODELING ... 119
4.1.1 TERMS FOR PEDAGOGICAL KNOWLEDGE DESCRIPTION ... 119
4.1.2 STRUCTURING THE PEDAGOGICAL KNOWLEDGE ... 122
4.2 USING MIIO METHOD ... 127
4.3 CONSTRIBUTIONS TO CONCEPTUAL MODELING OF THE ADAPTIVE PEDAGOGICAL STRATEGIES ... 128
4.4 CONTRIBUTIONS TO FORMAL MODELING OF THE ADAPTIVE PEDAGOGICAL STRATEGIES ... 129
4.5 CONCLUSIONS AND CONTRIBUTIONS ... 134

5. CONCLUSIONS, CONTRIBUTIONS AND FUTURE RESEARCH ... 135

6. LIST OF PUBLISHED AND PRESENTED PAPERS ... 139

7. BIBLIOGRAPHY ... 153

I. APPENDIX I – LARGE STUDY ON ONTOLOGICAL ENGINEERING ELEMENTS ... 1
AI.1. OBJECTIVES OF ONTOLOGICAL ENGINEERING ... 1
AI.2. MODELS AND LANGUAGES FOR KNOWLEDGE REPRESENTATION IN ONTOLOGICAL ENGINEERING ... 1
AI.2.1 Frame-based models ... 2
AI.2.2 Description logic models ... 3
AI.2.3 Conceptual graphs based models ... 3
AI.2.4 Web markup languages for ontologies ... 10
AI.3. METHODOLOGIES, METHODS AND TOOLS IN ONTOLOGICAL ENGINEERING ... 14
AI.3.1 Methodologies, methods and tools for building ontologies ... 15
AI.3.2 Methodologies, methods and tools for ontologies evaluaton ... 20
AI.3.3 Others methodologies, methods and tools used in ontological engineering ... 22
AI.4. GUIDING PRINCIPLES FOR ONTOLOGICAL ENGINEERING ... 25

II. APPENDIX II - OntL_CG APPLICATION ... 1

III. APPENDIX III – THE CONCEPTUAL GRAPHS FORMALISM USED IN THIS WORK ... 1
AIII.1. TERMS ... 1
AIII.2. ARGUMENTS FOR CHOOSING THE CONCEPTUAL GRAPHS FORMALISM ... 2
AIII.1. THE FORMAL SYSTEM OF CONCEPTUAL GRAPHS ... 5
AIII.3.1. Formal definitions for support and conceptual graphs ... 5
AIII.3.2. Reasoning in CG representation language ... 10
AIII.4. THE SCG (Simple Conceptual Graph) FAMILY ... 16
AIII.4.1 λ-abstractions ... 16
AIII.4.2 Rules and constraints ... 16
AIII.4.4 Computational complexity ... 18

INTRODUCERE

Lucrarea de faţă se înscrie pe direcția studiilor şi cercetărilor din domeniul instruirii asistate de calculator. Acest domeniu, apărut ca punte de legătură între ştiinţele educaţiei şi ştiinţele inginereşti (informatică, inteligenţă artificială) îşi propune să dezvolte şi să integreze în procesele de învăţare *sisteme de instruire care folosesc calculatorul*. Preocupările intense din domeniul instruirii asistate, cât şi cele din domeniile colaterale (filosofie, psihologie, medicină, ştiinţele educaţiei, sociologie) au condus la rezultate care demonstrează eficacitatea acestor sisteme în procesul instruirii şi validează numeroase teorii.

Sistemele software de instruire asistată de calculator - mai mult sau mai puţin "inteligente", cu rol complementar sau "înlocuitor" (măcar sub diferite aspecte) al profesorului uman, inflexibile sau capabile să se adapteze ritmului de lucru individual al elevului şi nivelului său de cunoaştere - încearcă să ofere soluţii pentru eficientizarea procesului de instruire.

Astfel, parafrazându-l pe Einstein, *întrebările* - "Ce este cunoaşterea şi care sunt sursele ei?", "Ce este învăţarea?", "Ce strategie pedagogică este indicată într-un anumit context al instruirii, pentru a obţine rezultate cât mai bune?" - *sunt aceleaşi ca acum câteva sute de ani, dar răspunsurile la aceste întrebări sunt răspunsuri noi.*

Propunerile, ipotezele, modelele teoretice și aplicative din lucrare se înscriu în încercarea de a oferi soluții acestor probleme.

Contextul cercetării:

Domeniul de aplicare (de interes): domeniul educaţiei, mai precis, educaţia mediată prin sisteme software inteligente (AIED).

Domenii de investigare: ingineria ontologică, ingineria cunoştinţelor.

Cuvinte cheie: inteligență artificială în educație, reprezentarea cunoștințelor (pedagogice), inginerie ontologică, grafuri conceptuale, sisteme autor, sisteme inteligente de instruire.

Structura lucrării

Lucrarea este structurată pe 5 capitole și conține 3 anexe.

Primul capitol prezintă stadiul actual al cercetărilor în domeniu, definind contextul în care vor fi tratate problemele propuse. Se evidențiază în principal aspectele pedagogice care trebuie luate în considerare la proiectarea sistemelor inteligente de instruire precum și rolul ontologiilor în dezvoltarea acestor sisteme.

Capitolul 2 (”*Contribuții la modelarea ontologică a cunoașterii*”) debutează cu un studiu preliminar care cuprinde abordările formale existente pentru conceptualizare, ontologie și angajament ontologic; sunt puse în evidență aspecte importante ale ingineriei ontologice care servesc obiectivelor propuse prin tematica abordată (un studiu mai detaliat se găsește în *Anexa I* a lucrării). Concluziile acestei analize au condus la propunerea unei metode integrate de inginerie ontologică (MIIO).

Capitolul 3 (*“Contribuții privind formalismul modelării conceptuale a domeniului”*) al lucrării are ca punct de plecare considerațiile anterioare, referitoare la modele, limbaje și instrumente de reprezentare a cunoașterii ontologice. Se propune şi se validează teoretic şi experimental un limbaj, o metodă şi un instrument de creare şi operaţionalizare a unei ontologii puternic formalizate; ea va putea constitui o bază de cunoștințe în sistemele de instruire, asigurând într-o mare măsură independența față de funcționalitatea acestora. Studiul de caz prezentat susține elementele teoretice și aplicative propuse, ilustrând cum pot fi utilizate în conceperea unei strategii pedagogice.

Capitolul 4 (“*Contribuții la modelarea formală a strategiilor de instruire adaptive*”) abordează aspecte ale modelării conceptuale și formale ale strategiilor de instruire adaptive. Modelele propuse pot fi implementate într-un sistem autor de dezvoltare.

Capitolul 5, care încheie lucrarea, expune concluziile, oferă o sinteză a contribuțiilor aduse și punctează direcțiile viitoare de cercetare.

Anexa I (*“Studiul detaliat al elementelor ingineriei ontologice”*) conține un studiu mai amplu asupra ingineriei ontologice, cu aplicabilitate în sisteme moderne bazate pe cunoștințe, într-o mare diversitate de domenii; aspectele cele mai importante, cu impact asupra obiectivelor concrete ale lucrării, au constituit baza contribuțiilor aduse prin capitolele 2 și 3.

Anexa II (“*Aplicația OntL_CG*”) – prezintă aplicația propusă în capitolul 3, secțiunea 3.6.

Anexa III (“*Formalismul grafurilor conceptuale folosit în lucrare*“) prezintă modelul formal folosit pentru reprezentarea cunoștințelor, bazat pe grafuri conceptuale simple, extinse cu reguli și constrângeri.

INTRODUCTION

This thesis aligns with the studies and research in the field of computer assisted instruction. This field, born to bridge the gap between the educational sciences and the engineering ones (informatics, artificial intelligence) has the aim of developing and integrating the computer-based systems of instruction into the teaching - learning process. The intense preoccupations in the computer assisted instruction as well as the ones in the side fields (philosophy, psychology, medicine, the educational sciences, sociology) have led to results which demonstrate the efficiency of these systems in the instruction process and gives validity to a lot of theories.

The computer assisted instruction systems- "more or less intelligent", complementary or replacements to (at least in a few aspects) the human teacher, inflexible or capable of adapting themselves to the student's individual working rhythm and to his level of knowledge - are trying to offer solutions in order to make the instruction process effective.

Thus, paraphrasing Einstein , the questions – "*What is knowledge and what are its sources?*", "*What is learning?*", "*What pedagogical strategy is recommended in a certain context of instruction in order to get the best possible results*?"- are the same now as they were several hundred of years ago, but the answers to these questions are new now.

The proposals, hypotheses, the theoretical and practical patterns are trying to find solutions to this query.

The context of the research:

The domain of application (of interest):the domain of education, more specifically, Artificial Intelligence in Education(AIED).

The domains of investigation: Ontological Engineering, Knowledge Engineering.

Key words: artificial intelligence in education, (pedagogical) knowledge representation, ontological engineering, conceptual graphs, authoring systems, intelligent instructional systems.

The Structure of the Thesis

The thesis consists of five chapters and three appendices.

Chapter 1 presents the current stage in the research in this field, outlining the context in which the proposed topics will be treated. Emphasis is laid on the main pedagogical aspects which

should be taken into account when the intelligent systems are modelled as well as the role of the ontologies in developing these systems.

Chapter 2 ("*Contributions to knowledge ontological modelling*") starts with a preliminary study containing the existing formal approaches for conceptualization, ontology and ontological commitment; the important aspects of ontological engineering are stressed, aspects which lead to the objectives proposed through the topics approached (a detailed study can be found in Appendix I of the paper). The conclusions of this analysis lead to the proposal of a integrative ontological engineering (MIIO).

Chapter 3 ("*Contributions regarding the conceptual modeling in the domain*") of the paper has its starting point in the previous considerations, which refer to the models, languages and tools for ontological knowledge representation. We recommend and validate theoretically and experimentally a language, a method and a tool of creating and functionalizing heavy-weight ontologies; it will become a knowledge base in the instruction systems, ensuring to a greater degree, its independence on their functionality. The case study presented maintains the theoretical and applied elements, illustrating how they can be used in conceiving a pedagogical strategy.

Chapter 5, at the end of the paper, contains the conclusions and offers a synthesis of the contributions brought and points out the future directions in research.

Appendix I ("*A Detailed Study of the Elements of Ontological Engineering*") contains an ample study into ontological engineering, applicable in modern based knowledge systems, in a wide diversity of domains; the most important aspects, which have an impact on the proper objectives of the paper, are the basis of the contributions brought by us to in chapters 2 and 3.

Appendix II ("*Ontl_CG Application*")- presents the application proposed in chapter 3, section 3.6.

Appendix III ("*Conceptual Graphs Formalism used in this thesis*") presents the formal model used to represent knowledge, based on Simple Conceptual Graphs extended with rules and constraints.

NOTAȚII ȘI ABREVIERI

Abrev.	Termen în limba română	Termen în limba engleză
AI	Inteligență artificială	Artificial Intelligence
DAI	Inteligență artificială distribuită	Distributed Artificial Intelligence
AIED	Inteligență artficială aplicată în educație	Artificial Intelligence in Education
CAI	Instruire asistată de calculator	Computer Assisted Instruction
CAL	Învățare asistată de calculator	Computer Assisted Learning
CBL	Învățare mediată de calculator	Computer Based Learning
CBT	Antrenare mediată de calculator	Computer Based Training
ICAI	Instruire inteligentă asistată de calculator	Intelligent Computer Assisted Instruction
ICAL	Învățare inteligentă asistată de calculator	Intelligent Computer Assisted Learning
ICBT	Antrenare inteligentă mediată de calculator	Intelligent Computer Based Training
IES	Sistem(e) educațional(e) inteligent(e)	Intelligent Educational System(s)
ILE	Mediu inteligent de învățare	Intelligent Learning Environment
IIS	Sistem inteligent de instruire	Intelligent Instructional System
ITS	Sistem(e) tutorial(e) inteligent(e)	Intelligent Tutoring System(s)
DITS	Sistem(e) tutorial(e) inteligent(e) dsitribuit(e)	Distributed Intelligent Tutoring System(s)
HMS	Sistem(e) hypermedia	Hypermedia System(s)
AHS	Sistem(e) hypermedia adaptiv(e)	Adaptive Hypermedia System(s)
OAHS	Sistem(e) hypermedia adaptiv(e) deschis(e)	Open Adaptive Hypermedia System(s)
ID	Proiectarea instruirii	Instructional Design
LD	Proiectarea învățării	Learning Design
CMS	Sistem(e) de management al cursului/conținutului	Courseware/Content management system(s)
LMS	Sistem(e) de management al învățării	Learning Management System(s)
LAMS	Sistem(e) de management al activităților de învățare	Learning Activity Management System(s)
SBC	Sistem(e) bazat pe cunoștințe	Knowledge Based System(s)
ACT-R	Arhitectura cognitivă numită ACT-R (Teoria controlului adaptiv al gândirii-raționalului)	Theory Adaptive Control of Thought—Rational
SOAR	Arhitectura cognitivă numită SOAR (Stare, Operator, Rezultat)	State, Operator And Result
CG/CGs	Graf conceptual/Grafuri conceptuale	Conceptual Graph(s)

LISTA FIGURILOR

Figura 1-1 Definirea educației din diferite puncte de vedere 2
Figura 1-2 Instruirea asistată de calculator – un domeniu pluridisciplinar 4
Figura 1-3 Evoluţia teoriilor învăţării şi influenţele lor în instruirea asistată 7
Figura 1-4 Sinteza teoriilor învăţării şi a strategiilor de instruire 11
Figura 1-5 Evoluția sistemelor de instruire asistată şi a tehnologiilor educaţionale 13
Figura 1-6 Arhitectura modulară a unui ITS 18
Figura 1-7 Abordări în construirea modelului elevului 19
Figura 1-8 Aspecte vizate de expertiza pedagogică 21
Figura 1-9 Sisteme educaţionale inteligente distribuite – Direcţii de cercetare 25
Figura 1-10 Sinteză - Principalele tipuri de sisteme autor şi caracteristicile acestora 29
Figura 1-11 Standarde pentru descrierea sistemelor educaţionale la diferite niveluri 31
Figura 1-12 Clasificarea ontologiilor în funcţie de diferite criterii 38
Figura 1-13 Diferite tipuri de ontologii 39
Figura 1-14 Legătura între principalele elemente utilizate în construirea unui model conceptual și operaționalizarea acestuia 42
Figura 1-15 Ontologiile de proiectare a instruirii - ontologii de bază pentru sistemele autor 46
Figura 2-1 Relaţiile dintre vocabular, conceptualizare, angajament ontologic şi ontologie; Ontologie şi semnificaţie (adaptat după [109]) 54
Figura 2-2 Ilustrarea relaţiilor dintre vocabularul, conceptualizarea şi ontologia (parţială) unui sistem de instruire 55
Figura 2-3 Sinteză: Metodologii, instrumente şi limbaje pentru ontologii 61
Figura 2-4 Ciclul de viaţă a unei ontologii 63
Figura 3-1 Procesul de reprezentare a cunoştinţelor 68
Figura 3-2 Arhitectura generală a unui SBC pentru rezolvarea de probleme 68
Figura 3-3 Arhitectura generală a unui SBC pentru gestiunea cunoştinţelor 68
Figura 3-4 Clasificarea ontologiilor formale după gradul de formalizare 70
Figura 3-5 Conceptele din ontologia de reprezentare OntL_CG-MetaOnto 76
Figura 3-6 Relațiile din ontologia de reprezentare OntL_CG-MetaOnto 77
Figura 3-7 Schema generală de operaționalizare a unei ontologii în vederea utilizării acesteia în cadrul unui SBC 78
Figura 3-8 Schema detaliată de operaționalizare a unei ontologii în vederea utilizării acesteia în cadrul unui SBC 80
Figura 3-9 Ciclul de inferență pentru ontologia operațională 81
Figura 3-10 Axioma-schemă bazată pe sintaxa CG pentru simetria unei relații binare (Reprezentarea în sintaxa CG a simetriei relației R) 83
Figura 3-11 Axioma-schemă pentru anti-simetria relației R 83
Figura 3-12 Reflexivitatea / anti-reflexivitatea relației R 83
Figura 3-13 Relația diff, pentru exprimarea diferenței a două concepte 84
Figura 3-14 Axioma schemă pentru moștenirea proprietăților algebrice 84
Figura 3-15 Axioma schemă pentru moștenirea disjunctă 84
Figura 3-16 Axioma schemă pentru moștenirea exclusivității 84
Figura 3-17 Axioma schemă pentru moștenirea incompatibilității 85
Figura 3-18 Axioma despre semnătura binară a unei relații 85
Figura 3-19 Axioma despre semnătura ternară a unei relații 85
Figura 3-20 Conformitatea semnăturii (în rol_i, i=1, 2 sau 3) 85
Figura 3-21 Cardinalitatea minimă 2 asupra primului element din semnătura unei relații binare prin cuplu de λ-expresii 85

Figura 3-22 Cardinalitatea maximă 2 asupra primului element din semnătura unei relații binare prin cuplu de λ-abstracțiuni 86
Figura 3-23 Incompatibilitatea între relațiile rel1(t_1, t_2) și rel2(t_1, t_2) 86
Figura 3-24 OntL_CG – Cazuri de utilizare 91
Figura 3-25 Pachetele din aplicație 92
Figura 3-26 Clasele din pachetul cgmodel 93
Figura 3-27 Clasa CogitantClient 94
Figura 3-28 Clasele Graph, Projection și idTable 95
Figura 3-29 Componentele ontologiei 97
Figura 3-30 Clasa Ontology 98
Figura 3-31 Clasa ConceptualPrimitive 99
Figura 3-32 Clasa Edge 100
Figura 3-33 Clase pentru raționament (axiome, constrângeri pozitive și negative, reguli) 102
Figura 3-34 Axiome-schemă și contextul de utilizare 103
Figura 3-35 Axiome-schemă cu o singură primitivă (AxiomSchemaOne) și clasele pentru proprietățile relațiilor 104
Figura 3-36 Axiome-schemă (AxiomSchemaTwo) cu două primitive 105
Figura 3-37 Pachetul matching și clasele din pachet 107
Figura 3-38 Pachetul graphdrawing și clasele din pachet 108
Figura 3-39 Pachetul iotools și clasele din pachet 109
Figura 3-40 Pachetul exceptions și clasele din pachet 110
Figura 3-41 Pachetul ui pentru interfața cu utilizatorul și clasele din pachet 111
Figura 3-42 Ferestre grafice pentru motorul de inferența, operaționalizarea axiomelor, contextul de utilizare 114
Figura 3-43 Documentarea aplicației OntL_CG 115
Figura 3-44 Ontologie pedagogică (Ulrich, Leidig) 116
Figura 3-45 Axioma corespunzătoare strategiei specificate 117
Figura 3-46 Deducerea unor noi relații pe baza tipului semantic specificat în ontologie și a unei strategii pedagogice de compunere a documentului virtual 117
Figura 4-1 Triunghiul pedagogic propus de Houssaye 121
Figura 4-2 Cunoaşterea pedagogică, la diferite niveluri de abstractizare 122
Figura 4-3 Cunoaştere *pedagogică generală* şi cunoaştere *pedagogică contextuală* 124
Figura 4-4 Cunoştinţe pedagogice de conţinut 126
Figura 4-5 Componente ale cunoașterii pedagogice de conținut 126
Figura 4-6 Exemplu de descompunere a unui task repetitiv 129
Figura 4-7 Arhitectuta multi-nivel a bazei de cunoștințe despre strategiile pedagogice 131
Figura 4-8 Graful task-urilor și metodelor 133
Figura 4-9 Raționamentul realizat de către motorul de inferență 134

Figura AI - 1 Exemplu de CG 4
Figura AI - 2 Grafuri conceptuale (neorientate, etichetate) echivalente, în notaţie grafică 4
Figura AI - 3 Grafuri conceptuale (digrafuri) echivalente, în notație grafică 4
Figura AI - 4 Grafuri conceptuale (neorientate, etichetate) echivalente, în notaţie lineară 4
Figura AI - 5 Grafuri conceptuale (digrafuri) echivalente, în notaţie lineară 4
Figura AI - 6 Etichetarea nodurilor concepte şi relaţii 5
Figura AI - 7 Grafurile conceptuale ca reprezentare grafică a logicii [199] 6
Figura AI - 8 Grafurile conceptuale - sistem formal cu semantică logică [200] 7
Figura AI - 9 Posibilităţi de utilizare a sistemului formal al grafurilor conceptuale 8
Figura AI - 10 Graful conceptual simplu: mulţime de pointeri semantici către ontologie 9
Figura AI - 11 Un exemplu de raţionament cu grafuri conceptuale 10

Figura AI - 12 Limbaje pentru web-ul semantic, adaptat după [190] (Berners-Lee, 2000) 11
Figura AI - 13 Limbajele de reprezentare a ontologiilor pentru web 14
Figura AI - 14 Activităţi în ingineria ontologică 15
Figura AI - 15 Principiile ingineriei ontologice 26

Figura AII - 1 Mesaje de conectare la serverul CoGITaNT 1
Figura AII - 2 Meniul principal și meniurile secundare 2
Figura AII - 3 OntL_CG – Interfața generală a aplicației 3
Figura AII - 4 Crearea/deschiderea/salvarea fișierului (.cgxml, .owl) cu ontologia 4
Figura AII - 5 Specificarea/salvarea contextului de operaționalizare a ontologiei 6
Figura AII - 6 Încărcarea unui scenariu de operaționalizare existent 6
Figura AII - 7 Afișarea componentelor ontologiei în panoul din stânga 7
Figura AII - 8 Vocabular multi-lingual 8
Figura AII - 9 Afișarea grafică a ierarhiei de concepte 8
Figura AII - 10 Afișarea grafică a ierarhiei de concepte și de relații 9
Figura AII - 11 Bare pentru instrumente de lucru 9
Figura AII - 12 Specificarea proprietăților unui concept din domeniul de instruire 10
Figura AII - 13 Proprietățile unui concept din ipoteză/concluzie 11
Figura AII - 14 Proprietățile unei relații ipoteză/concluzie 12
Figura AII - 15 Proprietățile algebrice ale unei relații 13
Figura AII - 16 Axiome pentru semnătura relației binare/ternare 14
Figura AII - 17 Axiome pentru conformitatea semnăturii, moștenirea proprietăților algebrice și incompatibilitatea moștenirii 15
Figura AII - 18 Moștenirea disjunctă/exclusivă 16
Figura AII - 19 Testarea ontologiei 17
Figura AII - 20 Alte opțiuni 18
Figura AII - 21 Pachetele din OntL_CG 19
Figura AII - 22 Ierarhia claselor din pachete 19
Figura AII - 23 Pachetul cgmodel (1) 20
Figura AII - 24 Pachetul cgmodel (2) 20
Figura AII - 25 Documentarea clasei CogitantClient din pachetul cgmodel 21
Figura AII - 26 Clasele din pachetul cgmodel și descrierea lor 21
Figura AII - 27 Documentarea clasei RelationAntisymetry din pachetul cgmodel 22
Figura AII - 28 Pachetul matching 22
Figura AII - 29 Clasa MatchingAlgorithm din pachetul matching 23
Figura AII - 30 Pachetul exceptions 23
Figura AII - 31 Pachetul iotools 24
Figura AII - 32 Pachetul ui (1) 24
Figura AII - 33 Pachetul ui(2) 25
Figura AII - 34 Pachetul graphdrawing 25

Figura AIII- 1 Bază canonică, grafuri canonice și grafuriconceptuale 9
Figura AIII- 2 Familia SGC și complexitatea ([175]) 18

LISTA TABELELOR

Tabelul 1-1 Proiecte (sisteme) şi modul de utilizare a ontologiilor 48
Tabelul 2-1 Clasificarea primitivelor folosite în reprezentarea cunoştinţelor 52
Tabelul 2-2 Studiu comparativ al metodologiilor și metodelor din ingineria ontologică 59
Tabelul 2-3 Studiu comparativ al principalelor instrumente existente pentru construirea ontologiilor 59
Tabelul 4-1 Tipul învăţării şi termenii folosiţi de cercetători 123

"Learning is founding out what we already know. Doing is demonstrating that you know it. Teaching is reminding others that they know just as well as you. You are all learners, doers and teachers."

"A învăța înseamnă a regăsi ceea ce deja cunoaștem. A aplica înseamnă a demonstra ceea ce cunoști. A preda înseamnă a reaminti celorlalți ceea ce ei știu chiar la fel de bine ca și tine. Voi toți sunteți elevi, executanți și profesori.."

(Richard Bach, n. 1936)

1. STADIUL ACTUAL AL CERCETĂRILOR ÎN DOMENIUL INSTRUIRII ASISTATE DE CALCULATOR

1.1 INSTRUIREA ASISTATĂ DE CALCULATOR

1.1.1 NOȚIUNI PRELIMINARE

Un aspect deosebit de important al activităţii umane, cu multiple implicaţii în domeniile cultural, economic şi social, îl constituie **educaţia** (termenul latinesc *"educatio"* - creştere, îndrumare, hrănire, formare). Deşi iniţial cuvântul era operant pentru întreaga lume vie, din secolul al XVI-lea el s-a cantonat exclusiv în sfera de influenţă a universului uman. Educaţia se ocupă de *formarea personalităţii umane* în vederea integrării ei active, creatoare în viaţa socială [1]. *Educabilitatea* – concept derivat din educaţie - este o particularitate proprie omului, de a fi modelat structural şi informaţional.

Educaţia poate fi privită şi definită din mai multe unghiuri de vedere: ca proces, ca acţiune de conducere, ca acţiune socială, ca interrelaţie umană şi ca ansamblu de influenţe (figura 1-1). Educaţia vizează postularea şi împlinirea unui proiect de devenire umană, deci nu poate fi realizată fără a avea în vedere finalitatea demersului. Finalitatea educaţiei - din punct de vedere teoretic şi practic - trebuie considerată atât din punct de vedere ***teleologic***, cât şi ***praxiologic***.

Sensul teleologic al educaţiei constă în faptul că educaţia - în fiecare secvenţă de manifestare - este *ghidată, orientată şi reglată de un sistem de valori acţionale* (comenzi, exigenţe, intenţii, dorinţe, etc.) conştientizate şi uneori exprimate de către factorii angajaţi în acţiunea instructiv-educativă. De aceea, *proiectarea acţiunii este intrinsecă actului educativ*.

Sensul praxiologic al educaţiei este dat de finalităţile sale, ca valori practice (praxiologice), deoarece sunt consecinţe ale intenţionalităţii umane şi vizează eficientizarea acţiunii. Un lucru bine făcut se înscrie în orizontul unui ţel la care se ajunge respectând anumite canoane de raţionalitate, convertite într-o "gramatică" a acţiunii. *A acţiona în plan educaţional* înseamnă a delibera asupra traiectului de parcurs, a tinde către un scop în condiţii date, a identifica cele mai

bune mijloace pentru atingerea scopului şi a introduce în activitatea educaţională variabile şi factori care (auto)reglează acţiunea spre finalităţile propuse.

Termenul de **pedagogie** provine din limba greacă ("*pais"*, *"paidos"* – copil; "*agoge"* - conducere, educaţie; "*paidea"* - învăţământ, educaţie; *"paidagogos"* - îndrumător de copii, pedagog). *Pedagogia* este *ştiinţa* educaţiei care studiază esenţa şi trăsăturile fenomenului educaţional, scopul şi sarcinile educaţiei, valorile şi limitele ei, conţinutul, principiile, metodele şi formele de desfăşurare ale acesteia.

Spre deosebire de celelate ştiinţe (în a căror preocupare intră educaţia), pedagogia abordează educaţia prin prisma a *două coordonate fundamentale şi complementare*:

- a finalităţii acţiunii educative;
- a tehnologiei realizării sale.

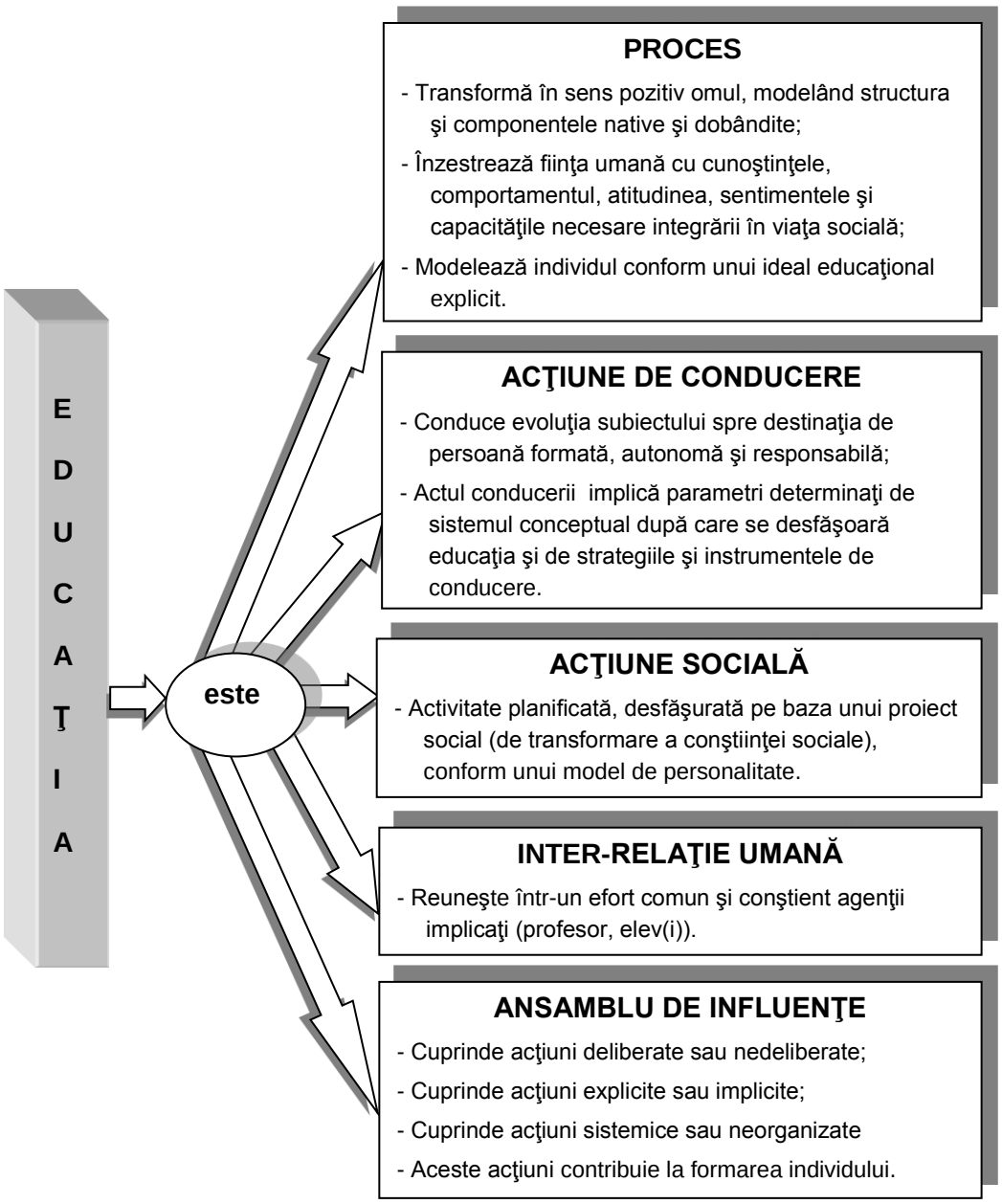


Figura 1-1 Definirea educației din diferite puncte de vedere

Cercetarea pedagogică trebuie privită ca o strategie desfăşurată cu scopul de a surprinde relaţii noi între componentele acţiunii educaţionale şi de a elabora - pe baza acestor relaţii -

soluţii optime ale problemelor ridicate de procesul educaţional. Consider că o bună strategie pedagogică trebuie să includă atât proiectarea prin conţinut, cât şi proiectarea prin obiective, privite într-o relaţie de complementaritate [2].

Ştiinţa instruirii a apărut încă din anul 1966, pornind cu teoria lui Bruner şi continuând cu cele ale lui Ausubel, Glaser, Gagné, Merril şi cu rezultatele cercetărilor psihologiei cognitive. Ştiinţa instrurii (IS) constă în teoriile, modelele şi metodologiile pentru instruire şi pentru cercetarea instruirii; ea are la bază *ştiinţa învăţării*, *ştiinţa cognitivă* şi *ştiinţa sistemelor*. Ştiinţa instrurii este o *ştiinţă a proiectării,* având atât *componente descriptive*, cât şi *prescriptive* [3].

Proiectarea instruirii (***I****nstructional* ***D****esign*, **ID**) – *parte a ştiinţei instruirii* formată din componentele prescriptive - reprezintă procesul sistematic de transpunere a principiilor generale ale ***învăţării*** şi ***instruirii*** în planuri pentru materialele de instruire şi învăţare. Este un proces sistemic şi sistematic de aplicare a strategiilor şi tehnicilor derivate din teoriile comportamentale (behavioriste), cognitive şi constructiviste, ca soluţie a problemelor instruirii. ID este independentă de domeniu, generică, bazată pe teorie; conţine concepte, reguli şi principii. Pentru ID, apare sinonimul “ingineria instruirii” [4].

Caracterul deosebit de complex al educaţiei şi necesitatea sporirii eficienţei şi calităţii sale - pe de o parte - şi progresele tehnologice şi din domeniul informaticii - de cealaltă parte - au condus la preocupări intense de a dezvolta şi de a integra în procesele de învăţare (instruire) *sisteme de instruire care folosesc calculatorul*.

Astfel, apare ***domeniul instruirii asistate de calculator***, ca punte de legătură între ştiinţele educaţiei şi ştiinţele inginereşti (informatică). Complexitatea problemelor pe care le ridică realizarea unor sisteme software suficient de flexibile pentru instruirea asistată de calculator a condus la ideea integrării în procesele de educaţie asistată a unor *tehnici specifice inteligenţei artificiale* (*“**A**rtificial **I**ntelligence”* - **AI**). Voi evidenţia în continuare etapele principale ale evoluţiei sistemelor de instruire asistată şi abordările care stau la baza acestora.

1.1.2 ABORDĂRI ȘI TEORII CU INFLUENȚĂ ASUPRA PROIECTĂRII PEDAGOGICE

Domeniul instruirii (inteligente) asistate de calculator este un domeniu pluridisciplinar, extrem de interesant, cu o problematică diversă, care încearcă să integreze tehnici diferite ale inteligenţei artificiale, modalităţi de reprezentare a cunoştinţelor, posibilităţi de modelare a raţionamentului, a agenţilor, a arhitecturilor multi-agent, a învăţării automate (engl. ”machine learning”) şi a comunicaţiei om-maşină. Disciplinele care au marcat importante schimbări în abordările din domeniul instruirii asistate de calculator sunt filozofia, psihologia, ştiinţele educaţiei şi inteligenţa artificială (figura 1-2). Punctele comune de studiu ale acestor discipline vizează *problematica mecanismelor de învăţare*, considerate ca *mijloace de inducere a instruirii asupra subiectului acestui proces* [5].

Din perspectiva *filozofică* (unde învăţarea este strâns legată de problema mai generală a cunoaşterii) se încearcă definirea cunoaşterii şi stabilirea surselor acesteia. Studiul mecanismulor de învăţare este un subiect central al cercetătorilor *psihologi*. *Pedagogia*, pe de altă parte, este ştiinţa care îşi propune să găsească cele mai adecvate şi eficiente căi de desfăşurare a proceselor de instruire. În sfârşit, dar nu în cele din urmă, *învăţarea automată,*

studiul strategiilor de instruire cooperative [6] *şi colaborative din inteligenţa artificială* au drept scop realizarea de sisteme de instruire adaptive, cu un comportament inteligent [7].

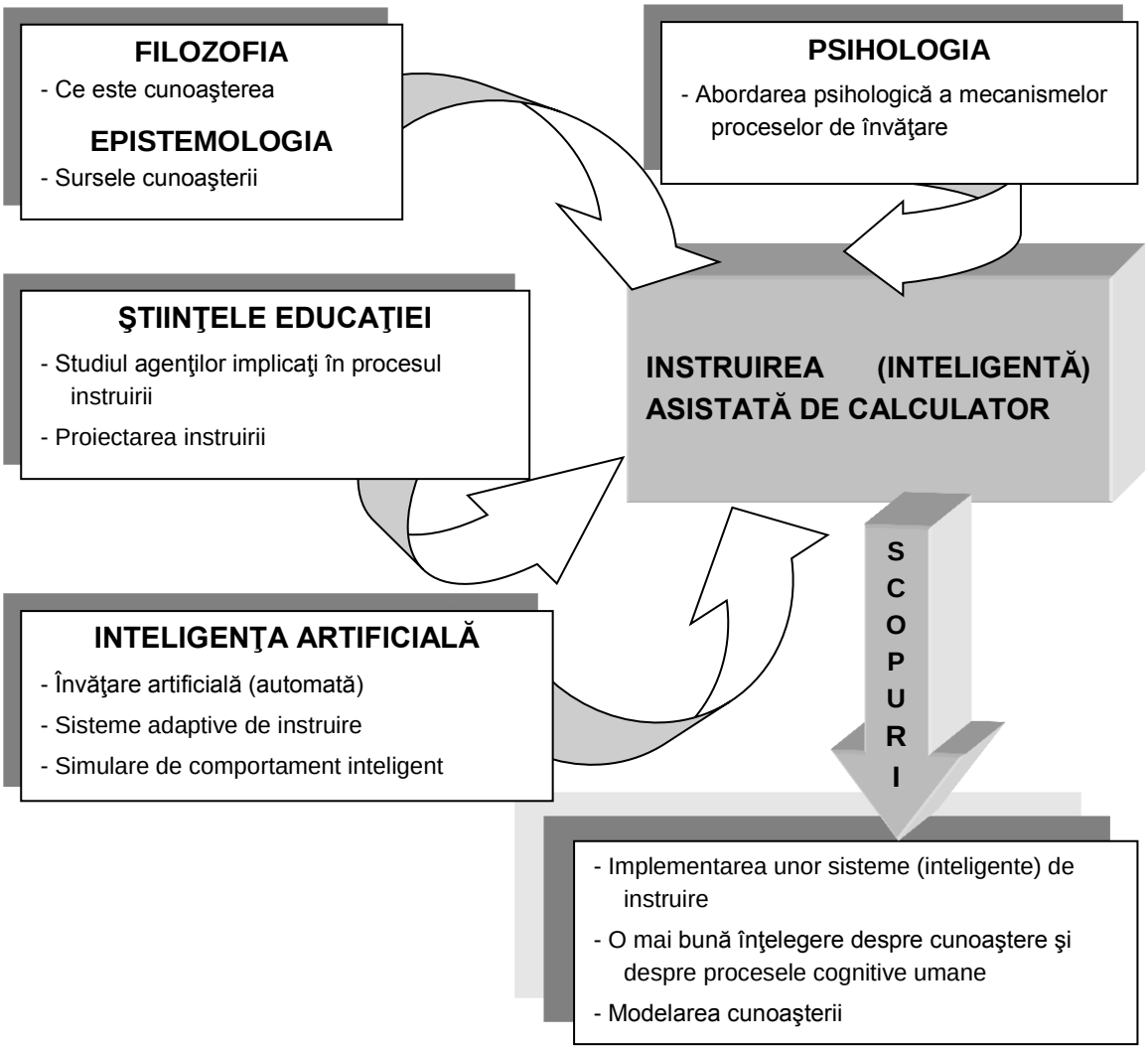


Figura 1-2 Instruirea asistată de calculator – un domeniu pluridisciplinar

PERSPECTIVA FILOZOFICĂ

În plan filozofic, problematica proceselor de învăţare – strâns legată de *problema definirii cunoaşterii şi a surselor acesteia* - a constituit un subiect de analiză şi dispută timp de milenii. Interesul studiului fenomenelor mentale apare în ***filozofia clasică*** în căutările lui *Socrate* despre natura cunoaşterii, în teoria învăţării lui *Platon* (conform teoriei expuse în dialogul "Menon", cunoştinţele noastre sunt înnăscute, iar procesul învăţării constituie, de fapt, un proces de reamintire) sau *Aristotel* (cunoaşterea abstractă este superioară oricărei alte forme de cunoaştere şi poate fi dobândită prin aplicarea logicii), sau în ***gândirea modernă***, în argumentarea lui Descartes (principiile evidente ale gândirii şi cunoaşterii sunt obţinute prin intuiţie şi îndoială, iar adevărurile certe – prin gândire, pe cale deductivă).

Practic, abordările diferitelor sisteme filozofice care încearcă să releveze sursele cunoaşterii, pot fi încadrate în:

- **Empiric vs raţional**

- *Cunoaşterea empirică* ia naştere cu ajutorul percepţiilor senzoriale, prin învăţarea din experienţă (observaţie, descriere sau experimente).
- *Cunoaşterea raţională* adânceşte cunoaşterea empirică printr-un proces de gândire abstractă (analiză, sinteză, inducţie, deducţie).

- **Înnăscut vs dobândit**
  - Sistemele filozofice care susţin *cunoaşterea înnăscută* consideră că mintea umană este înzestrată – încă de la naştere – cu un bagaj de cunoştinţe. Aceste sisteme filozofice consideră că realitatea din lumea înconjurătoare poate fi complet şi corect structurată în termenii entităţilor, proprietăţilor şi ai relaţiilor, iar raţionalitatea constă în manipularea simbolurilor abstracte.
  - Sistemele filozofice care susţin *cunoaşterea dobândită* consideră că nu există structuri cognitive înnăscute. Fiecare individ dobândeşte cunoştinţe (învaţă), îşi construieşte pas cu pas structuri cognitive complexe pe baza experienţei anterioare şi a prelucrării meta-cognitive, sau a reflecţiei (*vederi multiple* asupra aceluiaşi domeniu, semnificaţii şi perspective diferite ale aceluiaşi proces, concept, etc.).

Controversa dintre cunoaşterea înnăscută şi cea dobândită este susţinută de teoriile lui Chomsky şi Piaget. **Chomsky** consideră creierul uman înzestrat de la naştere cu o structură nervoasă înalt specificată, având capacitatea realizării unor structuri lingvistice complexe. Maturizarea creierului se realizează prin "developarea" (punerea în evidenţă) structurilor deja existente, nu prin dezvoltare sau învăţare. **Piaget** – deşi nu crede în existenţa structurilor cognitive înnăscute ale inteligenţei - acceptă că inteligenţa implică mecanisme nervoase de tipul reţelei booleene; pornind de la ele, subiectul uman construieşte pas cu pas structuri cognitive cât mai complexe, iar evoluţia de la starea iniţială la cea finală este rezultatul învăţării, al construcţiei interioare. Piaget susţine "*teoria organizaţională"*, văzută ca o serie de modele statice, progresiv şi ierarhic înlănţuite, în care copilul învaţă (evoluează) prin construirea unor cunoştinţe din ce în ce mai complexe [8].

Cele două teorii – teoria *structurilor lingvistice înnăscute a lui Chomsky* şi *teoria inteligenţei generale a lui Piaget* – au ca punct comun antiempirismul, susţinând că dobândirea cunoaşterii este un proces raţional, care îmbină atât aspecte înnăscute, cât şi construite (dobândite).

Abordările asupra surselor cunoaşterii din plan filozofic constituie *puncte de plecare ale teoriilor psihologice* privind procesele de învăţare (Figura 1-3).

PERSPECTIVA PSIHOLOGICĂ

Principalele teorii psihologice [9] ale învăţării sunt:

- **Psihologia behavioristă (comportamentală);**
- **Psihologia cognitivă;**
- **Psihologia constructivistă.**

Psihologia comportamentală priveşte *învăţarea ca fiind o modificare observabilă şi măsurabilă a comportamentului subiectului*. Numită şi *învăţare asociativă sau învăţare prin condiţionare*, are la bază învăţarea automată a unor răspunsuri pe baza unor stimuli [10]. În această abordare se încadrează:

- Teoria *condiţionării clasice a lui Pavlov* (1849-1936), bazată pe asocierea stimulilor şi repetarea acestor asocieri

- Teoria *conexionismului* propusă de Thorndike (1847-1949) care priveşte învăţarea ca pe un mecanism de formare a unei conexiuni (legătură de asociere) între stimuli şi răspuns
- Teoria lui *Watson* (1878-1958) care demonstrează rolul condiţionării în dezvoltarea răspunsurilor emoţionale la anumiţi stimuli şi explică apariţia sentimentelor de teamă, frică sau prejudecăţi la subiecţii umani
- Teoria lui *Skinner* (1904-1990) care tratează aspectele comportamentului operant (instrumental) prin mecanisme de întărire sau pedeapsă, cu aplicaţii în domeniul legislativ, religios, economic sau educativ.

Învăţarea comportamentală permite explicarea generalizărilor (privite ca reacţii similare la stimuli similari) sau a diferenţierii răspunsurilor la stimuli aparent similari (dar deosebiţi esenţial). Abordarea surprinde însă doar o mică parte din aspectele învăţării, deoarece simplifică psihicul uman, neglijează procesele cognitive şi reduce gândirea umană la reflexe condiţionate. Abordarea empirică a genezei cunoaşterii este insuficientă: nicio cunoaştere nu se datorează numai percepţiilor, acestea fiind dirijate şi încadrate în scheme de acţiune.

Psihologia cognitivă

Deşi susţine o serie de ipoteze ale psihologiei comportamentale - învăţarea prin stabilirea unor conexiuni stimuli-răspuns, importanţa întăririi (engl. “reinforcement”) şi rolul reacţiei inverse (engl. “feedback”) în procesele de învăţare - psihologia cognitivă *priveşte învăţarea ca pe un proces de achiziţie şi reorganizare a structurilor cognitive umane prin care omul dobândeşte, memorează şi prelucrează cunoaşterea*.

Conceptele principale ale teoriei cognitive aplicate în educație sunt:
- Schemele cognitive (scheme mentale) – structuri interne ale cunoaşterii. Prin combinarea, extinderea sau modificarea acestora, se pot obţine noi cunoştinţe. Achiziţia cunoaşterii este descrisă ca o activitate mentală.
- Modelul de prelucrare al memoriei – punct de plecare al *arhitecurilor cognitive* ACT[1] şi SOAR[2] – este format din *registrul senzorial de intrare*, *memoria de scurtă durată* şi *memoria de lungă durată*.
- Învăţarea este privită ca o schimbare (tranziţie) în spaţiul cunoaşterii.
- Elevul este un participant activ la procesul de instruire.

Aplicarea psihologiei cognitive în educație vizează descoperirea capacităţilor computaţionale şi de reprezentare a psihicului uman şi a proiecţiilor lor structurale şi funcţionale în creier [11], deci aspecte legate de meta-cunoaștere, cunoaștere declarativă/cunoaștere procedurală sau modul de organizare a cunoștințelor. Un **sistem cognitiv** este un sistem fizic care posedă două *proprietăţi*: *de reprezentare* şi *de calcul* [12].

Psihologia constructivistă

Psihologia constructivistă, iniţiată de Barlett (în anul 1932), consideră că învăţarea este procesul prin care subiectul îşi construieşte structurile de cunoaştere pe baza percepţiilor (concept din psihologia comportamentală) şi a experienţei sale, a cunoştinţelor anterioare, a

[1] ACT/ACT-R – Arhitectura cognitivă Controlul adaptiv al gândirii (engl., “Theory Adaptive Control of Thought—Rational”)

[2] SOAR – Arhitectura cognitivă Stare, operator, rezultat (engl., “State, Operator And Result”)

schemelor mentale (concepte din psihologia cognitivă) şi a credinţelor despre interpretarea obiectelor şi a evenimentelor [13].

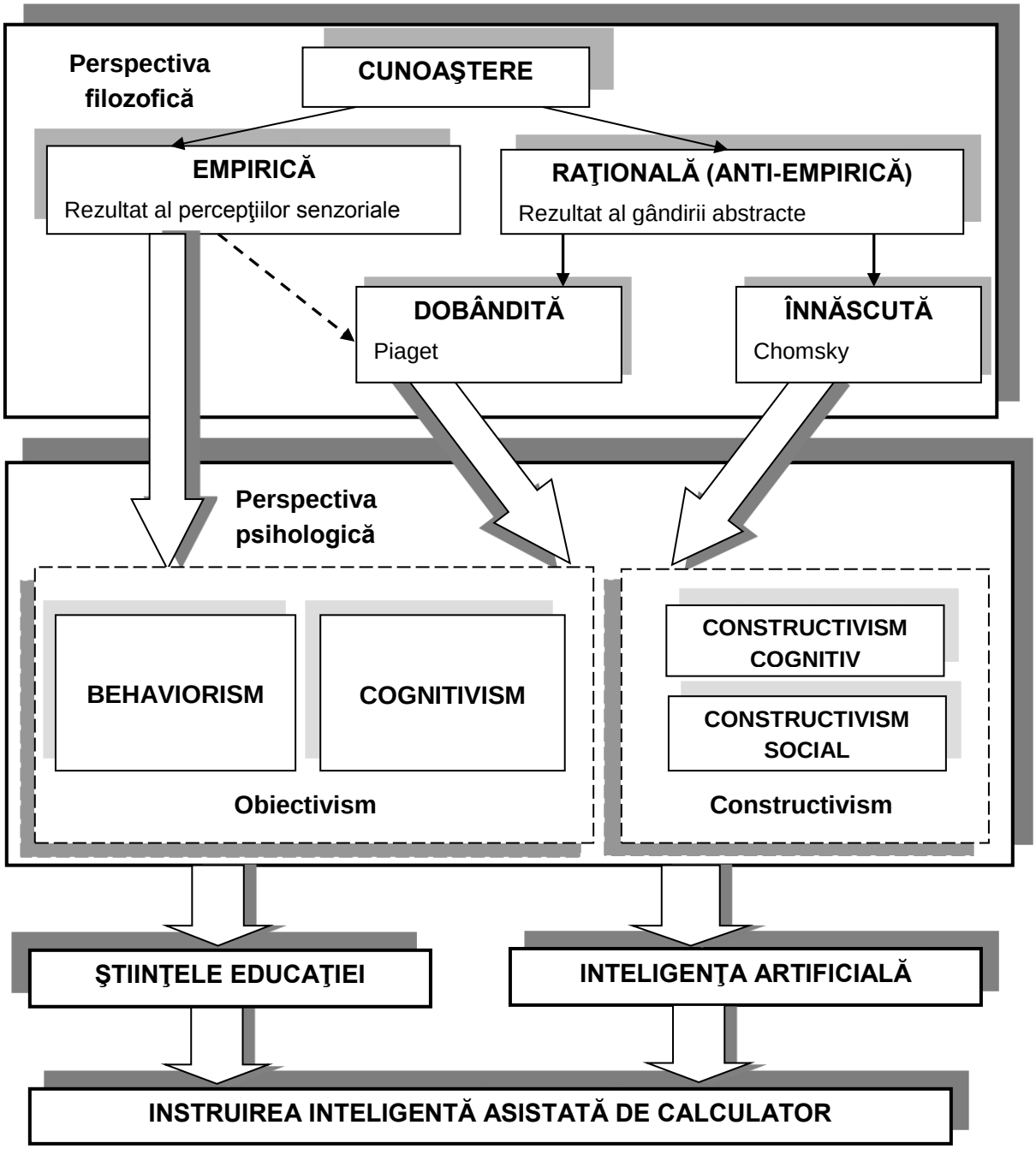


Figura 1-3 Evoluţia teoriilor învăţării şi influenţele lor în instruirea asistată

Conceptele de bază ale constructivismului evidenţiate de Merrill [14] sunt:

- Cunoaşterea este construită pe baza experienţei şi a schemelor mentale anterioare;
- Învăţarea constituie un proces activ de interpretare personală a lumii;
- Negocierea semnificaţiei, partajarea perspectivelor multiple asupra universului de discurs şi schimbările reprezentării interne a cunoaşterii prin colaborare favorizează învăţarea;
- Procesul de învăţare trebuie situat într-un context real, iar testarea trebuie să fie o activitate integrată în acest proces (nu o activitate separată).

Primele două principii conduc la abordarea **constructivistă cognitivă**, iar ultimele două – la **abordarea constructivistă socială** [15].

- *Cognitiviştii* încearcă să construiască modele formale ale structurilor psihice şi ale proceselor din memorie;
- *Constructiviştii* susţin în cea mai mare măsură conceptele cognitivismului, însă privesc învăţarea ca pe o construcţie personală a elevului, rezultată în urma interacţiunii cu mediul sau alţi participanţi.

Datorită multitudinii de concepte comune dintre psihologia cognitivă şi cea constructivistă, unii cercetători consideră că abordările principale în psihologie se reduc la *psihologia comportamentală* şi la cea *cognitivă*.

Din punct de vedere epistemologic, abordările psihologilor se împart între *obiectivism* (structurarea realităţii în entităţi, proprietăţi şi relaţii şi manipularea acestora) şi *constructivism* (construirea individuală a realităţii), cu sub-abordarea *situaţionismului* (în care învăţarea este un proces de construire individuală a cunoştinţelor prin interacţiunea cu mediul) [16].

Aşa cum se evidenţiază în această secțiune, teoriile învăţării au cunoscut o multitudine de abordări. Cu toate acestea, o teorie a învăţării nu le exclude automat pe celelalte, fiecare reprezentând un model explicativ al unei anumite laturi a procesului de învăţare şi nu al învăţării în general. Disputele dintre aceste orientări sunt de domeniul trecutului; în prezent critica lasă locul sintezei, complementarităţii şi asimilărilor reciproce. Nu putem privi o teorie ca fiind "mai bună" decât alta, ci doar ca fiind mai *adecvată într-o anumită situaţie de învăţare*.

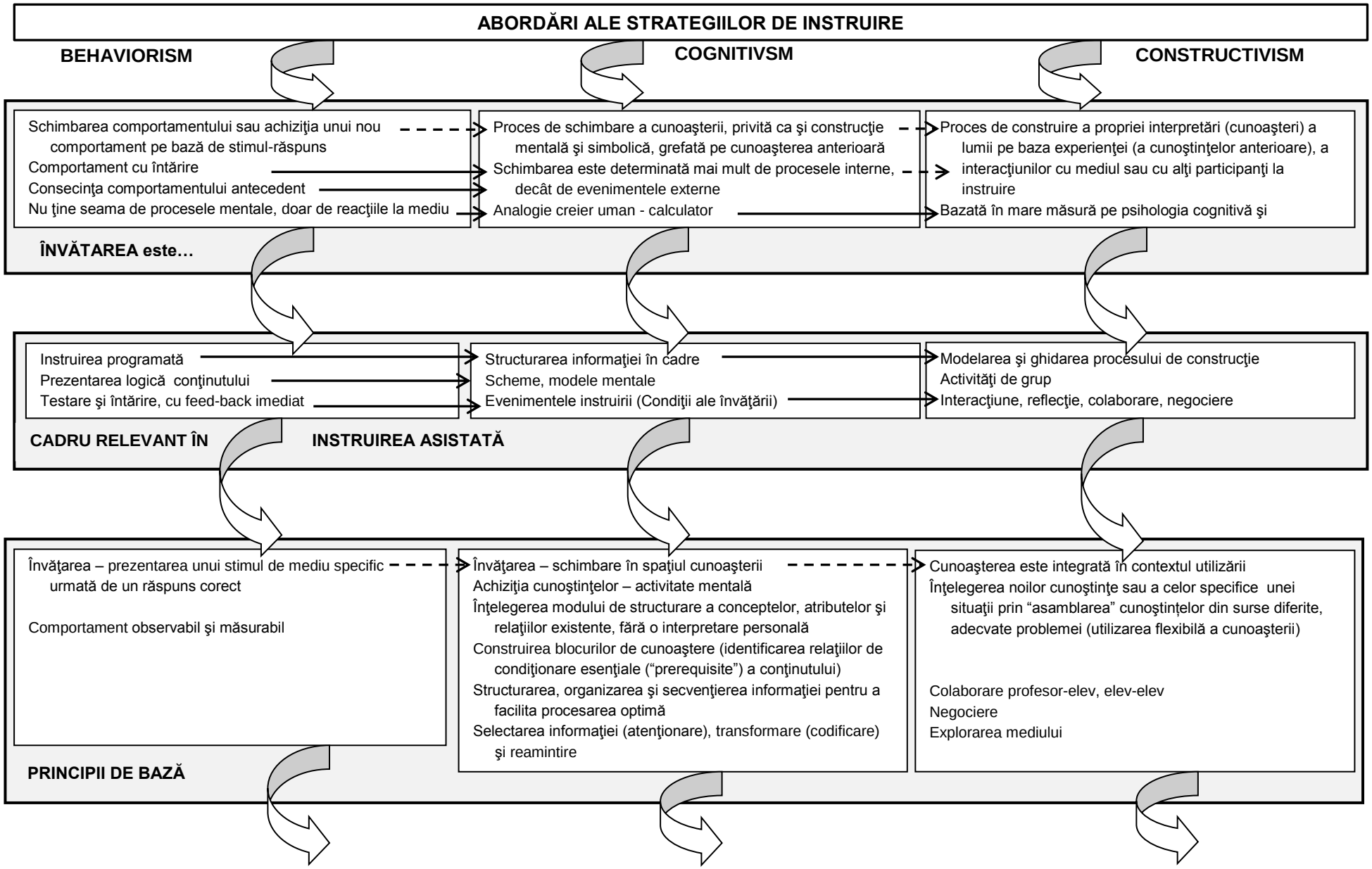
Aspectele care diferenţiază o teorie a învăţării de alta, astfel ca ea să fie utilizată în proiectarea instruirii, sunt [17]:

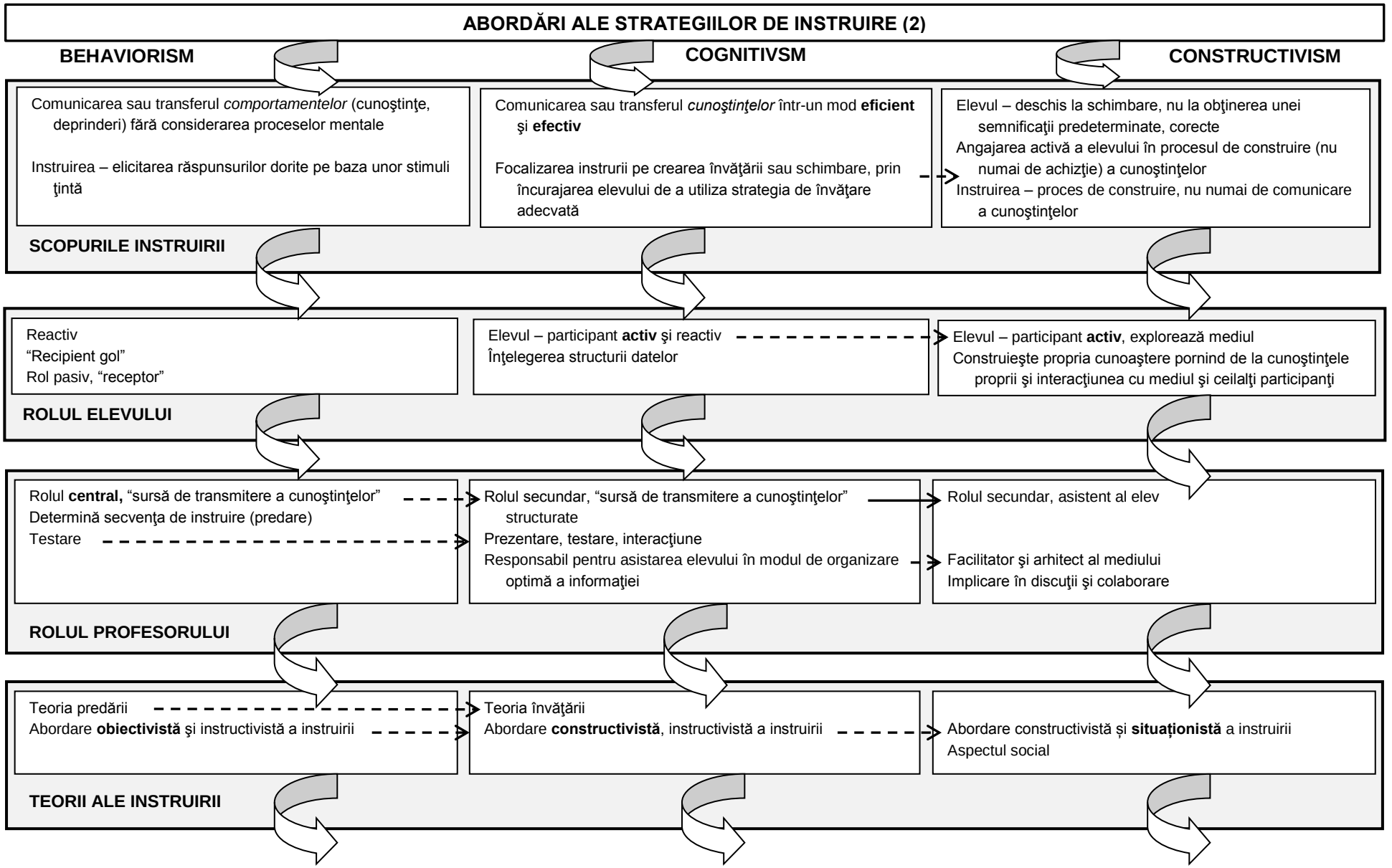
- Definirea procesului de învăţare și stabilirea factorilor care influenţează învăţarea;
- Rolul memoriei în învăţare și modalităţile de achiziţie şi transfer ale cunoaşterii;
- Tipurile de învăţare explicate printr-o anumită teorie și principiile relevante ale teoriei în proiectarea instruirii;
- Modul de structurare a instruirii în vederea eficientizării acesteia.

Deoarece problemele privind învăţarea sunt distribuite între filozofie, psihologie, educaţie, sociologie şi inteligenţa artificială, John Self [18] propune crearea unei noi discipline, numite ***Computational Mathetics*** (matetică de calcul). Adjectivul ***mathetic*** corespunde învățării (*legat de învăţare* - "*pertaining to leaning"*, de la grecescul "*manthanein"*, care înseamnă "*de învăţat"*), iar "*computational mathetics would be the study of matters pertaining to learning, and how it may be promoted, using the techniques, concepts and methodologies of computer science and artificial intelligence" (studiul disciplinelor care favorizează învăţarea, utilizând tehnici, concepte şi metodologii din ştiinţa calculatoarelor şi inteligenţa artificială)*.

O sinteză a teoriilor care apropie aspectele fundamentale *ale inteligenţei, respectiv ale procesului de învăţare* de *ştiinţa calculatoarelor,* influențând proiectarea pedagogică a unui sistem de instruire asistată, este prezentată în figura 1-4.

Toate aceste paradigme și teorii constituie surse ale cunoașterii pedagogice, așa cum se va ilustra și în capitolul 4.


ABORDĂRI ALE STRATEGIILOR DE INSTRUIRE
BEHAVIORISM
COGNITIVSM
CONSTRUCTIVISM
Schimbarea comportamentului sau achiziţia unui nou comportament pe bază de stimul-răspuns
Comportament cu întărire
Consecinţa comportamentului antecedent
Nu ţine seama de procesele mentale, doar de reacţiile la mediu
Proces de schimbare a cunoaşterii, privită ca şi construcţie mentală şi simbolică, grefată pe cunoaşterea anterioară
Schimbarea este determinată mai mult de procesele interne, decât de evenimentele externe
Analogie creier uman - calculator
Proces de construire a propriei interpretări (cunoaşteri) a lumii pe baza experienţei (a cunoştinţelor anterioare), a interacţiunilor cu mediul sau cu alţi participanţi la instruire
Bazată în mare măsură pe psihologia cognitivă şi
ÎNVĂTAREA este…
Instruirea programată
Prezentarea logică conţinutului
Testare şi întărire, cu feed-back imediat
Structurarea informaţiei în cadre
Scheme, modele mentale
Evenimentele instruirii (Condiţii ale învăţării)
Modelarea şi ghidarea procesului de construcţie
Activităţi de grup
Interacţiune, reflecţie, colaborare, negociere
CADRU RELEVANT ÎN INSTRUIREA ASISTATĂ
Învăţarea – prezentarea unui stimul de mediu specific urmată de un răspuns corect
Comportament observabil şi măsurabil
Învăţarea – schimbare în spaţiul cunoaşterii
Achiziţia cunoştinţelor – activitate mentală
Înţelegerea modului de structurare a conceptelor, atributelor şi relaţiilor existente, fără o interpretare personală
Construirea blocurilor de cunoaştere (identificarea relaţiilor de condiţionare esenţiale (“prerequisite”) a conţinutului)
Structurarea, organizarea şi secvenţierea informaţiei pentru a facilita procesarea optimă
Selectarea informaţiei (atenţionare), transformare (codificare) şi reamintire
Cunoaşterea este integrată în contextul utilizării
Înţelegerea noilor cunoştinţe sau a celor specifice unei situaţii prin “asamblarea” cunoştinţelor din surse diferite, adecvate problemei (utilizarea flexibilă a cunoaşterii)
Colaborare profesor-elev, elev-elev
Negociere
Explorarea mediului
PRINCIPII DE BAZĂ


ABORDĂRI ALE STRATEGIILOR DE INSTRUIRE (2)
BEHAVIORISM
COGNITIVSM
CONSTRUCTIVISM
Comunicarea sau transferul *comportamentelor* (cunoştinţe, deprinderi) fără considerarea proceselor mentale
Instruirea – elicitarea răspunsurilor dorite pe baza unor stimuli ţintă
Comunicarea sau transferul *cunoştinţelor* într-un mod **eficient** şi **efectiv**
Focalizarea instrurii pe crearea învăţării sau schimbare, prin încurajarea elevului de a utiliza strategia de învăţare adecvată
Elevul – deschis la schimbare, nu la obţinerea unei semnificaţii predeterminate, corecte
Angajarea activă a elevului în procesul de construire (nu numai de achizţie) a cunoştinţelor
Instruirea – proces de construire, nu numai de comunicare a cunoştinţelor
SCOPURILE INSTRUIRII
Reactiv
"Recipient gol"
Rol pasiv, "receptor"
Elevul – participant **activ** şi reactiv
Înţelegerea structurii datelor
Elevul – participant **activ**, explorează mediul
Construieşte propria cunoaştere pornind de la cunoştinţele proprii şi interacţiunea cu mediul şi ceilalţi participanţi
ROLUL ELEVULUI
Rolul **central,** "sursă de transmitere a cunoştinţelor"
Determină secvenţa de instruire (predare)
Testare
Rolul secundar, "sursă de transmitere a cunoştinţelor" structurate
Prezentare, testare, interacţiune
Responsabil pentru asistarea elevului în modul de organizare optimă a informaţiei
Rolul secundar, asistent al elev
Facilitator şi arhitect al mediului
Implicare în discuţii şi colaborare
ROLUL PROFESORULUI
Teoria predării
Abordare **obiectivistă** şi instructivistă a instruirii
Teoria învăţării
Abordare **constructivistă**, instructivistă a instruirii
Abordare constructivistă și **situaționistă** a instruirii
Aspectul social
TEORII ALE INSTRUIRII

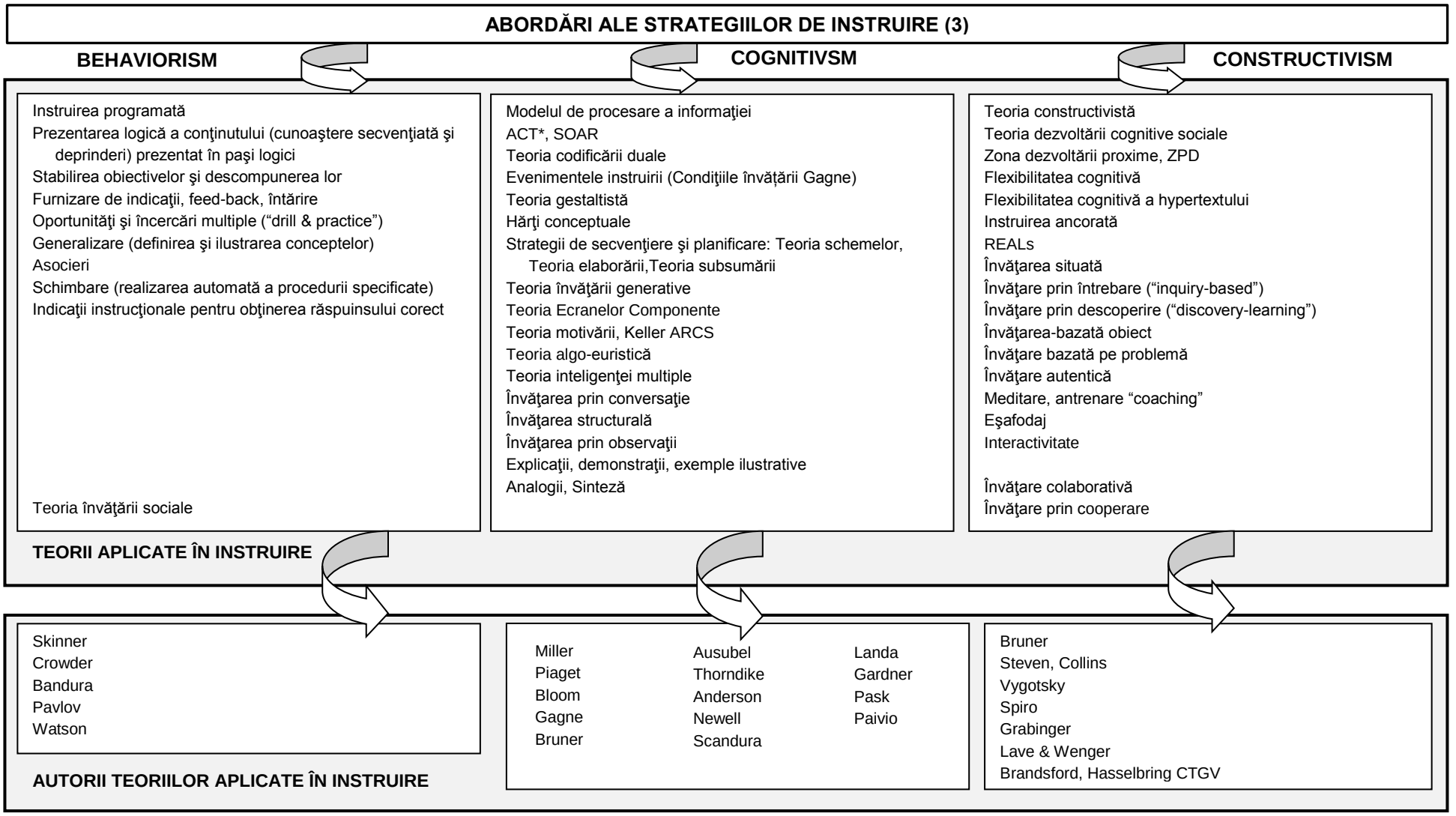


Figura 1-4 Sinteza teoriilor învăţării şi a strategiilor de instruire

1.2 SISTEME DE INSTRUIRE ASISTATĂ DE CALCULATOR

1.2.1 EVOLUȚIE ȘI CARACTERISTICI

Evoluția sistemelor de instruire asistată (figura 1-5) a fost determinată pe de o parte de *tehnologie* (hypertextul, multimedia și interneteul; e-learning-ul; web-ul semantic), iar de cealaltă parte de *schimbările din celelalte domenii (psihologie, filozofie, pedagogie, educație și inteligența artificială).* Tehnologiile hypermediei şi Internetului sunt prin ele însele mijloace utilizate pe scară largă în educaţie, fiind incorporate în prezent în majoritatea sistemelor de instruire (clasice sau inteligente [19]).

Primele maşini (**1920-1960**) folosite în procesul instruirii au fost maşinile de predare (se concentrau mai mult asupra hardware-lui), bazate pe psihologia comportamentală[3] şi teoria predării ("teaching theory"). Foloseau **metoda instruirii programate**, fiind promovate de **Skinner** ("părintele instruirii programate"). Câteva exemple de astfel de sisteme sunt:

- *Drum Tutor* (1925) – o maşină de testare-evaluare, cu teste de tip alegere multiplă (engl. "multiple-choice");
- *Maşina lui Skinner, "Foringer"* (1954) – bazată pe teoria operării condiţionale – care testează cunoştinţele elevului şi îi furnizează un feedback de întărire în cazul unui răspuns corect;
- *Maşina lui Crowder*[4] care antrena soldaţii americani în localizarea defectelor echipamentelor de zbor și care încearcă să adapteze instruirea - pentru prima dată - la *caracteristicile individuale ale elevului, pe baza unor* **ramificări**;
- *Maşina lui Pask* (1957) – care introduce o formă de interacţiune (cu elevul) rudimentară (susţinând *teoria învăţării prin conversaţie*) şi un *precursor al modelului student din sistemele inteligente;*
- *PLATO* (Programmed Logic for Automated Teaching Operations) (1960), sistem de instruire prin exerciţii şi practică (*"drill & practice tutoring"*), precursorul sistemelor de instruire la distanţă, putând conecta 1000 de terminale, aflate în locuri diferite în U.S.A.;
- *TICCIT* (Time-shared, Interactive, Computer Controlled Television) (1970) – sistem de instruire prin televiziune a adulţilor.

Deși aceste sisteme au fost utilizate în special în laboratoarele de cercetare, ele demonstrează eficacitatea sistemelor de instruire în educaţie: *"The PLATO and TICCIT systems opened the gateway in the potential of a technology market in education"* (Sistemele PLATO și TICCIT au deschis poarta potențialului tehnologiei în educație) [20].

Sistemele de instruire clasice (**1960-1980**), desemnate prin sintagmele CAL/CAI, CBT/CBL), construite pe principiile ***"programării lineare" sau "programării ramificate"***, sunt folosite cu succes (în variante îmbunătăţite şi din ce în ce mai performante) şi în perioada actuală [21], în

[3] Locul şi rolul teoriilor behavioriste în istoria sistemelor de instruire asistată pot fi comparate cu locul şi rolul sistemului de numeraţie binar în dezvoltarea calculatoarelor: indispensabile, dar insuficiente .

[4] Aceluiaşi cercetător i se datorează şi *apariţia strămoşului dispozitivelor multimedia*, Auto-Tutor II.

foarte multe domenii educaţionale. La baza dezvoltării lor s-a aflat obiectivul de **individualizare a instruirii**, realizat însă *doar prin formele de interacţiune cu elevul* și prin *generarea automată a materialului didactic*. Aceste sisteme prezintă o **slabă adaptivitate** la *nevoile cognitive individuale* ale elevului şi la *ritmul propriu de învăţare*.

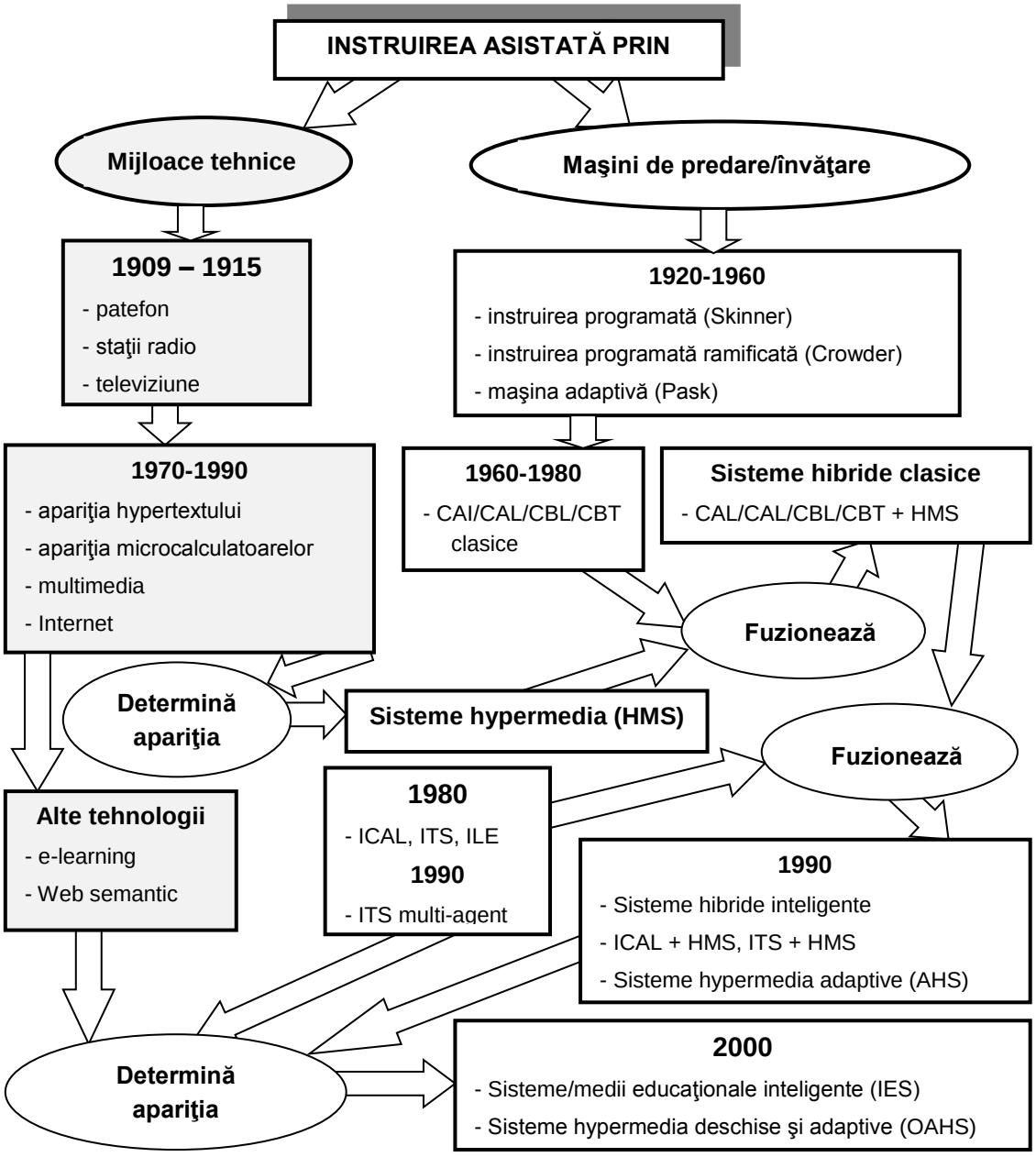


Figura 1-5 Evoluția sistemelor de instruire asistată şi a tehnologiilor educaţionale

Caracteristicile sistemelor CAI/CAL clasice sunt:

- *Cunoştinţele domeniului* instrucţional sunt *statice*, implementate în forma finală [22];
- *Modul de interacţiune* cu elevul este *ghidat* (predominant) – caz în care căile de interacţiune sunt şi ele predefinite – sau *liber*;
- *Obiectivele instruirii* se limitează la memorarea unor cunoştinţe declarative sau a unor secvenţe de proceduri;
- *Strategiile pedagogice* sunt *implicite* sau *absente* (în hypertextul nestructurat, empiric);

- *Individualizarea instruirii* este ineficientă, fiind dată doar de modul de secvenţiere a conţinutului educaţional [23];
- *Inabilitatea* de a adapta strategiile pedagogice pentru diferiţi elevi;
- *Inabilitatea de a accepta şi trata răspunsuri neaşteptate* ale elevilor, deoarece toate reacţiile elevului în interacţiunea cu sistemul sunt anticipate;
- *Inabilitatea* sistemului de a lua decizii autonome privind procesul de instruire;
- *Inabilitatea* sistemului software de a se *autoinstrui*.

În diversitatea lor, sistemele clasice de instruire – indiferent de domeniul în care sunt folosite, de categoriile utilizatorilor sau de tehnologia pe care se bazează - pot fi privite ca nişte "cărţi electronice sofisticate". Prezentarea conţinutului instrucţional se bazează pe **cadre** de cunoaştere, a căror structură şi secvenţiere este predefinită de autorul cursului (instruirea programată şi instruirea programată ramificată).

Apariţia microcalculatoarelor (1977), a hypertextului (numit *"Guttenberg 2"*) care prezintă avantajul delinearizării textului, a multimediei, hypermediei şi a Internetului – au condus la apariţia unor *noi modalităţi de comunicare şi prezentare a informaţiei*, fiind totodată *medii importante* de realizare a unor studii în *psihologia cognitivă* şi cea *constructivistă*.

Hypertextul a oferit şansa *"deliniarizării"* textului (deci a modului de comunicare a ideilor) şi reprezintă un mediu important pentru realizarea unor *studii în psihologie şi ştiinţele instruirii:*

- Dacă hypertextul este *nestructurat*, pe baza observaţiilor despre modul de navigare a utilizatorului şi de asimilare a informaţiei prezentatate, pot fi relevate diferenţele individuale dintre elevi (obţinerea unor modele ale utilizatorului); aceste diferenţe reprezintă repere importante pentru o bună proiectare sau secvenţiere a unui curs.
- Hypertextului *structurat* logic într-o reţea semantică, facilitează învăţarea (achiziţia cunoaşterii, integrarea şi sintetizarea cunoştinţelor, conform abordării cognitive [24].

Pot fi analizate, astfel, efectele unor structuri alternative ale aceluiaşi material asupra comportamentului elevului. Diferenţele dintre structurile cunoştinţelor dobândite de elev folosind hypertextul nestructurat şi cele dobândite folosind hypertextul structurat pot furniza o proiectare euristică completă.

Suporturile multimedia permit prezentarea şi comunicarea informaţiei (cunoştinţelor) [25] într-un mod cât mai eficient şi captivant, iar vizualizarea (un studiu comparativ al tehnicilor de vizualizare găsim în [26]) are roluri multiple şi variate în instruire: *captarea atenţiei elevului*, *motivarea*, *clarificarea unor concepte sau procese*, *modalitate alternativă de prezentare* [27].

Hypertextul şi sistemele multimedia fuzionează în **sisteme hypermedia**. Avantajele importante oferite de sistemele hypermedia [28]:

- Un grad înalt de interactivitate cu elevul, acces ușor și multi-utilizator la distanţă, la toate tipurile de informație;
- Mediu de realizare a studiilor în psihologia cognitivă. Procesele cognitive legate de căutarea şi prelucrarea informaţiei au rol fundamental în înţelegerea proceselor de învăţare umană. Bogatul conţinut informaţional – chiar nestructurat, lipsit de semnificaţia semantică - oferă posibilităţi de interpretare prin reguli foarte complexe, fără a fi necesare prea multe cunoştinţe adiţionale;
- Sunt considerate ***sisteme de învăţare***, care furnizează un control deplin al accesului elevului la cunoştinţe (fără expertiză pedagogică). Libertatea elevului de a naviga prin

multitudinea informaţiilor - navigare ghidată de curiozitate, inteligenţă şi interes pentru anumite informaţii – conduc la satisfacerea nevoilor afective ale elevului [29];
- Controlul şi iniţiativa oscilează între mediu şi deciziile luate mai târziu în funcţie de individualitatea elevului, fiind condiţionate, pe rând, de adaptivitatea şi flexibilitatea mediului în funcţie de diferenţele individuale [30];
- Elevul ”învaţă prin descoperire”.
- Suport mixt pentru exprimarea informaţiei într-un mod mult mai eficient, captivant şi expresiv.
- Fac posibilă *învăţarea la distanţă* şi *învăţarea prin claborare sau cooperare* (engl.“collaborative learning”, ”cooperative learning”).

Totuşi, bogăţia conţinutului informaţiei poate constitui şi un dezavantaj major: elevul se poate pierde în reţeaua de informaţii, poate deveni dezorientat, iar învăţarea nu va mai avea efectul scontat [31].

Toate aceste considerente conduc *fuzionarea* sistemelor de instruire (clasice sau inteligente) cu cele hypermedia, în ***sisteme hibride de instruire,*** **sisteme de e-learning*/sisteme de instruire bazate pe web*** [32] ***(eventual, web semantic)***, care să îmbine avantajele oferite de ambele tipuri de sisteme, stabilind o balanţă între nevoile cognitive şi cele afective ale elevului [33].

Sistemele clasice hibride sunt sisteme clasice combinate cu cele hypermedia, oferind avantajele accesului la distanţă, multi-utilizator şi a varietăţii metodelor de prezentare (desene, animaţie, sunet, etc.).

Soluţia de depăşire a limitărilor [34] sistemelor “clasice” s-a conturat în anii **1970-1980**, prin introducerea tehnicilor **inteligenţei artificiale**. Etapa **sistemelor de instruire inteligente** (desemnate inițial prin sintagmele ICAI, ICAL, ICBT, ILE sau ITS) – este deschisă prin cercetările lui Carbonell, care propune paradigma de *”**orientare a structurii informaţiei**”* şi utilizează reţelele semantice pentru un model explicit de reprezentare a cunoştinţelor domeniului în care se realizează instruirea. Sunt **sisteme bazate pe cunoştinţe**, cu capacitatea **adaptivă**, obţinută – în principal – prin *structurarea şi separarea cunoştinţelor domeniului instrucţional* de *cunoştinţele pedagogice* care conduc acest proces.

Primii paşi în obţinerea sistemelor inteligente au fost realizaţi prin *sistemele generative* (în care problemele propuse elevului erau generate dinamic cu ajutorul gramaticilor, a şabloanelor textuale sau aleator) și prin *sistemul SCHOLAR*, prin care Carbonell propune paradigma de ”*orientare a structurii informaţiei*”. Dintre *primele sisteme inteligente experimentale*, majoritatea fiind *sisteme expert*, putem aminti:

- *SCHOLAR* (1970, Carbonell&Collins) – instruire în Geografia Americii de Sud, dialog socratic (cu iniţiativă mixtă) și diagnoza greşelilor elevului, structurarea cunoştinţelor domeniului, explicitarea cunoaşterii expertului;
- *WHY* (1975 Collins&Stevens) – extensie a sistemului SCHOLAR, pentru studiul naturii dialogului socratic (în stailirea cauzelor precipitațiilor);
- *BUGGY* (1981, Suppes) pentru diagnoza greşelilor elevilor în matematică;
- *SOPHIE* (1982, Brown& Burton), mediu reactiv pentru simularea și diagnoza defectelor circuitelor electrice;
- *WEST* (1982, Brown) mediu reactiv cu meditare la cerere (engl. “coach”), critică deciziile elevului, îi furnizează sugestii şi comentarii;

- *GUIDON* (1983, Clancey), urmaş al MYCIN-ului, pentru diagnoza medicală;
- *LISP Tutor* (1984, Anderson&Brown) – instruire în programare, compară performanţele expertului cu cele ale elevului, aplică teoria ACT și “modelul păstrării urmei” (engl. ”model tracing”) pentru furnizarea de indicații;
- Geometry TUTOR (1987, Anderson&Burton) - sistem interactiv pentru introducere în geometria plană (calculul ariilor figurilor), aplică teoria ACT, realizează meditarea ghidată avansată (engl ”*Advanced Computer Tutoring*”).

Tehnicile de inteligenţă artificială încearcă să răspundă următoarelor întrebări:

- Care este natura cunoaşterii şi care sunt modelele care explică procesele de învăţare umană (înţelegerea proceselor)?
- Cum pot fi elaborate sisteme automate de învăţare, care să imite procesele cognitive umane (simularea pe calculator a proceselor)?
- Care sunt modalităţile de structurare şi modelare a cunoaşterii şi modalităţile de secvenţiere şi planificare a conţinutului instrucţional, pentru eficientizarea învăţării?
- Cum pot fi valorificate experienţele anterioare, prin partajarea sau reutilizarea cunoştinţelor, în vederea obţinerii unor sisteme inteligente de instruire, într-un timp mai scurt şi cu cost cât mai mic?

Ca urmare, din punct de vedere a aplicării tehnicilor de inteligență artficială în dezvoltarea sistemelor inteligente de instruire, perioada până în 1980 este considerată *perioada sistemelor expert*, perioada 1980-1990 este considerată *perioada sistemelor bazate pe cunoștințe* (reprezentarea declarativă a cunoștințelor, orientare către conținutul acestora), iar perioada de după 1990 este cea a *reprezentării partajate și explicite a cunoștințelor, deci a ontologiilor*.

Noile tehnologii, cercetările din psihologie şi cele din inteligenţa artificială au determinat concentrarea atenţiei ***ştiinţei proiectării instruirii*** în noi puncte de interes:

- De strategiile de predare la cele de învăţare, meditare şi ghidare;
- De la strategiile de predare expozitive la cele active, interactive;
- De la *predarea şi învăţarea ”**instructivistă**”* (elevul are un *rol pasiv*, fiind privit doar ca “receptor” al noilor cunoştinţe) la cea *”**constructivistă**”* (elevul are un *rol activ*, îşi construieşte noile cunoştinţe pe baza celor anterioare sau a interacţiunii cu mediul) [35].
- De la *cunoaşterea ”**obiectivistă**”* la cea *”**constructivistă**”*: din perspectiva obiectivismului scopul sistemelor inteligente de instruire a fost acela de a ajuta elevul să înţeleagă şi să ”înveţe” modul de structurare a conceptelor şi a relaţiilor din domeniul în care se instruieşte (*structură definitivă*, *unică*, dată de expertul în domeniu, cu o singură semnificaţie, independentă, obiectivă), să înveţe “adevărul” (adevăr care poate fi măsurat precis, cu ajutorul testelor); din perspectiva c*onstructivismului* (care presupune că realitatea este construită individual - pe baza experienţei anterioare, a prelucrării metacognitive, sau a reflecţiei (*vederi multiple* asupra aceluiaşi domeniu, semnificaţii şi perspective diferite ale aceluiaşi proces, concept, etc.) **sau** ca rezultat al interacţiunii cu mediul **sau** alţi participanţi, învăţarea constă în strategiile (de achiziţie a cunoştinţelor, de exemplu *dialogul socratic*) folosite pentru atingerea unui obiectiv şi poate fi estimată prin observaţii şi dialog (este mai mult o estimare subiectivă decât una obiectivă).

1.2.2 ARHITECTURA MODULARĂ A UNUI SISTEM INTELIGENT DE INSTRUIRE

Tutorialele inteligente sunt ***sisteme bazate pe cunoştinţe***. De aceea, una dintre cele mai importante probleme este *delimitarea categoriilor de cunoştinţe* care trebuie avute în vedere, precum şi *modalităţile cele mai adecvate de reprezentare şi prelucare a acestora*.

Tutorialele inteligente (ITS, engl. "Intelligent Tutoring System") pot fi descrise (cel puţin teoretic) prin patru modele/expertize:

- *Modelul domeniu* (numit şi "*subject matter*" (engl.), "***domeniul expert"*** sau *"modulul expert"* sau "*expert articulat"*) care conţine cunoştinţele domeniului de instruire, numite uneori şi "*cunoştinţe de context*" [36];
- *Modelul elev* care conţine cunoştinţele despre elev (ce cunoaşte şi ce nu cunoaşte, etapa curentă din procesul instruirii, dar şi etapele parcurse anterior, etc.);
- *Modelul pedagogic* sau *modelul didactic* sau *modulul tutorial* (numit şi *"instruction expert*" sau *"teaching expert" (engl.))* care conţine cunoştinţele referitoare la tacticile, strategiile şi euristicile de conducere a procesului de instruire;
- *Modelul de comunicaţie* sau *de interfaţă* (mai mult sau mai puţin distinct de cel tutorial) care conţine cunoştinţe despre modul de comunicare cu elevul şi prezentare a materialului instrucţional.

Nu toate aceste modele [37] se regăseasc în arhitectura unui ITS: *modulul tutorial şi cel de comunicare pot fi tratate împreună*, deoarece formele de comunicare propuse elevului sunt elemente determinante pentru învăţare. În mod ideal, un ITS trebuie să fie un ***triplu sistem expert***: în domeniul în care realizează instruirea (***"CE"?***), în analiza cunoştinţelor elevului (***"CUI"?***) şi în pedagogie (***"CUM"?***).
Bazele de cunoştinţe ale celor mai multe tutoriale inteligente înglobează primele trei baze de cunoştinţe (cunoştinţele domeniului, cunoştinţele despre elev şi cele referitoare la tacticile şi strategiile tutoriale). Toate aceste categorii de cunoştinţe, între care există *interconexiuni multiple*, sunt prelucrate în vederea obţinerii unor *regimuri flexibile, individualizate* pentru fiecare elev. Datorită *diversităţii cunoştinţelor* (*cunoştinţe declarative "CE"*, *cunoştinţe procedurale "CUM"*, *cunoştinţe cauzale "DE CE",* etc.) tehnicile de reprezentare sunt variate (orientate obiect, cadre, reţele semantice, grafuri conceptuale, logici de descriere, reguli de producţie, formule logice, etc.).

Diferenţele dintre tutoriale inteligente existente constau tocmai în această diversitate de reprezentare a modelelor expertizei. Interdependenţa bazelor de cunoştinţe ale unui ITS este relativă, deoarece acestea conţin cunoştinţe complexe, care necesită moduri de reprezentare diferite: structura modelului elev se găseşte în strânsă relaţie cu structura domeniului; forma strategiilor de instruire depinde de structura modelului domeniului și de modelul domeniului; acţiunile din strategiile de instruire depind de modelul interfeţei.
Concepţia fiecăruia dintre module ridică o serie de probleme în ceea ce priveşte *funcţionalitatea* acestora şi *modul lor de integrare într-o arhitectură informatică unitară*, abordată uneori în termenii de ***sistem multiagent***. Arhitectura prezentată (figura 1-6) o întâlnim în tutorialele inteligente care aplică ***strategia pedagogică directă*** (care implică agentul artificial – profesorul şi agentul uman – elevul); în ***sistemele cu strategii pedagogice de grup*** (în care intervin mai mulţi agenţi tutoriali) arhitectura va suferi unele modificări.

Modelul domeniului

Modelul domeniului conţine cunoştinţele domeniului de instruire, *"CE"* trebuie învăţat: conceptele, atributele şi relaţiile dintre concepte, dar şi informaţii despre procesele de

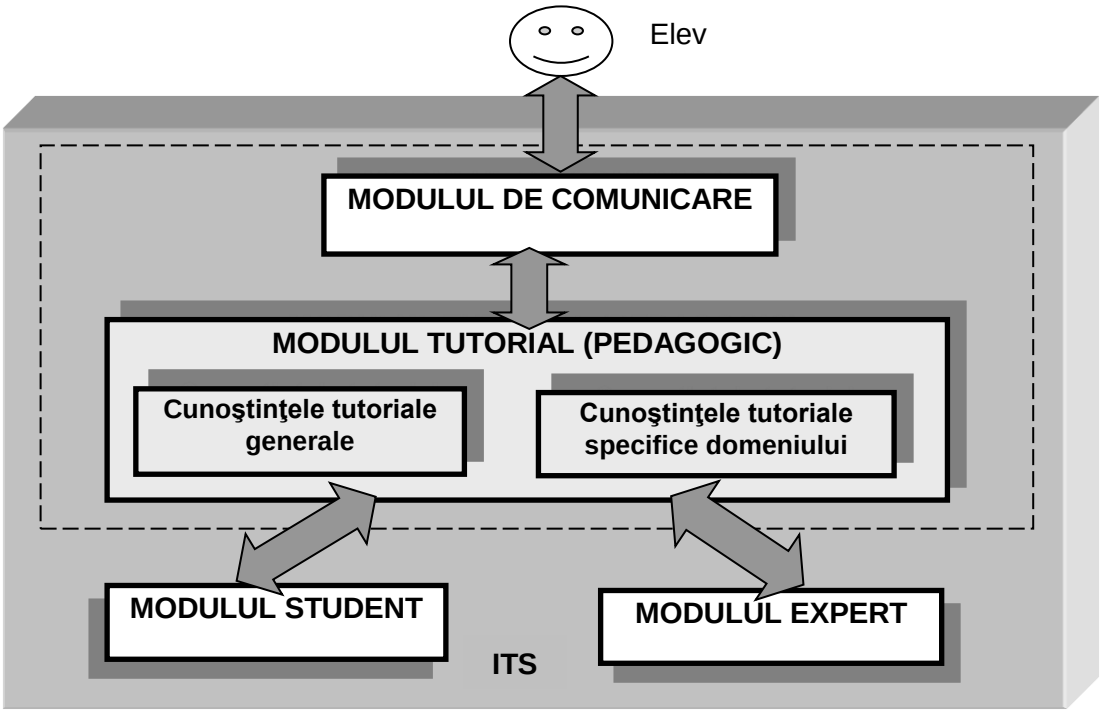


Figura 1-6 Arhitectura modulară a unui ITS

raţionament. Din această perspectivă, cunoaşterea domeniului trebuie să includă atât cunoştinţe structurale (*declarative*), cât şi *procedurale* (despre modul de rezolvare a problemelor). Deasemenea, modelul domeniului poate conţine *cunoştinţe de suprafaţă* (*compilate*) – ca euristici, reguli - sau *cunoştinţe de adâncime (articulate),* necesare atât pentru a putea răspunde elevului la întrebări de genul *"De ce?"* sau *"Cum?"*, dar şi pentru a putea compara cunoştinţele elevului cu cele ale domeniului.

Alegerea formalismului de reprezentare a cunoaşterii domeniului depinde atât de tipurile de cunoştinţe care vor fi modelate, cât şi de modul în care acestea vor fi utilizate (deci de strategiile tutoriale vizate). Formalismele de reprezentare a cunoaşterii domeniului - utilizate de-a lungul istoriei dezvoltării sistemelor inteligente reprezentarea prin reţele semantice şi cadre (*SCHOLAR, SOPHIE, BIP, PROUST*), reprezentarea prin reguli de producţie (*Geometry Tutor, BUGGY, GUIDON*) sau reprezentarea prin script-uri (*WHY*).

Modelul elevului

Modelul elevului trebuie să furnizeze sistemului de instruire informații despre starea cunoştinţelor (un model epistemologic) pe care le are elevul la un moment dat şi istoricul interacţinilor cu sistemul. Cunoştinţele despre elev (mai precis, cele despre procesele cognitive ale acestuia) sunt esenţiale pentru studiul procesului de înţelegere a subiectelor prezentate, cât şi pentru raţionamentele referitoare la capabilităţile elevului. Modelul elevului oferă un mecanism de bază în comunicarea individualizată între mediu şi sistem, reprezentând o sursă

importantă de adaptibilitate a sistemului [38]. Caracteristică pentru modelul elevului este ***dinamicitatea*** acestuia.

În majoritatea sistemelor ITS existente, baza de cunoştinţe referiroare la elev este construită în strânsă conexiune cu baza de cunoştinţe ale domeniului [39], [40].

- Abordarea bazată pe premiza ***submulţimii cunoştinţelor (al expertizei parţiale)***, care consideră că baza de cunoştinţe ale elevului – complet reprezentată - este o submulţime a bazei cunoştinţelor domeniului (***"modelul overlay"*** (engl.)) (figura 1-7).
  Abordarea overlay este *adecvată* în situaţiile în care scopul sistemului este de acela de a împărtăşi studentului cunoştinţele domeniului [41], iar cunoştinţele domeniului sunt exhaustive şi suficient de simple.
  Dezavantajele acestui model sunt imposiilitatea de a lua în considerare cunoştinţe ale elevului neconforme cu cele din baza de cunoştinţe ale domeniului și funcționarea incorectă a sistemului atunci când sunt necesare cunoştinţe de adâncime.

- **Abordarea diferenţială**

Este o variantă a modelului overlay, cu deosebirea că nu se compară cunoştinţele elevului cu cele ale domeniului, ci *performanţele elevului* cu *cele ale expertului*. Cunoştinţele din baza de cunoştinţe ale domeniului care nu sunt în modelul elev, pot fi împărţite în două categorii: *cunoştinţe pe care elevul nu le are* şi *cunoştinţe pe care acesta le-ar putea avea*. Această soluţie rafinează abordarea anterioară, dar suferă, în principal, de aceleaşi deficienţe.

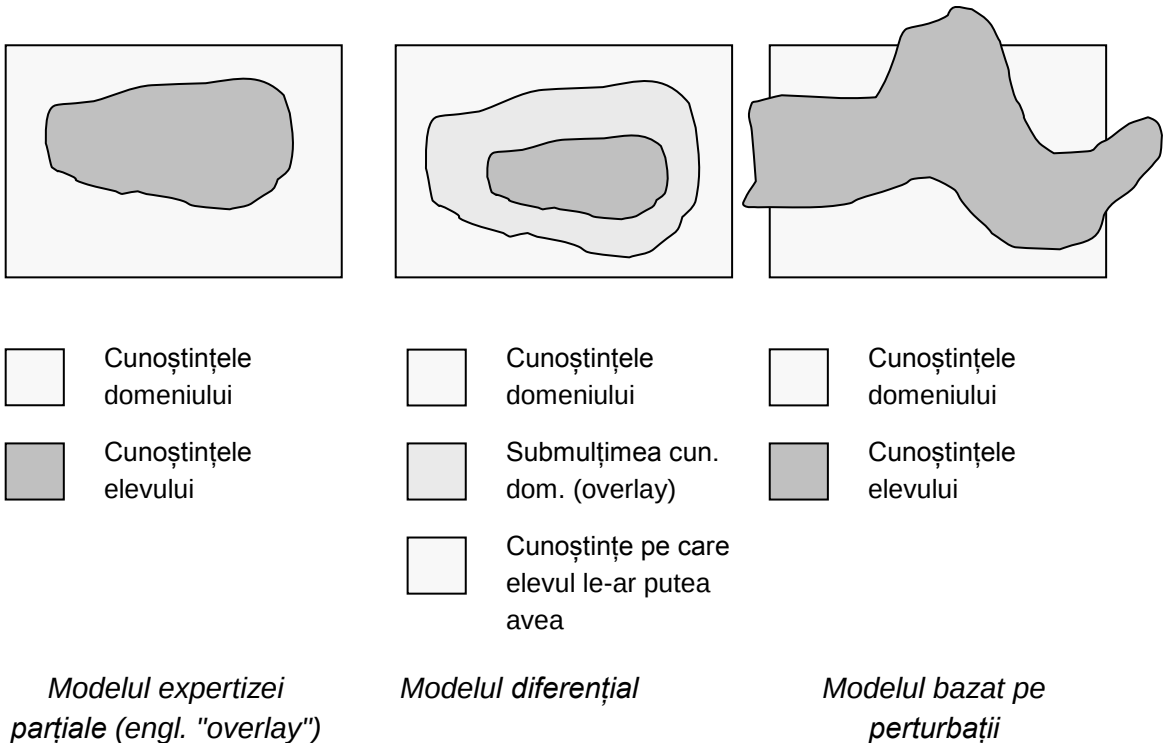


Figura 1-7 Abordări în construirea modelului elevului

- **Abordarea bazată pe perturbaţii**

Este abordarea cea mai complexă şi cea mai apropiată de realitate, eficientă şi în învăţarea cunoaşterii procedurale, în care baza de cunoştinţe ale elevului poate conţine şi cunoştinţe neincluse în baza de cunoştinţe ale domeniului. O comportare eronată a studentului se poate

datora lipsei de cunoştinţe (ca în abordările anterioare) sau utilizării unor cunoştinţe neadecvate sau eronate. În această abordare, trebuie avut în vedere faptul că, în realitate, într-un proces educaţional, când profesorul şi elevul nu au cunoştinţe comune, profesorul îşi va schimba modelul asupra elevului, încercând să realizeze comunicarea într-un alt mod. În acest caz, baza de cunoştinţe poate fi un set de proceduri care descriu comportamentul elevului (engl. ***"modelul buggy"***). Erorile pe care le poate comite elevul în rezolvarea unei probleme pot fi reprezentate sub forma unei *simple liste* (engl. "bug list") sau sub forma unor *structuri* care să permită clasificări ale erorilor (cataloage ale erorilor) sau să permită nu numai explicarea erorilor observate în comportamentul elevului, dar şi predicţii asupra tipurilor de erori care pot apare în anumite contexte (engl. ***"repair theory"***).

În funcţie de *scopul temporal* urmărit la crearea modelului elev, se pot evidenţia *modelul "instantaneu", de scurtă durată* (engl. ***"snapshot"***) - pentru urmărirea evoluţiei elevului pe parcursul unei singure sesiuni de lucru – și *modelul de lungă durată* (engl. ***"long-term"***), al evoluţiei pe o perioadă de timp mai lungă (ca durată a tuturor sesiunilor de lucru).

În funcţie de *profunzimea modelului*, se deosebesc *modele de suprafaţă* (engl. ***"shallow"***) și *modelele de adâncime* (engl. ***"deep"***).

Cele mai utilizate formalisme de reprezentare a cunoaşterii pentru modelul elev sunt reţelele neuronale, reţelele bayesiene, grafurile genetice, reţelele semantice și tehnicile fuzzy.

Modelul elev poate fi privit ca *model de comportament în rezolvarea problemelor*, ca *model de diagnosticare automată* [42] sau ca model pentru construirea unui *agent elev artificial* în sistemele sociale de instruire.

În urma acestei analize, *concluziile* privind construirea unui model al elevului cât mai bun, sunt [43]:

- Analiza modelul elevului trebuie făcută în strânsă legătură cu mediul de instruire în care va funcţiona sistemul. Este necesar să fie luate în considerare cunoştinţele de adâncime, inclusiv cele referitoare la psihologia elevului. Se preconizează introducerea în modelul elev a unor cunoştinţe care să individualizeze profilul elevului şi care, împreună cu cunoştinţele pedagogice, pot conduce la găsirea celor mai adecvate regimuri de instruire;
- Regimurile de funcţionare normală ale unui ITS trebuie să alterneze cu regimuri de achiziţie a cunoştinţelor şi de învăţare automată, care să permită dezvoltarea şi rafinarea bazelor de cunostinţe implicate. Analiza succeselor şi a eşecurilor unei sesiuni de lucru ar putea declanşa mecanisme de învăţare automată, cu efect asupra cunoştinţelor referitoare la tacticile şi strategiile pedagogice. Deasemenea, un model iniţial, rudimentar, al elevului ar putea fi construit prin tehnici specifice achiziţiei de cunoştinţe.

Modelul tutorial

Modelul tutorial are un rol central într-un ITS, deoarece acest modul hotărăşte *"CE"*, *"CÂND"* şi *"CUM"* i se va prezenta elevului. Cunoştinţele tutoriale sunt descrise în *termeni ai conţinutului* (subiecte, exemple, indicaţii, întrebări, etc.) şi în ***termeni ai strategiei aplicate într-un anumit context*** (când intervine sistemul, care este tipul intervenţiei) [44]).

Aspectele tratate de către modulul tutorial sunt diverse şi deosebit de complexe: conduita pedagogică a sesiunii, gestiunea curriculum-lui, urmărirea activităţii elevului, deciziile şi conţinutul intervenţiilor sistemului.

Au fost propuse diferite stiluri de ghidare şi de control, explicitarea planurilor pedagogice, reguli şi strategii tutoriale şi diferite tehnici de realizare. Expertiza didactică este responsablă de selecţia unui anumit subiect şi modul de prezentare a acestuia, ***raportate la cunoştinţele domeniului şi cele ale elevului*** [45]. Deasemenea, strategia tutorială trebuie să urmărească o serie de *obiective* clar definite pentru fiecare unitate de predare.

Categoriile de aspecte vizate de modelul tutorial sunt (figura 1-8.):

- Aspectele didactice și de adaptare;
- Diagnoza elevului;
- Gradul de control al elevului.

În ceea ce privește gradul de control al elevului, deși cele mai multe sisteme folosesc monitorizarea (controlul fiind deținut de către sistem), nu putem să nu evidențiem importanța dialogului socratic şi a meditării la cerere. Prin *dialogul socratic* (*SCHOLAR, WHY*) - un dialog cu iniţiativă mixtă - sistemul angajează elevul într-un dialog care să îi permită descoperirea proprie a erorilor şi a cunoştinţelor contradictorii; deciziile pedagogice sunt luate în funcţie de rezultatele dialogului. Dialogul socratic este eficient atât pentru prezentarea cunoştinţelor factuale, cât şi pentru învăţarea de reguli, principii sau rezolvarea problemelor.

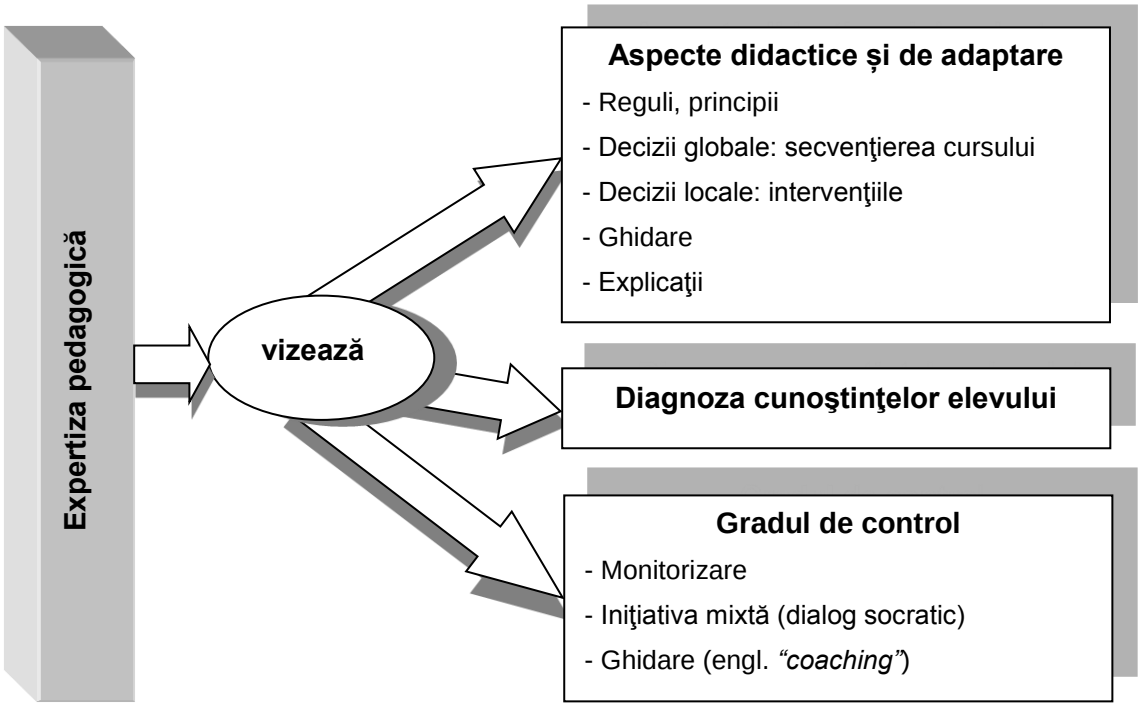


Figura 1-8 Aspecte vizate de expertiza pedagogică

Meditarea la cerere, ghidare (*"coaching"*), permite elevului să deprindă cunoştinţele procedurale prin intermediul exemplelor, al exerciţiilor şi al problemelor. Sistemul observă comportamentul elevului pe parcursul sesiunii de rezolvare a problemei; în funcţiie de rezultatele acestor observaţii, ia decizii de întrerupere a elevului în anumite momente, cu scopul de a-i prezenta cunoştinţe ajutătoare sau de a-i sugera noi strategii de rezolvare.

Modelul de comunicare

Tratată ca o componentă *separată* sau *integrată* în modulul tutorial, interfaţa joacă un rol important prin faptul că activităţile şi formele de comunicare propuse elevului sunt elemente determinante pentru învăţare, susţinând dialogul sistem-elev. Cele mai multe dintre intefeţele existente folosesc un limbaj de comandă bazat pe meniuri, iconiţe şi reprezentări grafice, puţine tutoriale putând susţine un dialog în limbaj natural [46]. Direcţiile celor mai recente cercetări sunt interfeţele multi-modale, comunicarea în limbaj natural, modalităţi de reprezentare a dialogului.

1.2.3 ALTE ASPECTE ALE SISTEMELOR EDUCAȚIONALE

Ideile fundamentale care trebuie urmărite la conceperea şi implementarea unui sistem inteligent de instruire sunt:

- Reprezentarea explicită a cunoştinţelor domeniului şi a mecanismelor de raţionament, care conferă sistemului capacitatea de a răspunde la întrebări şi de a rezolva exerciţii ale căror soluţii nu au fost prevăzute explicit sau ale căror enunţuri au fost propuse de elev;
- Explicitarea strategiilor tutoriale, pentru a permite sistemului să intervină dinamic în funcţie de o anumită situaţie, de obiectivele pedagogice şi de modelul elevului;
- Comunicaţii suple şi variate, cu posibilitatea de intervenţie şi iniţiativă a elevului;
- Separarea reprezentării conţinutului instruirii şi a strategiilor de instruire;
- Modularizarea conţinutului instruirii în vederea utilizării multiple sau a reutilizării;
- Crearea unor strategii de predare şi de îndrumare generice, care să poată fi utilizate pentru diferite conţinuturi instrucţionale;
- *Modelarea instruirii trebuie să fie un proces ingineresc, care să aibă în vedere reprezentarea explicită a entităţilor pedagogice abstracte, deci proiectarea la nivel pedagogic şi nu la nivelul mediului fizic*.

TIPURI DE SISTEME EDUCAȚIONALE

Datorită diversităţii şi combinării abordărilor, este greu de stabilit o tipologie a sistemelor inteligente de instruire. Voi schiţa, totuşi, posibile categorii de sisteme inteligente de instruire, după diferite criterii.

Din punct de vedere *a abordării psihologice a învăţării*, sistemele pot fi clasificate în (1) ***sisteme obiectiviste*** (*SCHOLAR, BIP, EXCHECK*) și (2) **sisteme *constructiviste clasice*** (*LOGO*, *MEMOLAB) sau* ***inteligente interactiviste*** **şi *bazate pe agenţi pedagogici*** (*STEVE, ADELE).*

Din punct de vedere a *strategiilor de învăţare*, sistemele de instruire inteligente pot fi (1) **sisteme *socratice*** (*SCHOLAR, WHY*), (2) **sisteme *procedurale***, bazate pe strategia de ghidare (engl. ”*coach”*) (*WEST*), (3) ***medii interactive*** de învăţare (ILE - Interactive Learning Environment) (*MIDI Draw),* (4) ***micro-lumi sau medii de demonstraţie*** (engl. *“micro-world”*), bazate pe simulări sau (5) ***medii interactive de învăţare şi simulare*** (*STEAMER, BIOTECH*).
Datorită importanţei deosebite pe care o au sistemele de instruire în care se folosesc simulările, unii cercetători propun ca perioada 1980-1990 să fie considerată *perioada ITS-lor*, iar perioada de după 1990 să fie considerată *perioada mediilor interactive de învăţare cu ajutorul calculatorului*.

O categorie importantă o reprezintă ***sistemele de e-learning*** (fie clasice, fie inteligente). *Sistemele de management al învățării* (*LMS*, engl. "Learning Management Systems") și *sistemele de management al conținutului* (*CMS*, engl. "Content Management Systems") și-au dovedit din plin utilitatea și sunt folosite pe scară largă. Un studiu comparativ, cât și propunerea unor metode de evaluare a caracteristicilor unor astfel de sisteme *(+CMS, Moodle, ATutor, Claroline, Dokeos, dotLRN/OpenACS, Drupal, ILIAS, LON-CAPA, Mambo, OLAT, Plone și Sakai*), găsim în [47] [48] și [49].

Sisteme hypermedia adaptive

Avantajele utilizării în procesul instruirii a unor sisteme software hibride (ITS+HMS), cu scopul *individualizării instruirii* și **Web-ul semantic** (Ontologiile aplicate Web-ului au creat Web-ul semantic au condus la "consacrarea" unor noi tipuri de *sisteme inteligente educaționale*, *bazate pe WEB* [50]:

- sistemele hypermedia adaptive[6] (AHS);
- sistemele hypermedia adaptive deschise[7] (OAHS) .

Un studiu detaliat al aportului și problemelor identificate în utilizarea Web-ului semantic în sistemele educaționale este în [51].

Un sistem hypermedia adaptiv este un sistem hypertext şi hypermedia care, pe baza unui model al utilizatorului, adaptează diverse aspecte în funcţie de utilizator. Aplicate în domeniul instruirii, sistemele (O)AHS aplică *strategii de personalizare şi adaptare* a materialelor (educaţionale). În plus, Web-ul semantic oferă sistemelor posibilităţile necesare în înţelegerea, prelucrarea şi raţionamentul asupra cererilor unui utilizator (furnizându-i materialele existente pe Web, adaptate utilizatorului) [52], [53]. Sistemele AHS au preluat de la ITS ideile de *secvenţiere adaptivă* a cursului (conţinut şi/sau prezentare) [54], *sprijin în rezolvarea problemelor*, *diagnoza* elevului [55], [56], *generarea feedback-ului adaptiv*, *testare adaptivă*, *managementul adaptiv al dialogului* sistem-utilizator. Un studiu al metodelor de personalizare folosite în sistemele AHS găsim în [57].

Similarităţile dintre tutorialele inteligente "clasice" (ITS, ITS+HMS) şi sistemele (O)AHS se regăsesc la atât la nivelul scopului, cât şi la cel al arhitecturii şi al implementării. Sistemele (O)AHS folosesc *tehnici de inteligenţă artificială* și extind *modelul elevului* la *modelul utilizatorului* (care ia în considerare nu doar caracteristicile cognitive sau afective ale acestuia, ci și pe cele personale, preferințe, etc.) și *modelul tutorial* la *modelul de adaptare*. Ca și ITS-urile, sistemele (O)AHS urmăresc individualizarea instruirii şi furnizarea de materiale educaţionale adecvate utilizatorului.

De aceea, aceste tipuri de sisteme sunt desemnate prin sintagmele ***sisteme educaţionale bazate pe web*** ("Web-based Educational system") sau, simplu, ***sisteme educaţionale inteligente***, fiind structurate în *modelul domeniului*, *modelul utilizatorului* şi *modelul de aplicare*. Pentru proiectarea acestor sisteme, au fost propuse sistemele autor web, hypermedia inteligente şi adaptive, care se remarcă prin **orientarea către pedagogie**.

[6] Sistem hypermedia adaptiv (AHS) - sistem hypermedia care aplică o serie de metode de personalizare şi adaptare asupra materialelor solicitate de către un utilizator

[7] Sistem hypermedia adaptiv deschis (OAHS) - AHS care operează asupra unei mulţimi "deschise" de documente din web

SISTEME SOCIALE DE ÎNVĂȚARE ŞI STRATEGII PEDAGOGICE DE GRUP

În sistemele ITS "tradiţionale", sistemul este un agent artificial care se comportă ca un tutor uman inteligent și adaptează instruirea la ritmul de învăţare, la cunoştinţele anterioare, la reacţiile şi progresele realizate de către elev. În aceste sisteme, controlul îl deţine sistemul, interacțiunea este 1-1, elevul doar asimilând cunoștințele predate de către expert (întotdeauna, expertul are mai multe cunoştinţe decât elevul). Strategia pedagogică folosită este numită ***strategie pedagogică directă*** (***tutor-tutee***).

Inspirate din dialogul socratic şi din cercetările didactice, strategiile având la bază noţiunile de *negociere şi argumentare* capătă o importanţă deosebită. Ca urmare, o altă categorie de ITS-uri adoptă o *abordare bazată pe cooperarea între elev şi sistem* sau *cooperarea între mai mulţi agenţi participanţi la procesul de instruire*. În astfel de sisteme (bazate pe psihologia constructivistă şi/sau situaţionistă), învăţarea este privită ca proces de asociere a construcţiilor cunoştinţelor în timpul interacţiunii cu ceilalţi participanţi la un grup de activitate, nu doar ca proces de transfer al expertizei de la profesor la elev [58]. Aceste sisteme în care intervin şi alţi agenţi artificiali (în afara "profesorului clasic artificial") sunt numite **sisteme sociale de învățare** și folosesc o ***strategie pedagogică de grup***.

O astfel de strategie de grup este ***strategia learning companion*** - indicată de Self şi introdusă de Chan [59] - constă în apariţia *celui de-al treilea agent, un elev artificial*, simulat (engl. "*peer tutoring"*, *elev tovarăş*), care acompaniază elevul uman. În funcţie de *nivelul cunoştinţelor* şi de *rolul* celui de-al treilea agent, se pot evidenţia situaţiile în care (1) elevul şi *tovarăşul (*engl. *"co-learner")* său au obiective şi nivel de cunoştinţe similare, lucrează împreună, analizează soluţia celuilalt, tutorul prezintând problemele şi criticând soluţiile elevilor, (2) elevul şi *partenerul* (engl. "*companion*") au obiective similare, nivelul cunoştinţelor partenerului este mai înalt ca cel al elevului, dar mai scăzut ca cel al tutor-ului, elevul observă demonstraţia realizată de partenerul său sau de tutor, lucrează sub îndrumarea partenerului sau al tutorului, (3) elevul explică și predă tovarăşului său (engl. *"inverted tutor*"), prin *auto-explicare*, structurându-și propriile cunoştinţe și (4) un agent-elev perturbator ("engl. *troublemaker*") prezintă o demonstrație corectă sau incorectă, dă sfaturi corecte/incorecte, aplicând învăţarea prin perturbaţii (engl. "*learning by disturbing"*).

Strategiile pedagogice de grup se aplică în *sistemele sociale de învăţare (cooperative sau colaborative)*.

- *Colaborarea* implică asocierea acţiunilor participanţilor şi înţelegerea mutuală a sarcinii pe care o au de executat, fiecare participant având propriile obiective.
- *Cooperarea* implică împărţirea responsabilităţilor între participanţi, în vederea executării unei anumite sarcini.

În sistemele sociale de învăţare, *interacţiunile* şi *contextul* în care acesta au loc sunt extrem de importante [60]. Principalele avantajele ale sistemelor sociale de învăţare sunt [61]:

- Motivează elevul uman, stimulează şi eficientizează într-o mai mare măsură procesele de învăţare. Astfel, *învăţarea prin predare* facilitează învăţarea prin externalizarea înţelegerii elevului uman; *învăţarea prin diagnosticarea soluţiei altui elev* favorizează acumularea unor cunoştinţe de adâncime; *învăţarea prin discuţii* deschise facilitează capabilităţile de gândire prin interacţiune;
- Contribuie la clarificarea aspectelor legate de dialog şi negociere [62] (aspect interesant, dar deocamdată ambiguu, care necesită înţelegerea limbajului natural) [63].

AGENŢI ÎN SISTEMELE DE INSTRUIRE

Problemele complexe pe care le ridică un sistem inteligent de instruire (un triplu sistem expert) a condus la noua paradigmă de **"sisteme tutoriale inteligente distribuite"** (**D**istributed **I**ntelligent **T**utoring **S**ystems - **DITS**), derivată din cea de "inteligenţă artificială distribuită", în care soluţia unei probleme este dată de rezultatul cooperării între agenţii autonomi rezolvitori de probleme [64]. În cazul sistemelor educaţionale inteligente distribuite, cercetările vizează – în principiu – aceleaşi direcţii ca şi în cazul inteligenţei artificiale distribuite (figura 1-9).

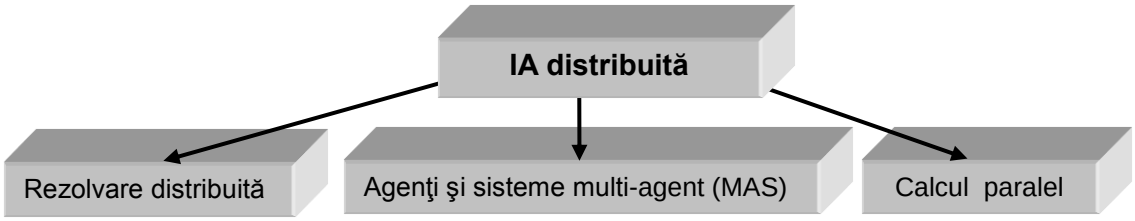


Figura 1-9 Sisteme educaţionale inteligente distribuite – Direcţii de cercetare

Arhitectura standard a unui ITS - modulul expert, modulul pedagogic, modulul elev şi interfaţa - simulează o singură entitate, dar o entitate inteligentă, ale cărei funcţii sunt realizate, de obicei, secvenţial. În învăţarea distribuită, funcţiile (ca de exemplu prezentarea conţinutului instrucţional, propunerea exerciţiilor şi problemelor, rezolvarea lor şi furnizarea explicaţiilor, etc.) sunt alocate diferiţilor agenţi [65]. Abordarea prin prisma agenţilor autonomi [66] prezintă un interes deosebit pentru sistemele artificiale de predare [67], care îşi propun să execute în paralel numeroase task-uri.

Agenţii pedagogici [68] – reprezintă un tip particular al agenţilor inteligenţi și sunt utilizaţi în sistemele de instruire (*"multi-agent"* sau nu), atât în *strategia pedagogică tutor-tutee*, cât şi în cele *sociale* [69]. *Agenţii pedagogici* sunt definiţi ca agenţi autonomi care îşi adaptează interacţiunile instrucţionale în funcţie de elev şi de starea mediului şi ajută elevul în depăşirea dificultăţilor; colaborează cu elevul sau cu alţi agenţi [70]; pot deţine capacitatea de a răspunde afectiv, socio-emoţional, elevului [71], [72];

În funcţie de complexitate, agenţii pedagogici îmbracă diferite forme [73]: *table-driven agent* (bazaţi pe paradigma perechii percepţie/acţiune); *agenţi reactivi* simplu reflex (bazaţi pe reguli condiţie-acţiune); *agenţi care păstrează urma* (o stare internă a situaţiilor mediului pe care o compară şi decid, în funcţie de aceasta, care va fi acţiunea următoare); *agenţi orientaţi către scop*, cu posibilitatea găsirii unor soluţii diferite de atingere a scopului şi de selectare a uneia dintre ele, pe baza satisfacerii rezultatului [74]; *agenţi orientaţi către utilitate*, care selectează o anumită acţiune pe baza maximizării utilităţii pentru elev (parcurgerea unui curs într-un timp cât mai scurt, dobândirea cât mai multor cunoştinţe despre un anumit subiect, etc.) [75].

Exemple de agenți pedagogici sunt cei din sistemul care implementează ***strategia learning companion*** [59], *agentul emoţional* ***Émile*** [76], *agentul* ***pedagogic*** *din sistemul* ***multi-agent LANCA*** - îmbunătăţeşte planuri, acţiuni, rezultate, informează studentul şi ceilalţi agenţi, este adaptiv, instruibil, cognitiv şi învață din experienţă [77], agentul ***Peddy*** - capabilităţi de recunoaştere şi sinteză a vocii [78], ***Einstein*** - personaj animat care răspunde în limbaj natural la întrebările utilizatorului, ***"Herman, the bug"*** - care ajută elevul în rezolvarea problemelor [79], agentul ***Cosmo*** - combină dinamic gesturi, mişcare şi vorbire, ajută și încurajează elevul în conectarea calculatoarelor într-o reţea, traficul datelor şi routare, **Steve** - agent hibrid [80] de

instruire în funcţionarea unor echipamente, **Adele** [81] - domeniul medical, operează în web, format din *agentul pedagogic propriu-zis (motorul de inferenţă* pentru luarea deciziilor), *persoana animată și simulator.*

ARHITECTURI MULTI-AGENT ȘI ORIENTATE PE SERVICII

Considerând că arhitectura unui ITS (prezentată în secțiunea 1.2.2) nu oferă suficientă modularitate, o serie de cercetări s-au focalizat pe *arhitecturi multi-agent* (din inteligența artificială) sau *arhitecturile orientate pe servicii* (din ingineria software), în care modulele tradiționale sunt implementate ca mulțimi de servicii sau agenți. Lavendelis [65] propune pentru dezvoltarea unui ITS o *arhitectură multi-agent, holonică*[8]*, deschisă*.

1.3 METODE ȘI INSTRUMENTE PENTRU PROIECTAREA SISTEMELOR DE INSTRUIRE

Având în vedere multitudinea de probleme pe care le implică dezvoltarea sistemelor de instruire inteligente, direcțiile principale de cercetare sunt conceperea unor sisteme/medii autor de dezvoltare, utilizarea standardelor, propunerea unor limbaje de modelare educațională sau a unor scenarii pedagogice.

1.3.1 SISTEME/MEDII DE DEZVOLTARE DE TIP AUTOR

Conceperea unor sisteme autor care să susțină în mod real activitatea de proiectare pedagogică a unui sistem educațional inteligent este una dintre cele mai importante direcții de cercetare. Primele sisteme apărute cu scopul de a susține proiectarea unui sistem de instruire sunt **sistemele/mediile autor de dezvoltare** (urmaşe ale *sistemelor de management al instruirii*), numite şi **sisteme de proiectare pedagogică**.

Majoritatea sistemelor autor de dezvoltare (engl.*"authoring system"*) sunt *sisteme expert*, fiind formate dintr-un nucleu (engl. "shell") ITS şi o interfaţă prin care utilizatorul (profesor, pedagog) poate să formalizeze, vizualizeze şi raţioneze asupra cunoştinţelelor din baza de cunoştinţe. Doarece modelul expertizei într-un ITS este format din *modelul domeniului*, *modelul student*, *modelul pedagogic* şi *modelul de comunicare* (al interfeţei), trebuie să existe instrumente de dezvoltare autor pentru conţinut, modelul student, strategii de instruire şi proiectarea interfeţei [82]. În momentul actual există o mare varietate de sisteme autor (în figura 1-10 se prezintă tipurile celor mai cunoscute sisteme, facilităţile oferite, metodele utilizate, caracteristicile şi limitările în capabilităţile de dezvoltare a sistemelor de instruire), care pot fi clasificate în diferite categorii, în funcţie de tipurile tutorialelor care pot fi dezvoltate cu aceste sisteme.

O altă clasificare a sistemelor autor este cea propusă de Murray [83]:

- Sisteme autor orientate către modelarea pedagogică (numite "*sisteme autor orientate către pedagogie*");
- Sisteme autor orientate către crearea de ITS-uri care concretizează modelele pedagogice (numite "*sisteme autor orientate către performanță*").

[8] Holon (grec. "holos" = întreg și "-on" = parte) - Termenul desemnează un obiect care este în același timp și parte, și întreg.

1.3.2 STANDARDE ÎN INSTRUIREA ASISTATĂ

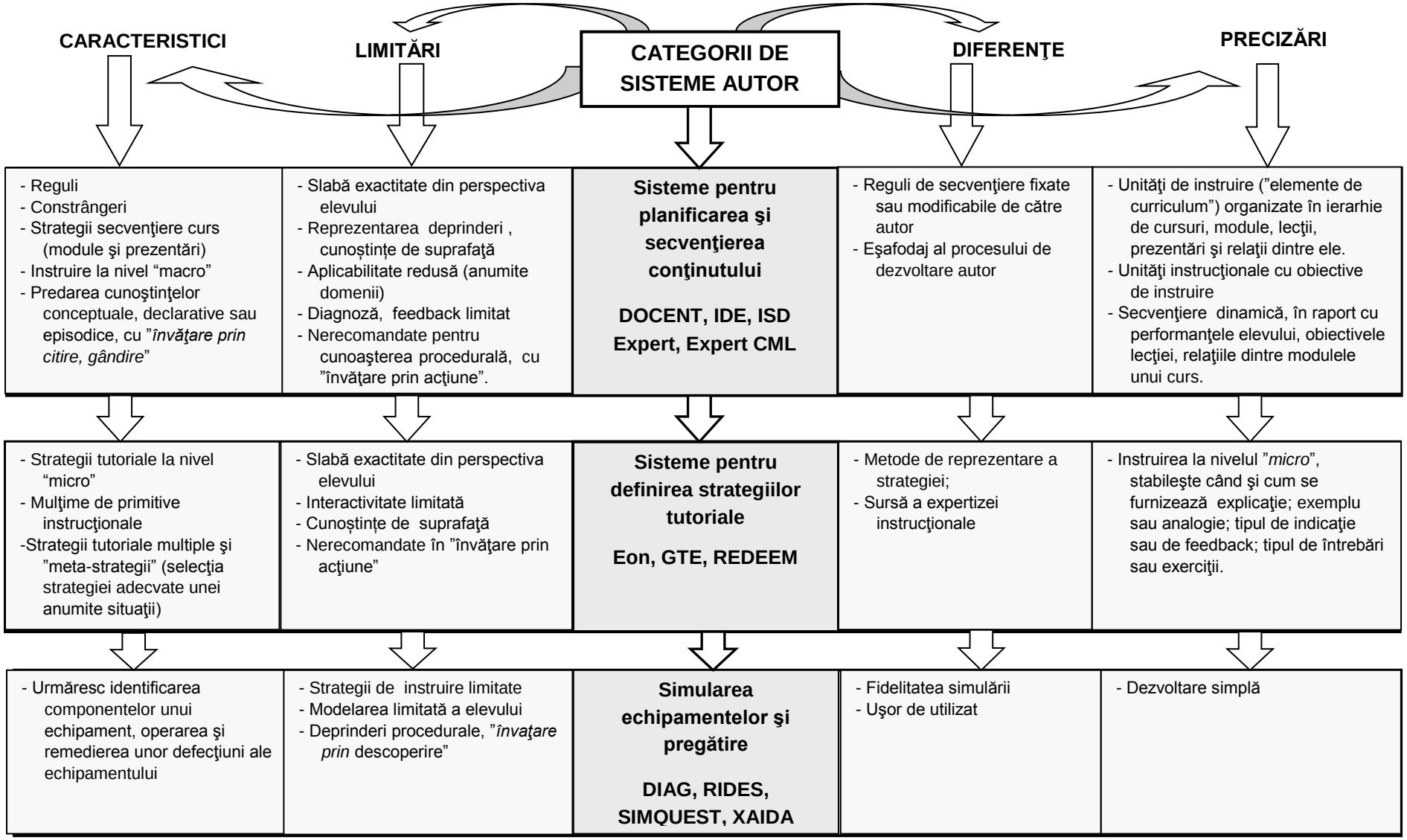
Metadatele şi standardele de interoperabilitate joacă un rol important în dezvoltarea şi descrierea sistemelor educaţionale, prin potenţialul oferit în partajarea şi reutilizarea acestora în diferite contexte (domenii, situaţii, etc.). Principalele *organizaţii şi grupuri de lucru* implicate în activitatea de standardizare sunt:

- *DCMI* (*Dublin Core Metadata Initiative*), care din 1995 se preocupă de metadate. Grupul de lucru în domeniul educaţiei (*DCMI Education Working Group*), interesat de stocarea, căutarea şi regăsirea resurselor web, a elaborat 15 categorii de atribute de bază (titlu, creator, descriere, editor, responsabil_conţinut, dată, tip_subiect, format, identificator, tipul resursei, drepturi, etc.). A propus utilizarea metadatelor în descrierea resurselor educaţionale [84], iar în 1999 a început elaborarea unor extensii pentru resursele pedagogice [85]. Temele de lucru din ultimii ani sunt *aplicațiile independente de platformă* (cu propunerea în 2007 a unui model abstract DCMI (DCAM)), *mapări între diferite vocabulare*, *vocabulare sustenabile*.

- *LOM* (*Learning Object Metadata*) propune descrierea obiectelor instrucţionale prin nouă categorii de atribute (unele obligatorii, altele opţionale). Prin obiect instrucţional (engl., ”Learning Object”) se înţelege orice entitate (digitală sau nedigitală) care poate fi utilizată, reutilizată sau referită în instruire [86]. Conţinutul multimedia, conţinutul instrucţional, obiectivele instruirii, mijloacele software, persoanele, organizaţiile sau evenimentele implicate în instruire constituie exemple de obiecte instrucţionale. Cele 60 de atribute ale obiectelor instrucţionale sunt împărţite în nouă categorii (tipul_obiectului_instrucţional, autor, termeni_de_distribuire, format, etc.), încercându-se stabilirea unei corespondenţe cu atributele DCMI şi specificarea în RDF [87].

- *ARIADNE* (Alliance for Remote Instructional and Authoring and Distribution Networks for Europe) se ocupă de [88]:
  - Elaborarea unor recomandări pentru metadatele educaţionale în urma implementării specificaţiilor LOM [89], [90].
  - Elaborarea de instrumente, protocoale şi metodologii de stocare, management şi reutilizare a componentelor curriculare mediate de calculator [91].

- *IMS Global Learning Consortium* (iniţial *Instructional Management System Project*), a propus crearea unor arhitecturi deschise şi a infrastructurii pentru tehnologia instruirii, elaborarea specificaţiilor tehnice pentru interoperabilitatea aplicaţiilor şi a serviciilor sistemelor educaţionale distribuite bazate pe web. În mare parte, *specificaţiile* descriu conţinutul pedagogic utilizat în instruirea online, urmărind aspecte precum [92]:
  - Specificaţii pentru metadatele de descriere a resurselor pedagogice; specificaţiile sunt strâns legate de LOM - *IMS-Meta-Data*;
  - Proiectarea instruirii – *IMS-LD* (*Learning Design*);
  - Secvenţierea curricumul-ului – *IMS-SS* (*Simple Sequencing*);
  - Structura şi modul de organizare a fişierelor în pachete, în vederea facilitării interschimbului - *IMS-CP* (*Content Packaging*);
  - Metodele, tipul şi conţinutul testelor de evaluare *IMS-QTI* (*Question & Test Interoperability*) [93];
  - Specificaţii despre elev IMS-LIP (Learner-Information Package).


CARACTERISTICI
LIMITĂRI
CATEGORII DE SISTEME AUTOR
DIFERENŢE
PRECIZĂRI
- Reguli
- Constrângeri
- Strategii secvenţiere curs (module şi prezentări)
- Instruire la nivel “macro”
- Predarea cunoştinţelor conceptuale, declarative sau episodice, cu ”*învăţare prin citire, gândire*”
- Slabă exactitate din perspectiva elevului
- Reprezentarea deprinderi , cunoștințe de suprafaţă
- Aplicabilitate redusă (anumite domenii)
- Diagnoză, feedback limitat
- Nerecomandate pentru cunoaşterea procedurală, cu ”învăţare prin acţiune”.
Sisteme pentru planificarea şi secvenţierea conţinutului
DOCENT, IDE, ISD Expert, Expert CML
- Reguli de secvenţiere fixate sau modificabile de către autor
- Eşafodaj al procesului de dezvoltare autor
- Unităţi de instruire (”elemente de curriculum”) organizate în ierarhie de cursuri, module, lecţii, prezentări şi relaţii dintre ele.
- Unităţi instrucţionale cu obiective de instruire
- Secvenţiere dinamică, în raport cu performanţele elevului, obiectivele lecţiei, relaţiile dintre modulele unui curs.
- Strategii tutoriale la nivel “micro”
- Mulţime de primitive instrucţionale
-Strategii tutoriale multiple şi ”meta-strategii” (selecţia strategiei adecvate unei anumite situaţii)
- Slabă exactitate din perspectiva elevului
- Interactivitate limitată
- Cunoștințe de suprafaţă
- Nerecomandate în ”învăţare prin acţiune”
Sisteme pentru definirea strategiilor tutoriale
Eon, GTE, REDEEM
- Metode de reprezentare a strategiei;
- Sursă a expertizei instrucţionale
- Instruirea la nivelul ”*micro*”, stabileşte când şi cum se furnizează explicaţie; exemplu sau analogie; tipul de indicaţie sau de feedback; tipul de întrebări sau exerciţii.
- Urmăresc identificarea componentelor unui echipament, operarea şi remedierea unor defecţiuni ale echipamentului
- Strategii de instruire limitate
- Modelarea limitată a elevului
- Deprinderi procedurale, ”*învaţare prin* descoperire”
Simularea echipamentelor şi pregătire
DIAG, RIDES, SIMQUEST, XAIDA
- Fidelitatea simulării
- Uşor de utilizat
- Dezvoltare simplă

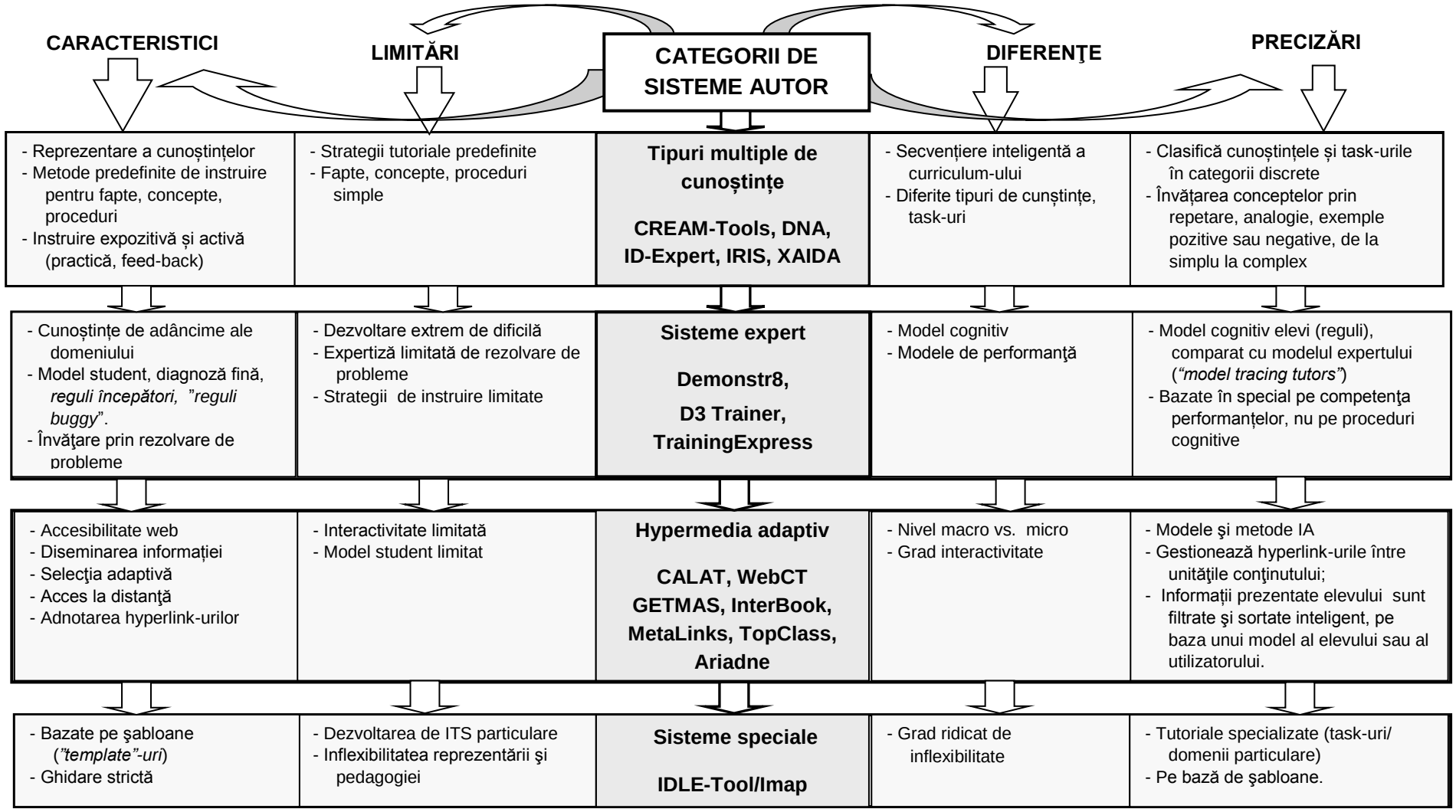


Figura 1-10 Sinteză - Principalele tipuri de sisteme autor şi caracteristicile acestora

- *ADL* (*Advanced Distributed Learning Initiative*) - format în 1997 din cercetători din Ministerul Apărării Americane şi Departamentul Ştiinţei şi Tehnologiei de la Casa Albă - se preocupă de reutilizarea şi accesibilitatea (de oriunde şi oricând) resurselor pedagogice din web și a propus modelul *SCORM* (*Sharable Content Object Reference Model*) [95].
  - *SCORM* se adresează autorilor conţinutului instrucţional, proiectanţilor şi distribuitorilor de sisteme de management al instruirii (LMS – Learning Management Systems). Oferă specificaţii bazate pe multiple surse (IMS, IEEE, ARIADNE), operabilitatea, accesibilitatea şi reutilizarea conţinutului instrucţional din web. Instrumentul pentru metadate – MDT (Metadata Tool) permite managementul metadatelor despre obiectele instrucţionale. Instrumentul CSF (extended Content Structure Format) permite reprezentarea standardizată a unui curs web. Proiectul *TinCan Api* își propune să ofere noi posibilități și schimbări în comunitatea e-learning.

- *AICC* (*Airline Industry Computer-based training Consortium*) – fondat în 1998 şi având ca scop standardizarea hardware-ului folosit în industria aeronautică – comută către standardizarea sistemelor de management a instruirii (LMS) [96].

- *GESTALT* (*Getting Educational Systems Talking Across Leading Edge Technologies*) este un proiect european pentru metadate şi arhitecturi utilizate în tehnologiile instrucţionale.

- PROMETEUS (PROmoting Multimedia Access to Education and Training in European Society) elaborează metadate pentru interoperabilitatea standardelor.

Principalele **organisme acreditate** în dezvoltarea standardelor în instruirea asistată sunt:
- **IEEE** (*Institute of Electrical and Electronic Engineering*) [97] prin comitetul LTSC (Learning Technologies Standardization Committee), compus din grupurile P1484.12 (autorul LOM), IMS, ARIADNE şi DCMI.
- **ISO** (*International Standard Organization*), care prin IJTC1 (International Joint Technology Committee) a creat grupul SC36 (Sub Committee 36 IT for learning Education and Training); lucrează în standardizarea vocabularului pentru resursele pedagogice (WG1/N0017), pentru spaţiile colaborative de lucru (WG2/N0022) şi pentru schemele de interacţiune student-student (WG2/N0024).
- **CEN** (*Comitetul European de Normalizare*) prin WS-LT (WorkShop Learning Technologies).

Cele trei organisme lucrează împreună, în vederea interoperabilităţii şi reutilizării resurselor pedagogice: CEN traduce specificaţiile LOM în limbile europene; IEEE propune standarde pentru specificaţiile LOM, standarde trimise apoi la ISO în vederea transformării lor în norme.

Practic, activitatea de standardizare se concentrează asupra următoarelor direcţii:
- Standarde generale, despre arhitectura sistemelor educaţionale şi vocabular;
- Standarde pentru obiecte instrucţionale;
- Standarde de management instrucţional;
- Standarde de conţinut instrucţional;
- Standarde pentru modelul elevului.

Aşa cum se observă din figura 1-11, cele mai multe activități de standardizare vizează conţinutul educaţional, *tratând sumar aspectele pedagogice*; tocmai de aceea şcoala japoneză a susţinut iniţial ideea că standardele îngrădesc progresul cercetării ştiinţifice privind sistemele educaţionale inteligente, fiind adepta dezvoltării ontologiilor. Cu toate acestea, standardele şi-au

dovedit utilitatea în descrierea, adnotarea și combinarea resurselor educaţionale existente pe web (în vederea obținerii unor materiale pedagogice adaptate utilizatorului). În această lucrare, folosim atât elemente din standardizare, cât și ontologii (studiile de caz din capitolul 3 și capitolul 4).

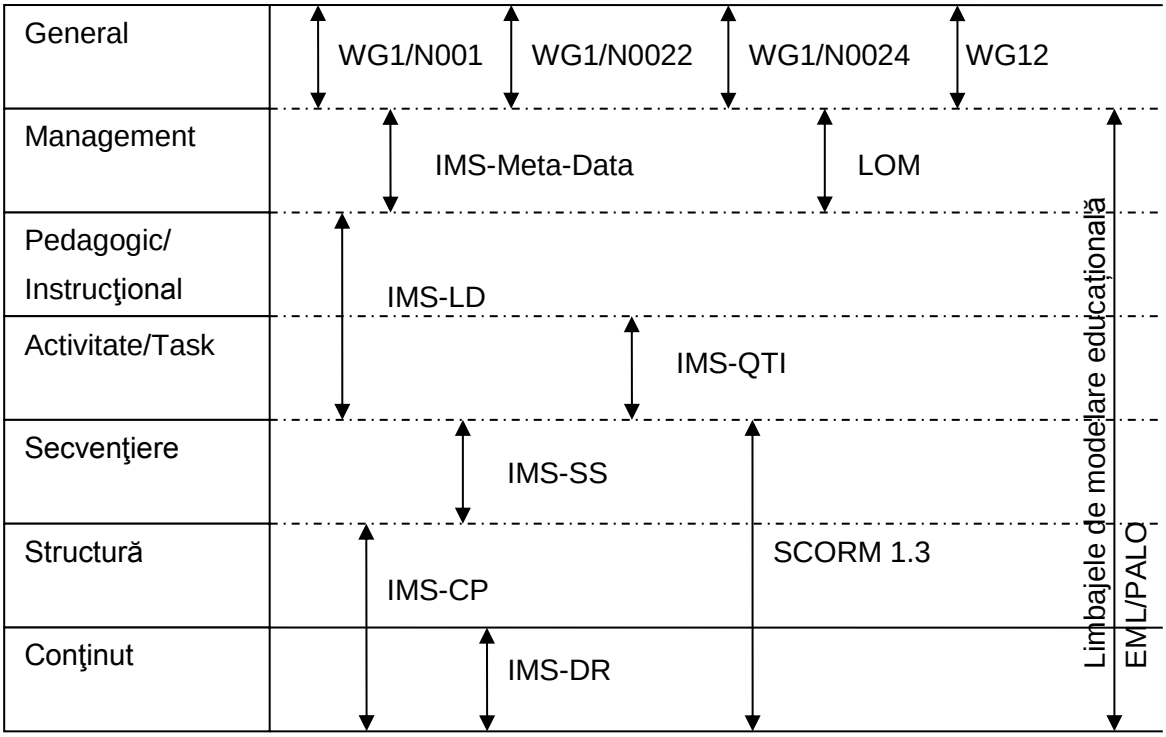


Figura 1-11 Standarde pentru descrierea sistemelor educaţionale la diferite niveluri

1.3.3 LIMBAJE DE MODELARE EDUCAȚIONALĂ

În paralel cu cercetările care vizează elaborarea standardelor pentru sistemele educaţionale, au fost propuse câteva “limbaje de modelare educaţională” (EML), menite să surprindă *aspectele pedagogice considerate într-un sistem de instruire*. Un astfel de limbaj poate reprezenta o nouă paradigmă a procesului de creare a materialului educaţional, deoarece încearcă să abstractizeze conţinutul şi procesele de învăţare dintr-un modul, curs sau de la o disciplină.

Aceste limbaje se pot împărţi în două categorii:

1. Limbaje care modelează doar structura şi conţinutul instrucţional, ignorând existenţa modelelor pedagogice.
    - *CDF* (Course Description Format), elaborat la Swiss Federal Institute of Technology (EPFL). Un CDF este un fişier XML, care poate fi uşor utilizat de către un sistem de management al instruirii, dacă acesta din urmă are acces operaţional la conţinutul pedagogic al cursului şi la fişierul de descriere (CDF).
    - *LMML*, elaborat la University of Passau, Germany (UP). LMML (The Learning Material Markup Language) este bazat pe meta-modelarea arhitecturii pentru managementul cunoaşterii.
    - *TargetTeam*, elaborat la Universität der Bundeswehr, München (UB) TargetTeam asistă pregătirea, utilizarea şi reutilizarea materialelor instrucţionale, managementul conţinutului pentru toate tipurile de situaţii de instruire.

- *TML*, elaborat la University of Bristol, UK (ILRT). TML ("Tutorial Markup Language") permite crearea unor întrebări despre prezentarea materialelor instrucţionale şi evaluarea elevului, prin separarea între conţinutul semantic al unei întrebării şi şablonul de prezentare sau formatare a acesteia.

2. Limbaje care tratează o serie de aspecte pedagogice [98] care trebuie modelate într-un sistem educaţional.
   - *PALO*, elaborat la UNED University, Spain. PALO adoptă o abordare cognitivă, descriind cursurile structurate în module. Fiecare modul include o declaraţie a structurii, activităţile elevului şi ale profesorului, programarea activităţilor şi a conţinutului.
   - *EML*, elaborat la Open University of the Netherlands (OUNL). Limbajul a fost selectat ca modalitate de specificare a IMS-LD (IMS Learning Design), fiind integrat cu IMS-CP (IMS Content Packaging) şi IMS-SS (IMS Simple Sequencing), aşa cum ilustrează și figura 1-11.

1.3.4 METODA MISA ȘI INSTRUMENTE

Cercetătorul Gilbert Paquette [4] propune o *nouă abordare a **proiectării instrucționale*** (engl., "Instructional Design"), numită ***Ingineria Pedagogică*** (engl., "*Pedagogical Engineering*", "*Instructional Engineering*"). Ingineria Pedagogică este o metodă care, pentru un sistem de e-learning, sprijină analiza, proiectarea și elaborarea de materiale pedagogice. Metoda integrează concepte, procese și principii din *proiectarea instrucțională*, *ingineria soft-ware* și *ingineria cognitivă*.

Metoda, numită MISA (**M**éthode d'**I**ngénierie des **S**ystèmes d'**A**pprentissage) propune un model care descrie grafic procesele și principiile aplicate în proiectarea unui sistem de instruire. Modelul conține trei sub-modele: un *model al cunoștințelor*, un *model pedagogic pentru scenariile de învățare* și un *model de difuzare a materialelor pedagogice*. Aplicațiile care implementează sub-modelele sunt **ADISA, MOT+** și **EXPLORA**.

Principala diferenţă între IMS-LD şi MISA: MISA este o metodă de proiectare similară ingineriei software: descrie procesele şi ajută la definirea elementelor de proiectare pentru orice tip de sistem de învăţare. Astfel, o unitate de învăţare IMS-LS ar putea reprezenta o ieşire posibilă, obţinută prin MISA. IMS-LD nu este o metodă de proiectare, ci o specificare standard a produselor rezultate prin aplicarea unei metode de inginerie a instruirii ca MISA.

1.4 ONTOLOGIILE – TEORII CADRU PENTRU DEZVOLTAREA SISTEMELOR INTELIGENTE DE INSTRUIRE

1.4.1 ONTOLOGII

În ultimii ani, interesul cercetătorilor în reprezentarea cunoştinţelor prin tehnici ale inteligenţei artificiale, a migrat de la *forma cunoştinţelor* către *conţinutul acestora*, de la *modurile de reprezentare a cunoştinţelor* către *ontologii*, de la *ingineria cunoştinţelor* către *ingineria*

ontologică[9] (*ca subdomeniu al ingineriei cognitive* [99]. Datorită recunoaşterii importanţei în reprezentarea cunoaşterii în ***sistemele bazate pe cunoştinţe***, ontologiile pătrund şi în sfera cercetărilor privind ***sistemele educaţionale inteligente***. Ontologiile sprijină proiectarea raţională a bazelor de cunoştinţe, conceptualizarea lumii de interes şi definirea strictă a semnificaţiei conceptelor de bază, oferind teorii şi tehnologii care permit acumularea şi partajarea cunoştinţelor. Ontologiile asigură astfel trecerea de la *o societatea informaţională* la o *societate a cunoaşterii*.

Aşa cum se evidenţiază în [100], societatea cunoaşterii - apărută ca rezultat al evoluţiei societăţii informaţionale - vizează îmbunătăţirea tehnicilor de achiziţie, gestiune, partajare şi prelucrare a cunoştinţelor (educaţionale, în cazul nostru) pe web. În acest scop, integrează sinergetic *tehnicile de reprezentare şi prelucrare a cunoştinţelor* cu *sistemele multiagent*, cu *facilităţile web* şi cu *tehnicile de prelucrare a documentelor bazate pe limbajele web-ului semantic.* În acest context, ontologiile şi metadatele joacă un rol esenţial.

1.4.2 DEFINIREA TERMENULUI DE ONTOLOGIE

Există multe interpretări în ceea ce priveşte definiţia ontologiei, în ciuda faptului că este unanim acceptat punctul de vedere conform căruia o ontologie este o *teorie cadru*.

Unii cercetători definesc ontologia din perspectiva filozofică, alţii din perspectiva inteligenţei artificiale, alţii - din cea sistemelor bazate pe cunoştinţe sau din perspectiva agenţilor. Sunt prezentate în continuare câteva dintre definiţiile întâlnite în literatură.

Definiţia 1.1 [101]: *Din perspectiva filozofică*, **O**ntologia[10] reprezintă ***o teorie a existenţei***, care încearcă să explice ”CE este existenţa?”, ”CE proprietăţi comune tuturor existenţelor pot fi evidenţiate?”, ”CE proprietăţi pot explica existenţa?”. Prin introducerea unui sistem de categorii şi de relaţii intrinseci care evaluează o anumită viziune a universului, Ontologia încearcă să răspundă la întrebările ”*CE* există?” şi ”*CUM* este configurată lumea?”.

Definiţia 1.2. [102]: Pornind de la accepţiunea filozofică a termenului, cercetătorii din inteligenţa artificială înţeleg prin ontologie o **descriere explicită şi neambiguă a conceptelor din lumea reală şi a relaţiilor dintre acestea**.

Definiţia 1.3. [103]: *Tot din perspectiva inteligenţei artificiale*, ontologia reprezintă ***o specificare explicită a conceptualizării***. Această definiţie este, în mare măsură, acceptată de cercetătorii în AI (deşi se bazează pe noţiunea de conceptualizare).

Definiţia 1.4. [104]: Din *perspectiva cercetărilor sistemelor bazate pe cunoştinţe*, o ontologie este definită ca ”***o teorie (un sistem) de concepte/vocabular utilizate pentru construirea unor blocuri (module) în sistemele de prelucrare a informaţiei***”. O ontologie este un vocabular şi o specificare a semnificaţiei intenasionale a acestuia.

Definiţia 1.5. [103]:O ontologie este ***o bază particulară de cunoştinţe***, care descrie faptele considerate întotdeauna adevărate pentru o comunitate de utilizatori: (1) în virtutea unor

[9] Perioada de după 1990 (demarată prin proiectul american *Knowledge Sharing Effort*, DARPA)

[10] **O**ntologia, cu O mare (Guarino, 1995), pentru a deosebi *termenul filozofic* de cel folosit în *inteligenţa artificială*

acorduri bazate pe semnificaţia vocabularului utilizat (cunoştinţele analitice); (2) a căror valoare de adevăr nu provine din semnificaţia vocabularului utilizat (cunoştinţe neanalitice, comune).

Definiţia 1.6. [105]:O ontologie este ***un consens asupra lumii de interes pentru o propunere specifică***. Ideea principală este aceea de a defini o ontologie cât mai vag posibil, fără a pierde, însă, esenţialul acesteia.

Definiţia 2.7. [102]: *Din perspectiva agenţilor*, **Gruber** *defineşte* ontologiile ca ***acorduri în ceea ce priveşte conceptualizările partajate***. Conceptualizările partajate includ cadre de lucru pentru modelarea domeniului cunoştinţelor, protocoalele specifice conţinutului pentru comunicarea între agenţi, angajamentele despre reprezentarea teoriilor particulare domeniului. În contextul partajării cunoştinţelor, ontologiile sunt specificate în forma definiţiilor vocabularului de reprezentare. Un caz foarte simplu poate fi cel al unei ierarhii de tipuri, care specifică clase şi relaţiile de subsumare dintre acestea. Schemele bazelor de date relaţionale servesc, de asemenea, ca ontologii, prin specificarea relaţiilor care pot exista într-o bază de date partajată şi a constrângerilor legate de integritatea acesteia. O ontologie este ***o descriere a conceptelor care pot exista pentru un agent sau pentru o comunitate de agenţi***.

Definiţia 1.8. [106]: Aproximativ acelaşi sens cu definiţia 1.7. îl are şi definiţia propusă de Wielinga şi Schreiber: O (AI-)ontologie se constituie într-o ***teorie despre entităţile care pot exista pentru un agent***.

Definiţia 1.9 [107]: O definiţie compusă este următoarea: O ontologie constă în ***conceptele cu definiţiile lor, organizate ierarhic, relaţiile dintre acestea*** (cele mai multe, de tipul IS-A şi PART-OF) ***şi axiomele care formalizează definiţiile şi relaţiile***.

Definiţia 1.10 [105]: O ontologie reprezintă ***specificarea conceptelor care vor fi utilizate pentru exprimarea cunoştinţelor: tipuri de entităţi, atribute şi proprietăţi, relaţii şi funcţii, constrângeri.***

Definiţia 1.11. [108]: o ontologie pentru un domeniu sau task particular descrie o ***taxonomie de concepte ale task-ului sau ale domeniului şi defineşte interpretarea semantică a cunoaşterii.***

Definiţia 1.12. [109]: O ontologie este o ***reprezentare explicită şi parţială a conceptualizării***. O conceptualizare este reprezentată printr-o mulţime de modele ale unui limbaj logic, L, care descrie interpretările intensionale (admisibile) ale simbolurilor nelogice (vocabularul).

Definiţia 1.13 [104]: O ontologie este o ***specificare a unui vocabular al simbolurilor nelogice*** şi include tipuri de entităţi, atribute şi proprietăţi, relaţii, funcţii şi constrângeri.

Definiţia 1.14 [110]: O ontologie este o ***teorie logică, o axiomatizare (posibil incompletă) care impune o serie de constrângeri asupra modelelor intensionale ale unui limbaj logic.***

Definiţia 1.15. [111]: O ontologie este o ***specificare explicită, parţială a conceptualizării, exprimând un punct de vedere (de meta-nivel) asupra unei mulţimi de teorii***

posibile ale unui domeniu, cu scopul de proiectare modularizată, reproiectare şi reutilizare a componentelor sistemului bazat pe cunoştinţe.

Definiţia 1.16. [107]: O ontologie este o specificare ***formală***[11] şi ***explicită***[12] a unei ***conceptualizări***[13] ***partajate***[14] (comune).

Aşa cum se observă din definiţiile prezentate, termenul de ontologie este un termen relativ nou, obţinut însă din conceptele deja existente (taxonomie, vocabular comun, model superior), prin adăugarea formalizării, a relaţiilor îmbogăţite şi a reprezentării explicite a cunoaşterii.

Fiecare dintre definiţiile anterioare surprinde câte un aspect al ontologiilor şi furnizează puncte de vedere diferite ─ dar complementare ─ asupra aceleeaşi realităţi. Ontologiile ***nu*** se disting neapărat prin *formă*, ci prin ***rolul*** lor [112], importanţa ontologiilor fiind recunoscută în cercetările din domenii ca ingineria cunoştinţelor, reprezentarea cunoştinţelor, integrarea informaţiilor, managementul cunoştinţelor, stocarea şi extragerea informaţiilor.

1.4.3 CLASIFICĂRI ALE ONTOLOGIILOR

Ontologiile pot fi clasificate în funcţie de diferite criterii, cum ar fi: subiectul conceptualizării, structura conceptualizării, complexitatea conceptualizării [112], granularitatea conceptelor, gradul de formalizare sau reutilizarea cunoştinţelor. În continuare sunt prezentate cele mai importante modalităţi de clasificare a ontologiilor - întâlnite în literatura de specialitate - şi o sinteză a acestora, ilustrată prin figura 1-12.

A. *Din perspectiva subiectului conceptualizării*, putem evidenţia clasificările propuse de către Van Heijst şi Chandrasekaran.

A1. În funcție de *subiectului conceptualizării*, *van Heist* [108] clasifică ontologiile în:

- Ontologii *superioare, de nivel înalt* (*Upper* sau *Top-level Ontology*) - Aceste ontologii sunt independente de o problemă sau de un domeniu particular [113], putând fi folosite de o comunitate largă de utilizatori. Ontologiile de acest tip descriu concepte generale, abstracte, cum ar fi: entităţi, stări, timp, spaţiu, evenimente, procese, relaţii, proprietăţi. Dezvoltarea unei astfel de ontologii are o importanţă deosebită, însă, până în momentul de faţă, există doar câteva încercări: *ontologia top-level* a lui Guarino, ontologia *CYC* (divizată în micro-teorii), *Mikrokosmos*, *laticea booleană a lui Sowa.*

- Ontologii *generice* - Numite şi *meta-ontologii* sau *ontologii nucleu* (engl., "*core ontology*"), aceste ontologii conţin cunoştinţe generice, mai "puţin abstracte" decât cunoştinţele descrise prin ontologiile superioare, dar "îndeajuns de generale" pentru a putea fi reutilizate în domenii foarte diferite. Exemple de acest tip sunt ontologia *topologică mereologică* (*Mereology Ontology*) şi *ontologia topologică* (*Topology Ontology*).

- Ontologii *ale domeniului* - Sunt specifice unui anumit domeniu şi pot fi reutilzate doar în cadrul domeniului respectiv. Furnizează vocabularul despre conceptele unui domeniu generic (medicină, electronică, etc.) şi relaţiile dintre acestea. Pot fi evidenţiate ontologiile

[11] Ontologia trebuie să poată fi tradusă într-un limbaj operaţional

[12] Tipurile de concepte, relaţiile dintre acestea şi constrângerile sunt declarative

[13] Model abstract al unui fenomen din lumea reală

[14] Ontologia capturează cunoştinţele consensuale ale unui grup sau ale unei comunităţi

EngMath (ontologie Ontolingua pentru modelarea matematică în inginerie), *PhysSys* (ontologie inginerească pentru modelarea, simularea şi proiectarea sistemelor fizice) şi *Ontologia Întreprinderii*. Cercetătorul Mizoguchi [114] consideră că ontologia domeniului caracterizează cunoştinţele domeniului sau problema (task-ul) vizată.

- Ontologii *ale aplicaţiei* - Ontologiile de aplicaţie reprezintă specializări ale ontologiilor generice şi ale domeniului, descriind conceptele specifice unei aplicaţii. Conceptele din aceste ontologii corespund rolurilor jucate de entităţile domeniului în timpul îndeplinirii unei anumite activităţi.

- Ontologii de reprezentare a cunoştinţelor [115] - Sunt ontologii care descriu o clasificare a primitivelor utilizate de un limbaj de reprezentare (concepte, relaţii, atribute); specifică formalisme de reprezentare a cunoştinţelor. Un exemplu din această categorie este *ontologia cadrelor (Frame Ontology*), care integrează primitive de reprezentare a limbajelor bazate pe cadre, cum sunt cele folosite în Ontolingua: clase, instanţe, faţete, proprietăţi (sloturi), relaţii, restricţii, valori permise, etc.

A2. Tot din perspectiva subiectului conceptualizării, Chandrasekaran [116] evidenţiază:

- Ontologia *metodelor de rezolvare a problemelor* - Furnizează definiţiile conceptelor şi relaţiilor folosite în specificarea procesului de raţionament utilizat în realizarea unui anumit task (abducţia în diagnoză).

- Ontologia *task-urilor* - Descrie vocabularul specific unei activităţi generice (planificare, proiectare, diagnoză, etc.) [117].

A3. În raport cu **nivelul de detaliere al conceptualizării ontologiei** (impus de obiectivul operaţional propus pentru ontologie), pot fi identificate următoarele categorii de ontologii:

- Ontologii *cu granularitate redusă, fină* - Sunt ontologii foarte detaliate, care folosesc un vocabular bogat, capabil să asigure o descriere detaliată a conceptelor pertinente ale domeniului [118] sau ale task-ului.

- Ontologii cu granularitate mare, largă - Sunt ontologii care folosesc un vocabular mai puţin detaliat, folosit în scenarii de utilizare specifice sau de către utilizatori care sunt de acord cu o conceptualizare sub-adiacentă.

B. În ceea ce priveşte ***nivelul de reprezentare a cunoştinţelor***[15], se evidenţiază clasificările propuse de către Mizoguchi şi Bachimont.

B1. Mizoguchi [119] [120] [121] propune o ***clasificare bazată pe procesele din ingineria ontologică***, pe trei niveluri[16]:

- *Nivelul 1* (sau *nivelul conceptual*), în care modelarea este adesea informală, realizându-se specificaţii în limbaj natural şi definirea minimală a conceptelor. Exemple tipice sunt

[15] Termenul "nivel de reprezentare a cunoştinţelor" (propus de Newell, în 1982) are interpretări diferite în viziunea cercetătorilor. Termenul a fost propus cu scopul de a diferenţia noţiunile "reprezentări" şi "cunoştinţe".

[16] În *unele lucrări*, procesele de la cele trei niveluri sunt numite *conceptualizare*, *ontologizare* şi *operaţionalizare*.

ierarhiile de subiecte găsite prin motoarele de căutare şi etichetele (tag-urile) folosite în descrierea meta-datelor.

- *Nivelul 2* (sau *nivelul formal*), este formalizat într-un limbaj de reprezentare care să poată fi interpretat de către un sistem de calcul. Sunt adăugate definiţiile formale, relaţiile şi constrângerile sunt exprimate prin axiome, iar definiţiile sunt declarative şi formale.

- *Nivelul 3* (sau *nivelul operaţional*) este implementat într-un limbaj de programare prin care specificaţia devine executabilă. Ontologia este executabilă în sensul că modelele construite plecând de la ontologie funcţionează folosind modulele furnizate printr-unul din codurile abstracte asociate conceptelor ontologiei. Exemple tipice pentru ontologiile de nivel 3 sunt *ontologiile task-urilor*, prezentate în secțiunea 1.4.4.

B2. Bachimont [122] propune o *tipologie axată pe semnificaţie*:

- *Nivelul semantic* (sau *interpretativ*), în care toate conceptele care se definesc pentru universul de discurs trebuie să îndeplinească patru principii diferenţiale. Aceste principii corespund *angajamentului semantic*, garantând astfel că eticheta fiecărui concept are un singur sens.

- *Nivelul formal* (sau *referenţial)* caracterizează o *ontologie referenţială*, în care conceptele referenţiale (sau formale) se caracterizează prin etichetele a căror semantică este definită prin extensiune.

- *Nivelul operaţional* (sau *computaţional*) codifică specificaţiile într-un limbaj de programare, obţinându-se astfel o specificaţie executabilă.

Figura 1-12 prezintă o sinteză a principalelor tipuri de ontologii, conform criteriilor evidenţiate în această secțiune.

C. Din punct de vedere al ***nivelului de formalizare a reprezentării cunoştinţelor***, Gruber [103] propune o clasificare în:

- Ontologii *informale*, descrise în limbaj natural, cu semantică deschisă;

- Ontologii *semi-informale*, descrise într-un limbaj natural structural şi limitat;

- Ontologii *semi-formale*, descrise într-un limbaj artificial definit formal;

- Ontologii *formale*, descrise într-un limbaj artificial cu semantică formală, teoreme şi demonstraţii ale proprietăţilor.

Având la bază tipurile de ontologii întâlnite în literatura de specialitate, un ***punct de vedere personal asupra unei posibile clasificări*** este ilustrat în figura 1-13.

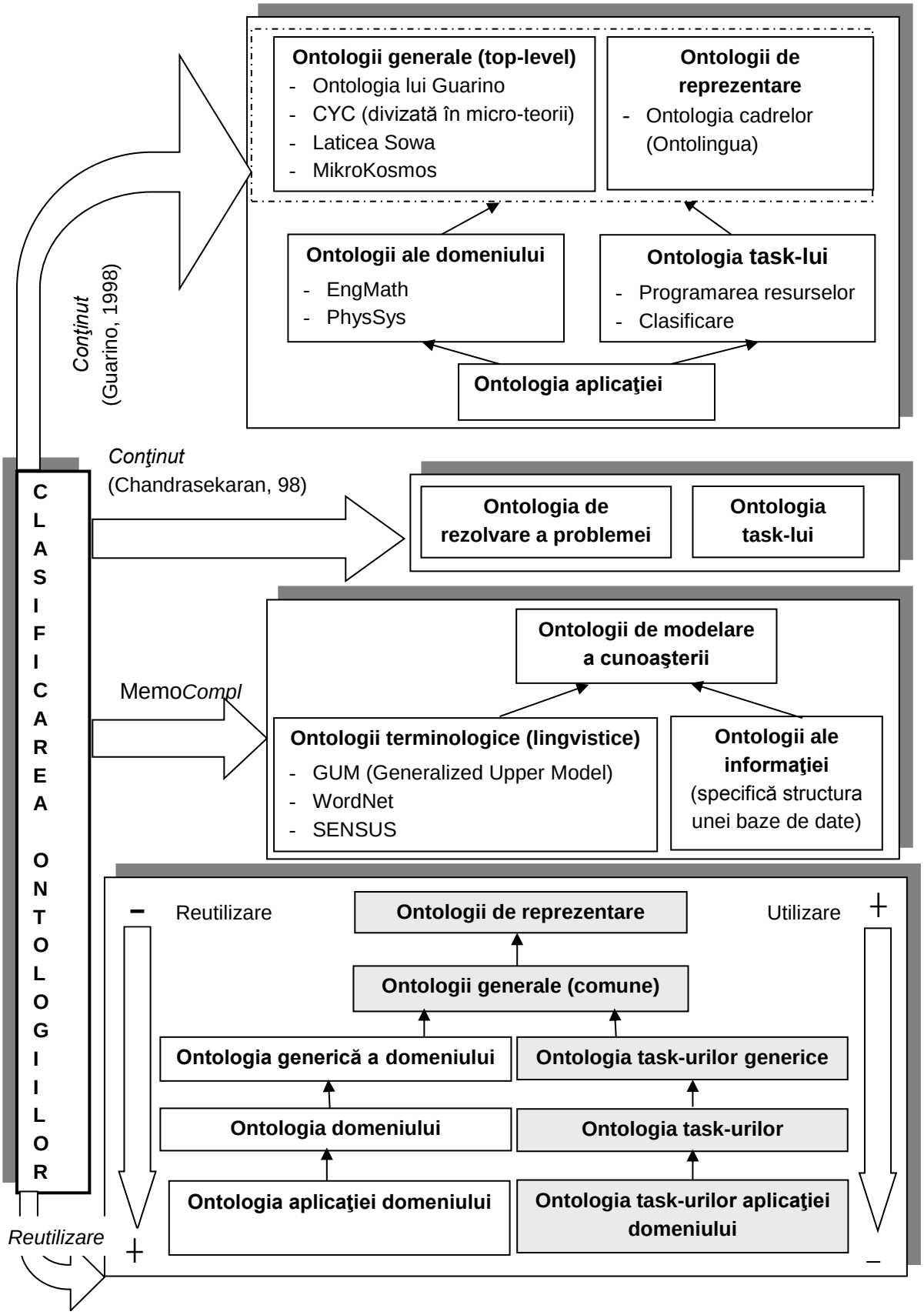


Figura 1-12 Clasificarea ontologiilor în funcţie de diferite criterii

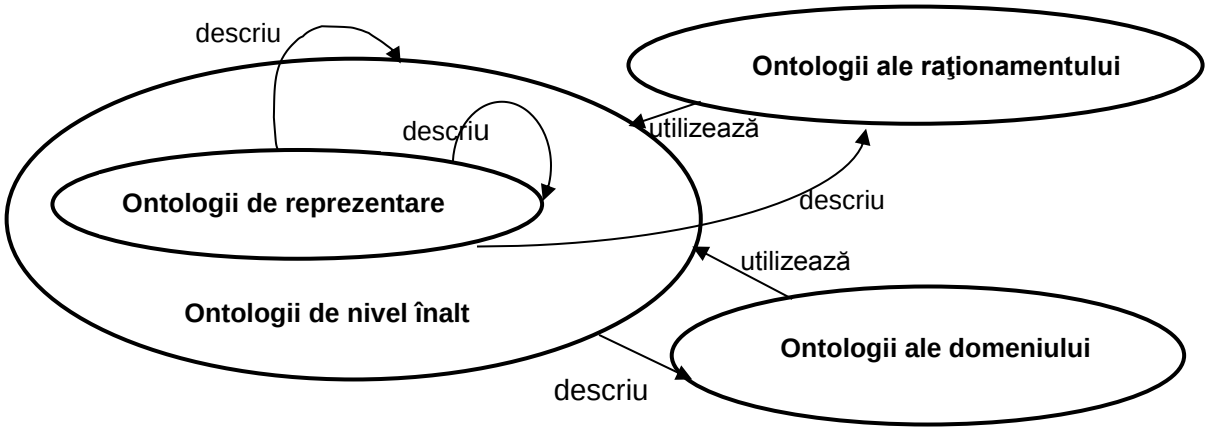


Figura 1-13 Diferite tipuri de ontologii

Aşa cum vor demonstra şi secţiunile şi capitolele următoare, potenţialul de sistematizare şi explicitare a cunoaşterii pe care îl oferă ontologiile se poate concretiza în diferite moduri în sistemele bazate pe cunoştinţe, deci şi în sistemele educaţionale inteligente.

1.4.4 ONTOLOGIA TASK-URILOR ŞI A METODELOR

Efortul cercetărilor privind reprezentarea şi reutilizarea cunoaşterii din sistemele inteligente s-au canalizat pe două direcţii: *abordarea conceptuală* şi *abordarea funcţională*.

- În *abordarea conceptuală, orientată către formă (întâlnită în primele generaţii de sisteme expert, bazate pe reguli),* cunoştinţele erau reprezentate independent de problema care trebuie rezolvată (task), neexistând o teorie a task-urilor care să ajute la identificarea tipurilor de cunoştinţe necesare; singurii termeni disponibili erau foarte generali şi independenţi de task (predicate, mulţimi, relaţii, etc.).
- Trecerea la următoarea etapă este iniţiată de cercetările privind construirea sistemelor bazate pe cunoştinţe; acestea promovează ideea existenţei unei conexiuni închise între (*task-urile*) *sarcinile* pe care sistemul trebuie să le îndeplinească, *metodele* alese şi *vocabularul* folosit pentru reprezentarea cunoştinţelor modelate. Apare astfel *abordarea funcţională (cu origini în ingineria software)*, în care sistemele bazate pe cunoştinţe sunt considerate *sisteme orientate către task*. Principalul susţinător al acestei abordări este Chandrasekaran, care lansează ideea că în rezolvarea unor probleme există cunoştinţe de control cu structuri similare, care pot fi aplicate în diferite domenii. Astfel, accentul cade asupra task-urilor şi a rolului diferitelor tipuri de cunoştinţe jucat în realizarea lor, furnizând un vocabular al termenilor (relativ la task), şi ***implicit***, al cunoştinţelor legate de task şi al modului în care acestea pot fi utilizate.

Cu cât va fi dezvoltată o înţelegere a mai multor task-uri, vor fi identificaţi - pentru reprezentarea cunoştinţelor - mai mulţi termeni ai nivelurilor cunoştinţelor.

Astfel, a*vantajele* abordării orientate către task au fost evidenţiate în numeroase lucrări:

- Bylander şi Chandrasekaran consideră că procesul de achiziţie a cunoştinţelor poate fi uşurat prin furnizarea unui vocabular - în termenii cunoştinţelor necesare unui anumit task - şi ghidarea în organizarea şi utilizarea cunoştinţelor achiziţionate;

- Chandrasekaran, Tanner şi Josephson văd în abordarea orientată către task o importantă pârghie în generarea explicaţiilor rezolvării problemelor;
- Anumite cunoştinţe şi structuri de control sunt comune unui anumit task (ca proiectarea sau diagnoza), iar acest task poate fi aplicat în diferite domenii.

Termenul de **task** este folosit pentru a desemna o instanţă a unei probleme, o clasă de probleme sau o clasă de probleme însoţită de o descriere abstractă a unei metode de rezolvare a problemei [123]. Astfel, Newell şi Simon înţeleg prin task, fie o instanţă a unei probleme, fie o clasă de probleme; Clancey desemnează prin task unul dintre sub-scopurile principale stabilite de către un sistem, task-ul incluzând un nivel înalt de descriere a metodei; Wielinga înţelege prin task secvenţa operaţiilor (la un nivel de abstractizare potrivit) pe care le realizează un anumit sistem; alţi cercetători folosesc noţiunea de task pentru un tip de problemă (diagnoza este un task, un tip de problemă în care scopul este identificarea cauzelor unei mulţimi de comportamente anormale ale unui sistem) [124].

Fie că înţelegem prin task reprezentarea unei clase de probleme, fie un element de control în sistemele bazate pe cunoştinţe, fie generalizarea unei funcţionalităţi, toate aceste accepţiuni converg către noţiunea de **mediu de rezolvare a problemelor**.

Un task poate avea diferite *niveluri de generalitate* (diagnoză, diagnoză medicală, diagnoza elevului într-un sistem de instruire, etc.) [125]. Tratarea diagnozei ca tip de problemă (ca task) permite studierea strategiilor generale (a modalităţilor de rezolvare) pentru această clasă de probleme.

Indiferent de terminologia utilizată (*task* sau *metodă de rezolvare a problemei*), scopul cercetărilor în această abordare este acela de a reprezenta (şi descrie) activităţi *independente* de un anumit domeniu şi modalităţile de realizare a acestora. Ca urmare, *dezvoltarea* bazelor de cunoştinţe se va realiza pe baza metodologiilor propuse de "*ingineria cunoştinţelor*[17]" (tehnologie pentru elicitarea cunoştinţelor, organizarea expertizei într-o structură operaţională şi construirea bazei de cunoştinţe).

În plus, se conturează mai bine conceptul de ***ontologie*** - privită ca teorie a vocabularului/conceptelor utilizate ca blocuri de construcţie sau ca blocuri reutilizabile în rezolvarea unor probleme – şi rolul acestora în procesul de achiziţie a cunoştinţelor.

Principalele abordări în crearea unei ontologii a task-urilor sunt:

- **Task-urile generice** [116]: Chandrasekaran formulează noţiunea de *task generic*, care conţine *atât sarcina propriu-zisă*, cât şi *metoda necesară*. Pentru creşterea flexibilităţii în specificarea metodei, se ajunge la *TSA* (*Task-Specific Architecture*). Chandrasekaran dezvoltă noţiunea de *structură a task-urilor*, pe baza căreia se identifică un număr de

[17] Ingineria cunoştinţelor = disciplină inginerească constând în procesul de elicitare, structurare, formalizare şi operaţionalizare a cunoaşterii dintr-un domeniu, cu scopul construirii unui produs software care realizează o anumită sarcină (task). Cunoaşterea reprezentată:

- Cunoştinţe despre task (orientată către și reprezentată prin descompunere funcţională);
- Cunoştinţe despre domeniu;
- Cunoştinţe despre raţionament (în principiu, paşi inferenţiali în cunoaşterea domeniului aplicaţiei în cadrul task-ului vizat).

metode alternative (moduri de îndeplinire a task-urilor) pentru un task; fiecare metodă fixează, la rândul ei, sub-task-uri, această analiză *task-metodă-subtask* mergând până la nivelul de detaliere în care task-urile sunt primitive ale cunoaşterii în baza de cunoştinţe.

- **Modelul specific situaţiei** [126]: Clancey a dezvoltat perspectiva de construire a modelului în care sistemele bazate pe cunoştinţe folosesc *modele ale domeniului* şi *reguli de inferenţă* pentru a construi un ***model specific situaţiei***. Acest model este un *graf* ale cărui noduri şi legături sunt construite prin paşii inferenţiali ai rezolvării problemei. Un model specific situaţiei este un subgraf al grafului de rezolvare a problemei. De remarcat în această abordare este conexiunea închisă între task-uri, metode şi termenii din modelul specific situaţiei, deoarece aceşti termeni sunt reflectaţi în cunoştinţele necesare implementării metodelor.

- **Cadrul componentelor** [127]: Cadrul componentelor de expertiză porneşte tot de la ideea că nivelul de descompunere a task-ului - cu ajutorul metodelor - variază în funcţie de scopurile analizei. Cadrul componentelor este descris din trei perspective: *perspectiva modelului domeniului* (descrie cunoştinţele utilizate de către task-uri: obiectele domeniului, proprietăţile şi relaţiile dintre ele, numite "modele ale cazului"), *perspectiva task-ului* (structura activităţilor pe care le realizează SBC-ul, numită şi structura task-urilor, specificarea ordinii de realizare) şi *perspectiva metodei* (*metode de descompunere* care definesc ordinea execuţiei subtask-urilor şi *metode soluţii,* care reprezintă operaţii executabile).

- **Metode de limitare a rolului** [128] identifică o mulţime de metode de rezolvare a problemelor, metode aplicabile unei categorii de task-uri. Fiecare dintre aceste metode are o *descriere abstractă* a modului de rezolvare a unui tip de problemă (task). Cu ajutorul *rolurilor* se specifică diferitele tipuri de cunoştinţe pe care le utilizează metoda.

- **Metodologia KADS**: Metodologia KADS [129] - proiect de cercetare european pentru modelarea cunoştinţelor - unifică teoriile anterioare, oferind un cadru coerent de modelare a proceselor de raţionament pentru sistemele bazate pe cunoştinţe.

Analiza task-urilor şi a metodelor poate furniza un repertoriu bogat de task-uri şi metode, la diferite niveluri, descrierea acestora reprezentând surse ale ontologiilor de rezolvare a problemelor. *Ontologia task-urilor* (şi/sau a metodelor de rezolvare a problemei [130], care furnizează un model conceptual al raţionamentului şi permite reprezentarea cunoaşterii inferenţiale într-un sistem bazat pe cunoştinţe, împreună cu *ontologia domeniului* în care este realizat task-ul, pot asigura nu numai dezvoltarea bazelor de cunoştinţe, dar şi reutilizarea şi partajarea cunoaşterii existente.

În scopul dezvoltării bazelor de cunoştinţe au fost imaginate două scenarii:
1) Reutilizarea bazelor de cunoştinţe existente pentru rezolvarea unei clase de probleme sau pentru realizarea unor task-uri diferite în acelaşi domeniu;
2) Utilizarea competenţelor altor sisteme în procesul de raţionament.

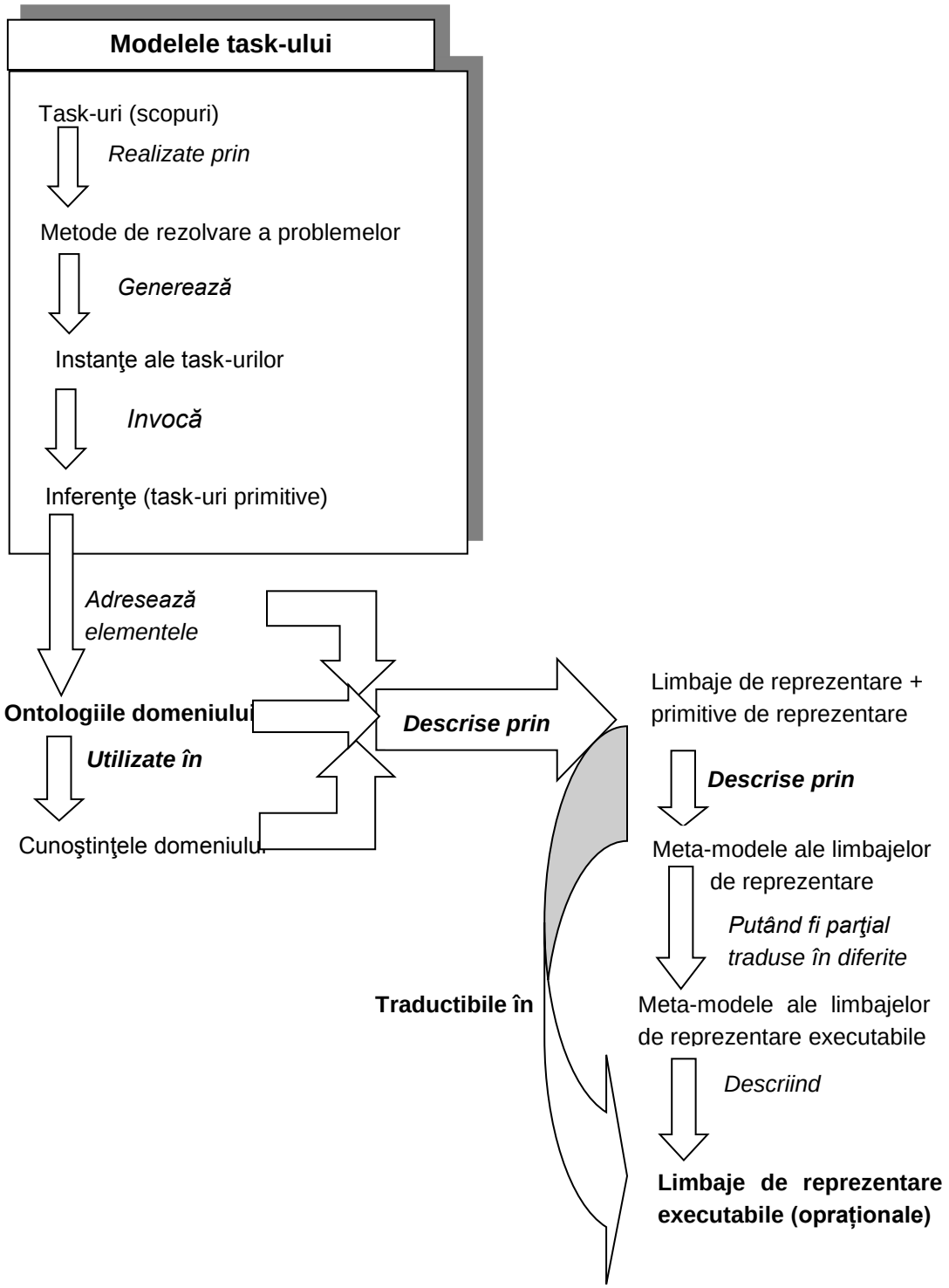


Figura 1-14 Legătura între principalele elemente utilizate în construirea unui model conceptual și operaționalizarea acestuia

1.4.5 ROLUL ONTOLOGIILOR ÎN SISTEMELE EDUCAŢIONALE

Cercetările orientate către conţinut reprezintă o tentativă de partajare a cunoştinţelor între agenţii umani şi/sau artificiali. Obiectivul principal al acestor cercetări constă în furnizarea unor tehnici şi teorii care să permită acumularea şi interoperabilitatea cunoştinţelor (pe web). Pe lângă rolul deosebit de important pe care aceste cercetări îl au în *generaţia următoare a sistemelor bazate pe cunoştinţe (deci şi în sistemele educaționale inteligente şi în sistemele autor)*, ele vor permite utilizatorilor (profesori, pedagogi, proiectanţi ai sistemelor educaţionale, elevi) accesul la materialele educaţionale relevante aflate în Internet.

Aşa cum s-a evidenţiat în secțiunile 1.4.3 şi 1.4.4, ontologiile variază în funcţie de:
- Gradul formalismului de reprezentare (de la informal către formal);
- Obiectivul operaţional vizat (comunicare între agenţi, interoperabilitate între sisteme, aplicarea unei probleme de inginerie cum ar fi reutilizarea componentelor sau rezolvarea de probleme);
- Subiectul conceptualizării (domeniul cunoaşterii, cunoştinţe de raţionament, cunoştinţe legate de modelul de reprezentare).

Tot din secțiunile anterioare şi din figura 1-12 (care ilustrează clasificarea ontologiilor după gradul de reutilizare) se pot distinge două scenarii diferite de utilizare a ontologiilor:
1. Primul scenariu, în care există o mulţime de ontologii reutilizabile, organizate într-o bibliotecă de ontologii care conţine ontologii de domeniu şi ontologii de task-uri. Chiar dacă numărul cunoştinţelor ontologice este redus, ceea ce este important este calitatea acestora (natura distincţiilor), care pot sprijini proiectarea în analiza conceptuală.
2. Cel de-al doilea scenariu, în care există o ontologie generală care va consta din distincţii brute la nivelul domeniului (între entităţile de bază ale domeniului universului) şi distincţii meta nivel (relative la tipurile de clase şi de relaţii).

Vom analiza în acest capitol rolul particular pe care o ontologie (**explicită**) îl poate juca în sistemele autor şi/sau în sistemele educaţionale, accentuând ***perspectiva arhitecturală*** (asupra căreia vom reveni), în care ontologia ghidează aspectele şi componentele unui sistem; vorbim astfel, despre ***sistemele orientate pe ontologie***.

În contextul educaţiei şi al web-ului semantic (care s-a impus deja ca tehnologie promiţătoare în implementarea e-learning-ului), ontologiile [131] pot fi aplicate într-o varietate de probleme complexe, ca achiziţia cunoştinţelor [132], reprezentarea cunoaşterii în sistemele educaţionale inteligente şi în sistemele autor, partajarea şi reutilizarea cunoaşterii [133], adnotarea şi căutarea obiectelor instrucţionale[18] [134], personalizarea conţinutului educaţional [135].

În cadrul acestei lucrări ontologiile vor fi utilizate pentru **rezolvarea unor probleme de achiziţie, reprezentare şi modelare a cunoaşterii (pedagogice)** în sistemele educaţionale, motivată fiind de afirmaţia lui Tom Murray [136], [137]: *”proiectanţii trebuie să reprezinte cunoştinţele pedagogice într-o manieră explicită şi modularizată, ceea ce constituie o problemă dificilă”*.

[18] LO – obiect instrucțional (engl, “learning object”)

Sunt prezentate în continuare câteva exemple de ontologii, scopul utilizării acestora şi sistemele care le operaţionalizează, ***insistând asupra celor care vin în sprijinul proiectării pedagogice***. Vom evidenţia astfel modul în care utilizarea ontologiilor a condus la obţinerea inteligenţei în sistemele educaţionale sau în sistemele autor.

1. Ontologia task-urilor educaţionale şi ontologii de proiectare a instruirii

Cercetări intense privind dezvoltarea unor ontologii pe care să se bazeze sistemele autor se desfăşoară în **Japonia**, în colectivul condus de Riichiro Miyoguchi. Aceasta este prima echipă de cercetare care a avansat o serie de teorii care susţin necesitatea elaborării unor ontologii de proiectare a instruirii, a anticipat modurile în care ingineria ontologică va ajuta la depăşirea problemelor din AIED [138] şi a lansat proiecte ambiţioase, de lungă durată, însă dificil de realizat.

Ideea de la care au pornit teoriile şi proiectele cercetătorilor japonezi este aceea că proiectanţii unui sistem educaţional inteligent de instruire trebuie să cunoască teoriile de instruire şi modelele care pot furniza principiile şi strategiile adecvate, componentele funcţionale necesare pentru atingerea scopurilor şi obiectivelor instruirii, acţiunile pedagogice adecvate fiecărei situaţii, arhitecturile şi strategiile care susţin acţiunile şi modul de structurare a domeniului cunoştinţelor, astfel încât sistemul obţinut să poată asigura un proces de învăţare coerent. În cele mai multe sisteme de instruire aceste caracteristici fundamentale sunt implicite şi neformalizate.

O soluţie a acestor probleme o constituie ***ingineria ontologică a instruirii*** [139], [140], [141], care poate ajuta specificarea la cel mai înalt nivel a funcţionalităţii unui sistem inteligent de instruire (IIS), constituind punctul de legătură dintre cunoştinţele umane şi cele din bazele de cunoştinţe. O ontologie a instruirii poate furniza suportul necesar construirii unui mediu de dezvoltare autor pentru IISs: la cererea autorului, un agent îi poate prezenta teoriile relevante, furnizându-i, din punct de vedere teoretic, justificările pentru strategiile de predare şi învăţare. Un server Web al cunoştinţelor instruirii poate asigura funcţiile prezentate [142]. În acest scop, primul pas îl constituie *extragerea unei ontologii din teoriile instruirii şi din modelele existente de proiectare a instruirii.* Datorită complexităţii sistemelor (inteligente) de instruire, a multitudinii şi diversităţii abordărilor şi teoriilor pe care se bazează proiectarea şi utilizarea acestora, mediile de dezvoltare autor trebuie să se bazeze pe **ontologii de proiectare a instruirii**, care să constituie teorii cadru (**ontologii de bază**) [143], [144], [145] (figura 1-14).

Pe lângă avantajele oferite de ontologii, un sistem autor bazat pe ontologia proiectării instruirii, trebuie să prezinte următoarele capabilităţi [146]:

- Încă din faza de specificaţii a viitorului sistem de instruire, autorul poate specifica tipul sistemului (ITS, IIS, ILE, AHS sau OLE) – care implică cerinţe de proiectare diferite, pentru a obţine un produs coerent şi congruent. De exemplu, cerinţele legate de proiectarea unui ILE pot fi: sistem individual sau sistem cu învăţare bazată pe o anumită temă; un astfel de sistem necesită o proiectare constructivistă: explicitarea principiilor de proiectare, luarea deciziilor autorului, finalitatea pedagogică. Cerinţele pentru proiectarea unui OLE pot legate de reprezentarea unor evenimentelor de învăţare externe: sistemul trebuie să raţioneze, să genereze ipoteze, să ia decizii despre evenimentele interne şi externe, adaptându-şi strategiilor de instruire, pe baza culturii şi a afectivităţii elevului.
- Pot fi alese condiţiile în care sistemul va fi utilizat: sistem de instruire complementar, suplimentar sau înlocuitor al predării [147].

- Poate oferi posibilitatea punerii în practică a unor teorii pedagogice diferite (de exemplu, abordarea lui Reigeluth, care consideră elevul un coproiectant al instruirii; ca urmare, alegerea strategiei de instruire poate fi realizată de către elev sau de către sistem) [148].

Câteva dintre principalele proiectele ale cercetătorilor japonezi sunt prezentate în continuare:

Unul dintre primele proiecte ale cercetătorilor japonezi a fost demarat în 1997: dezvoltarea unui **cadru general**, **FITS**, pentru modelarea conceptuală a tutorialelor inteligente (FITS – general **F**ramework for **ITS**). **Ontologia task-urilor educaţionale**, care cuprinde concepte ca *Scopurile educaţiei*, *Starea elevului*, *Funcţionalitatea sistemului*, *Interacţiunea elev-sistem* şi *Cunoştinţele materialului de predare*, poate fi specializată în ontologii ale task-urilor domeniului de instruire. Ontologia, (parţial implementată, în Ontolingua, CLEPE) ajută la formalizarea procesului de construire a unui sistem educaţional, furnizează primitive care facilitează descrierea cunoaşterii la nivelul conceptual, ajută la construirea modelului explicit, iar axiomele ghidează autorul în construirea sistemului.

Poate fi utilizată în diferite contexte, ca de exemplu: susţine *elaborarea specificaţiilor* pentru sistemele inteligente de instruire sau furnizează un *cadru de generare a tutorialelor adaptive* în timp real, prin utilizarea unor resurse distribuite (sprijinind luarea unor decizii tactice în vederea asamblării şi adaptării unor componente existente).

În sistemul autor **SmartTrainer**, un sistem multi-agent [149], care antrenează elevii în remedierea accidentelor produse la staţiile de putere electrică, rolul ontologiei este acela de a susţine comunicarea dintre agenţi. Conceptele, relaţiile şi axiomele ontologiei furnizează vocabularul necesar formulării de mesaje şi interogări (de exemplu, nerespectarea constrângerilor de ordin semantic determină generarea mesajelor corespunzătoare), dar şi controlul şi mentenanţa instanţelor ontologice.

Inaba [150] propune **ontologie a domeniului învăţării colaborative**, care reprezintă conceptele comune unor teorii de învăţare (învăţarea prin observare, teoria flexibilităţii cognitive, teoria socio-culturală, etc.), care să furnizeze funcţionalităţi de mediere a învăţării într-un sistem cu învăţare colaborativă. Selectând o anumită teorie şi formând un grup de învăţare, se poate ajunge la o învăţare colaborativă eficientă.

Tot cercetătorii japonezi au dezvoltat o **ontologie a învăţării, instruirii şi proiectării instruirii (L/I/ID)**, numită OMNIBUS. Ontologia formalizează procesul de învăţare/instruire descriind cum evenimentele de nivel superior (numite evenimente educaţionale) pot fi descompuse în evenimente I_L[19]. O unitate de instruire/învăţare este descrisă ca un eveniment I_L şi definită ca o combinaţie între o acţiune instrucţională şi o modificare a stării elevului (de exemplu, acţiunea instrucţională *"pregătirea elevului pentru instruire"* este asociată cu schimbarea de stare *"gata de instruire"*). Ontologia sprijină proiectantul instruirii şi furnizează indicaţii şi recomandări pentru o anumită teorie de învăţare.

Pe baza acestei ontologii au fost dezvoltate **două sisteme de proiectare instrucţională**, conştiente de ontologie: **SMARTIES** [151] pentru *învăţarea individuală* şi **CHOCOLATO** [152] pentru *învăţarea colaborativă*.

[19] **I**nstruction_**L**earning (**I**nstruire_**Î**nvăţare)

2. Ontologii şi sistemele autor CREAM, CREAM-TOOLS

Ontologia capacităţilor, **ontologia obiectivelor** şi **ontologia resurselor** sunt folosite în **sistemul autor CREAM** pentru generarea automată a curriculum-ului şi în sistemul autor **CREAM-TOOLS** [153] pentru generarea conţinutului instrucţional într-un ITS.

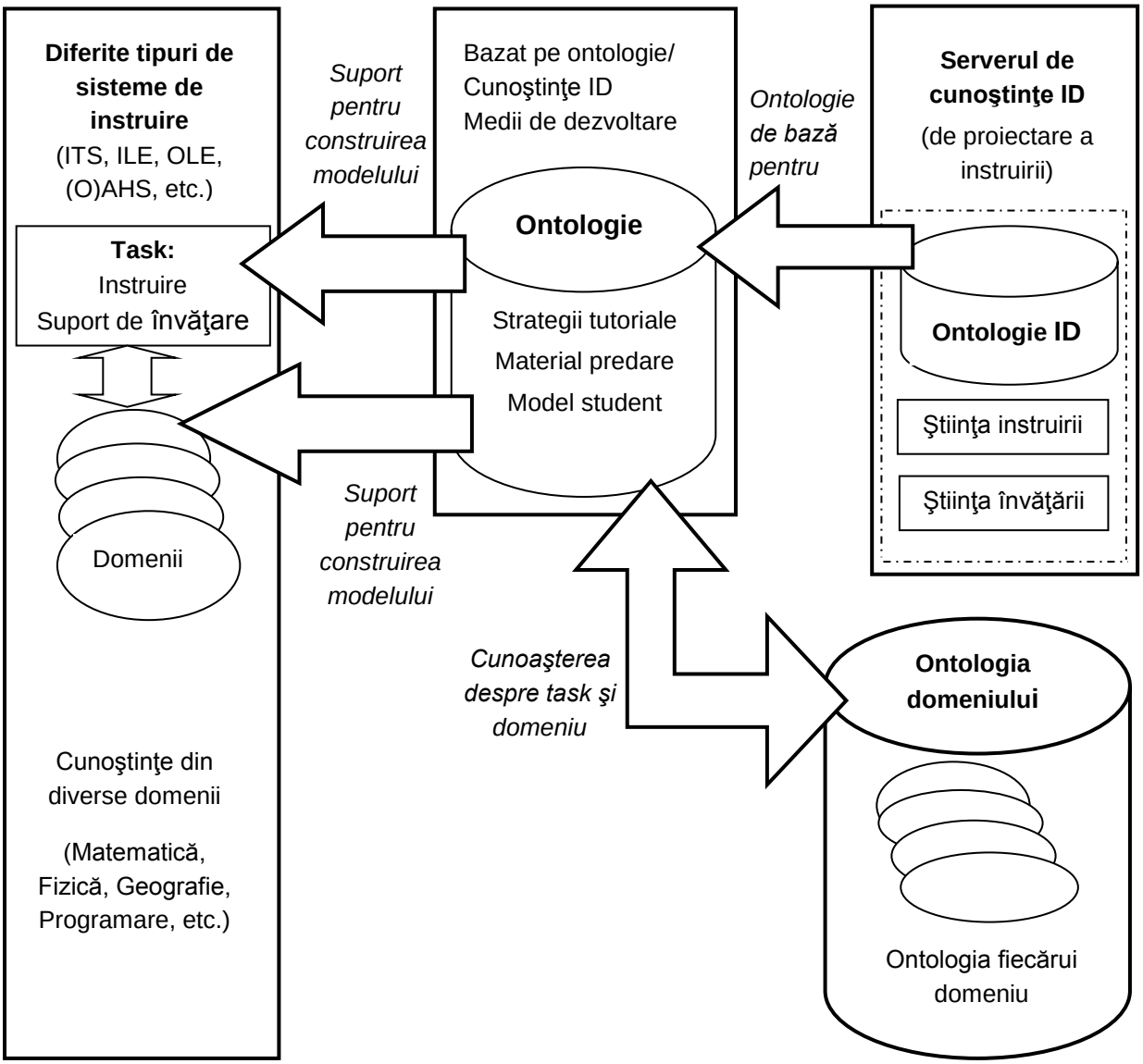


Figura 1-15 Ontologiile de proiectare a instruirii - ontologii de bază pentru sistemele autor

3. Ontologii şi sistemul autor EON

În acest caz, **ontologiile** descriu conţinutul instrucţional, strategiile pedagogice, modelul elevului şi interfaţa. **Sistemul autor EON** (prezentat şi în figura 1-10) poate fi privit ca un meta-sistem de proiectare pedagogică, permiţând definirea unor strategii tutoriale de către profesor [82].

4. Ontologia unui domeniu de instruire şi sistemul de vizualizare a modelului elevului

Instrumentul MERCURIO construieşte automat o **ontologie într-un domeniu de instruire**[20] pornind de la resurse online[21]. **Sistemul** de vizualizare a modelului elevului **VIUM** (Visualisation of Large User Models) [154] permite utilizatorului (elev sau profesor) să selecteze un anumit concept din ontologie. Pe baza noţiunii selectate (şi eventual a altor informaţii despre elev) şi ale inferenţelor realizate, sistemul VIUM afişează utilizatorului conceptele cele mai apropiate semantic de conceptul selectat.

5. Ontologia task-urilor de proiectare pedagogică şi sistemele de asistenţă şi explicare Explor@ şi ADISA

Ontologia task-urilor [155] propune un cadru metodologic comun care să permită construirea unor sisteme explicative. Ontologia identifică şi defineşte componentele task-ului considerat şi noţiunile care vor fi luate în considerare în sistem. **Sistemul explicativ Explor@** se adresează elevilor care vor să se instruiască într-un anumit domeniu folosind resurse web. Sistemul **ADISA** se adresează proiectanţilor de materiale pedagogice.

6. Ontologii şi proiectul DIVA-BCTA

În proiectul DIVA-BCTA (**B**ase de **C**onnaissances pour le **T**élé**a**pprentissage) o **ontologie în domeniul e-learning-ului** susţine crearea unei baze de cunoştinţe şi regruparea expertizei pedagogice în obiecte instrucţionale care vor fi referite cu ajutorul ontologiei (ontologia oferă mijloacele de indexare şi clasificare a obiectelor pedagogice). O **ontologie a task-ului** permite definirea unor cazuri de utilizare a acestor obiecte pedagogice, sprijinind actvitatea de consultare, navigare şi vizualizare a utilizatorului (pedagog sau elev) [156].

7. Ontologii şi documente pedagogice dinamice

Pornind de la o **ontologie a domeniului de instruire**, sistemele **KARINA** şi **SIBYL** generează documente pedagogice adaptate elevului [157]. În sistemul KARINA strategia de prezentare a unui subiect este descrisă printr-o gramatică formală care specifică modul de compunere a resurselor. În sistemul SIBYL, pe lângă ontologia domeniului de instruire, este propusă o ontologie pedagogică, cu reguli care exprimă modalităţile de compunere a fragmentelor[22] în vederea obţinerii documentelor pedagogice.

Exemplele anterioare evidenţiază utilizarea ontologiilor *în sistemele de proiectare pedagogică*, dar şi *în sistemele de diseminare a documentelor pedagogice* sau în *gestiunea obiectelor pedagogice*.

Utilizatorii acestor sisteme sunt atât ***proiectanţii pedagogici*** (cazul mediilor autor, în care accesul la cunoștințele pedagogice declarative din ontologie poate susține procesele decizionale, îmbogăți sau ameliora proiectarea sistemelor de instruire și furnizează o reprezentare explicită și partajată a cunoștințelor pedagogice), cât şi ***elevi*** (cazul sistemelor interactive de învăţare).

[20] În cazul prezentat domeniul instruirii este ştiinţa calculatoarelor

[21] În cazul prezentat resursa online este FOLDOC (Free On-Line Dictionary Of Computing)

[22] Numite "**B**rique d'**I**nformation", reprezintă un fragment al unui document, disponibil pe (cel puţin) un mediu, caracterizat printr-un model conceptual, putând fi inserate într-un document pedagogic.

Tabelul 1-1 prezintă o sinteză a sistemelor prezentate, a ontologiilor pe care se bazează şi a contextului în care sunt utilizate ontologiile în cadrul acestor sisteme.

Tabelul 1-1 Proiecte (sisteme) şi modul de utilizare a ontologiilor

	Ontologie ***Proiect (sistem)***	**Scopul utilizării ontologiei**							
		1	**2**	**3**					
				A	**B**	**C**	**D**	**E**	**F**
MEDII AUTOR	Ontologia capacităţilor, obiectivelor, resurselor ITS *CREAM, CREAM-TOOLS*	√	√		√			√	
	Ontologia modelului elevului pentru vizualizare *VIUM*		√		√				
	Ontologia învăţării colaborative	√						√	
	Ontologie pentru compunerea dinamică a obiectelor pedagogice (OP) *SIBYL şi KARINA*		√	√	√	√		√	√
	Ontologia sintactică şi pedagogică pentru indexarea elementelor unui OP *IMAT*		√	√	√				
	Ontologia OMNIBUS *SMARTIES, CHOCOLATO*	√	√			√			√
	Ontologie pentru un sistem multi-agent *SmartTrainer*	√	√				√		√
	Ontologia unui sistem explicativ *ADISA*	√	√					√	
	Ont. pentru KB şi o bancă de OP interoperabile *DIVA – BCTA*	√	√	√	√	√		√	√
	Ontologie cu scop special (strategii pedagogice) *EON*	√	√					√	
MEDII INTERACTIVE DE ÎNVĂŢARE	Ontologia modelului elevului pentru vizualizare *VIUM*		√		√				
	Ontologia învăţării colaborative	√		√					
	Ontologie sintactică şi pedagogică pentru indexarea elementelor unui OP *IMAT*			√	√				
	Ontologia unui sistem explicativ *Explor@*	√	√						
	Ont. pentru KB şi o bancă de OP interoperabile *DIVA – BCTA*	√		√	√	√		√	

Legenda pentru tabelul 1-1: Scopurile în care este utilizată ontologia:

1. Comunicarea între persoane/organizaţii/sisteme informatice (interoperabilitate)
2. Inferenţă computaţională
3. Ingineria sistemelor educaţionale bazate pe web
   - 3.A. Partajare şi reutilizare
   - 3.B. Căutarea informaţiilor
   - 3.C. Achiziţia,organizarea cunoştinţelor
   - 3.D. Fiabilitate
   - 3.E. Specificare
   - 3.F. Mentenanţă

1.5 CONCLUZII

În acest capitol am realizat o prezentare a domeniului sistemelor de instruire, a teoriilor care stau la baza dezvoltării acestor sisteme şi a direcţiilor de cercetare din acest domeniu pluridisciplinar.

Apărute la începutul secolului al XX-lea, sistemele de instruire au evoluat de la sistemele CAI/CAL clasice (1970-1980) la tutorialele inteligente (ITS) (1980-1990), sistemele bazate pe simulare şi sistemele educaţionale inteligente (bazate pe web-ul semantic).

Această evoluţie a fost determinată nu numai de factori economici şi sociali (aspect mai puţin important în cadrul acestei lucrări), dar şi de *noile tehnologii* şi de *abordările din ştiinţele adiacente domeniului instruirii asistate*. Abordările din plan filozofic (cunoaşterea empirică, cunoaşterea raţională; înnăscut, dobândit) generează abordările psihologice privind procesul învăţării (comportamentală, cognitivă, constructivistă).

Este de subliniat faptul că o teorie a învăţării nu le exclude automat pe celelalte, fiecare reprezentând un model explicativ al unei anumite laturi a procesului de învăţare, şi nu al învăţării în general. Disputele dintre aceste orientări sunt de domeniul trecutului; în prezent critica lasă locul sintezei, complementarităţii şi asimilărilor reciproce. Nu putem privi o teorie ca fiind "mai bună" decât alta, ci doar ca fiind mai *adecvată într-o anumită situaţie de învăţare*. Acest context determină schimbări majore în ştiinţele instruirii. Strategiile aplicate în procesul instruirii trebuie să ia în consideraţie toţi aceşti factori menţionaţi (procesele de învăţare, noile tehnologii, etc.).

Deşi sistemele CAI/CAL, combinate cu tehnicile hypermedia, sunt cele mai răspândite sisteme de instruire asistată, ele prezintă deficienţa de a nu adapta – în mod real - procesul instruirii pentru fiecare elev. Scopul de a construi sisteme inteligente de instruire, caracterizate printr-un nivel ridicat de *individualizare a instruirii* a condus la utilizarea tehnicilor de inteligenţă artificială în cadrul acestor sisteme. Se conturează astfel un nou domeniu de cercetare, cel al *inteligenţei artificiale aplicate în educaţie* (AIED).

Sistemele care folosesc tehnici ale inteligenţei artificiale, cu scopul de a simula un comportament cât mai apropiat de cel al unui profesor uman, sunt tutorialele inteligente (ITS), sistemele hypermedia adaptive (într-o oarecare măsură), într-un cuvânt, *sistemele educaţionale inteligente*.

Între cele două categorii de sisteme (*tutorialele inteligente* şi *sistemele hypermedia adaptive*) există similitudini multiple, dar şi argumente pro/contra.

Am prezentat argumentele care susţin faptul că tutorialele inteligente sunt sisteme bazate pe cunoştinţe, în care *cunoaşterea despre strategiile şi tacticile pedagogice* de instruire trebuie reprezentată **separat** de *cunoaşterea domeniului în care se realizează instruirea*. Evident, cunoştinţele domeniului şi cunoştinţele tutoriale nu sunt independente, dar această separare (între "formă" şi "conţinut"), însoţită de o reprezentare explicită a cunoştinţelor tutoriale, poate remedia într-o măsură apreciabilă ceea ce li se "reproşează" tutorialelor inteligente, şi anume, "*pedagogia implicită*".

Problematica complexă a sistemelor inteligente de instruire conduce la **ideea** că elaborarea lor trebuie să fie un proces ingineresc bazat pe modele riguros fundamentate matematic, proces asistat de un **sistem (ideal, un sistem inteligent) de dezvoltare (un sistem autor)**.

Incorporarea tehnologiilor specifice web-ului semantic în sistemele educaţionale şi în sistemele autor poate conduce la obţinerea aşa-numitei *AAAL* [158] - *Anytime, Anywhere, Anybody Learning ("învăţare oricând, oriunde, de către oricine")*.

Ontologiile - teorii cadru - joacă un rol deosebit de important în reprezentarea cunoştinţelor, impunându-se şi dovedindu-şi din plin utilitatea în sfera cercetărilor privind *sistemele inteligente de instruire asistată de calculator*. De aceea, tot în acest capitol se prezintă aspectele importante ale ontologiilor (definiţii, clasificări, roluri), cu scopul de a identifica modalităţile de implicare şi utilizare a ontologiilor în domeniul nostru de studiu.

În cercetările din inteligenţa artificială - orientate către conţinutul cunoaşterii – *ontologia* este considerată *un concept nucleu*. Ontologiile sunt teorii ale conţinutului prin conceptualizările pe care încearcă să le capteze, prin vocabularul comun (obţinut prin definirea strictă a semnificaţiei conceptelor de bază) şi prin facilitarea comunicării între agenţi. Prin rolul lor de metateorie şi prin potenţialul de sistematizare şi explicitare a cunoaşterii, ontologiile pot furniza tehnici şi teorii efective care să permită acumularea şi interoperabilitatea cunoştinţelor. Oferind veritabile punţi de legătură între teoriile de bază şi tehnologiile fundamentale, ontologiile au un rol important în generaţia următoare de prelucrare a cunoştinţelor.

O importantă concluzie a prezentului capitol a fost aceea că un sistem inteligent de instruire asistată, trebuie să fie un *sistem bazat pe cunoştinţe*, proiectat conform unor *următoarelor principii bine stabilite*:

- Reprezentarea separată a conţinutului instruirii şi a strategiilor de instruire;
- Modularizarea conţinutului instruirii în vederea utilizării multiple sau a reutilizării;
- Crearea unor strategii de predare şi de îndrumare generice, care să poată fi utilizate pentru diferite conţinuturi;
- Reprezentarea explicită a entităţilor pedagogice abstracte;
- Proiectarea la nivel pedagogic, şi nu la nivelul mediului fizic.

Respectarea acestor principii de proiectare poate conduce la obţinerea unor sisteme inteligente de instruire al căror comportament să fie uşor de modificat, sisteme modularizate, care să ofere posibilitatea de a inspecta şi reutiliza cunoaşterea.

Necesitatea ***reprezentării explicite a cunoaşterii pedagogice***, dar şi a proceselor de raţionament (multe dintre acestea fiind, practic, ***procese de planificare pedagogică***, aşa cum va ilustra capitolul 4) pe care trebuie să le realizeze un sistem inteligent de instruire, au condus la ideea realizării *de instrumente autor de dezvoltare (sisteme bazate pe cunoştinţe, susţinute de ontologii)*.

În capitolele următoare voi demonstra valabilitatea acestei soluţii şi modul în care ontologiile pot interveni, pe diferite niveluri, la obţinerea unor rezultate concrete ***în activitatea de proiectare pedagogică***. Deasemenea, având în vedere rolul important al metadatelor și al standardelor, pentru reprezentarea *cunoştinţelor pedagogice*, abordările din lucrare (studiul de caz din capitolul 3 și capitolul 4) folosesc atât ontologiile, cât și elemente de standardizare.

"It would be possible to describe everything scientifically, but it would make no sense; it would be without meaning, as if you described a Beethoven symphony as a variation of wave pressure."

"Ai putea descrie în mod ştiinţific orice, dar ar putea să nu aibă sens; ar putea fi fără semnificaţie, ca şi cum ai descrie o simfonie a lui Beethoven ca pe o variaţie a presiunii unei unde.."

(Albert Einstein, 1879-1955)

2. CONTRIBUŢII METODOLOGICE LA MODELAREA ONTOLOGICĂ A CUNOAȘTERII

2.1 STUDIU PRELIMINAR

2.1.1 ONTOLOGII ŞI REPREZENTAREA CUNOŞTINŢELOR

Aşa cum am ilustrat în capitolul 1, termenul de ontologie denotă rezultatul unor activităţi de analiză conceptuală şi modelare, activităţi realizate prin anumite metodologii.

Din *punct de vedere metodologic*, ontologiile se caracterizează prin adoptarea unei abordări interdisciplinare, în care filozofia şi lingvistica joacă un rol fundamental într-o analiză - cu nivel ridicat de generalitate şi vocabular formulat clar, riguros - a structurii unei realităţi date.

Din *punct de vedere arhitectural*, ontologiile se caracterizează prin rolul pe care îl pot avea într-un sistem informatic (în cazul nostru, un sistem inteligent educaţional).

NIVELUL ONTOLOGIC ÎN REPREZENTAREA CUNOAŞTERII

Reprezentarea şi prelucrarea cunoaşterii sunt elementele esenţiale în obţinerea comportamentului inteligent al unui sistem. Reprezentarea cunoaşterii unui sistem inteligent (de instruire, în cazul nostru) urmăreşte descrierea elementelor universului de discurs în vederea efectuării raţionamentelor care conduc la îndeplinirea obiectivelor sistemului. Această reprezentare se structurează pe mai multe niveluri; primitivele utilizate în reprezentare sunt evidenţiate în tabelul 2-1, pentru fiecare nivel.

O ***observaţie importantă în demersul nostru*** este aceea că prin introducerea nivelului ontologic, *separarea* existentă în sistemele expert şi în sistemele bazate pe cunoştinţe dintre *reprezentarea cunoştinţelor* şi *mecanismele de raţionament* **poate fi completată** de *separarea* între *semantica formală* a unui domeniu (care constrânge interpretarea cunoaşterii) şi *semantica operaţională* (care precizează modul în care cunoştinţele vor fi utilizate în raţionamente).

Tabelul 2-1 Clasificarea primitivelor folosite în reprezentarea cunoştinţelor

Nivel	**Primitive**	**Concepte primitive**	**Facilităţi**	**Interpre-tare**
Implementare	Locaţii de memorie, regiştri			-
Logic	Propoziţii, predicate, funcţii, operatori logici	Predicate	Formalizare	Arbitrară
Epistemologic	Tipuri de concepte, relaţii de structurare	Primitive de structurare	Structurare	Arbitrară
Ontologic	Primitive	Satisfac postulatele semnificaţiei	Semnificaţie	Constrânsă
Conceptual	Relaţii conceptuale, obiecte, acţiuni	Primitive cognitive	Conceptua-lizare	Subiectivă
Lingvistic	Termeni lingvistici	Primitive lingvistice	Limbaj	Subiectivă

ABORDAREA FORMALĂ A CONCEPTUALIZĂRII ŞI A ONTOLOGIEI

Apărute cu scopul iniţial de a susţine achiziţia cunoştinţelor din sistemele informaţionale, ontologiile *nu* au fost riguros definite în raport cu procesul general de reprezentare a cunoaşterii, deoarece:

- În sens filozofic, ontologia este un sistem de categorii care tratează o anumită viziune a universului (sistem care nu va depinde de un limbaj particular);
- În inteligenţa artificială termenul de ontologie referă o disciplină bazată pe o construcţie inginerească, numită ***ingineria ontologică***, constituită dintr-un vocabular specific (utilizat pentru a descrie o anumită parte a realităţii) şi o mulţime de ipoteze explicite privitoare la semnificaţia intensională a cuvintelor; ca urmare, două ontologii care partajează aceeaşi conceptualizare, pot să difere prin vocabularul utilizat.

Pentru a nu genera confuzii, noţiunea de conceptualizare necesită o formalizare adecvată. Conceptualizarea este definită de Genesereth şi Nillson în modul următor:

Definiţia 2-1 [159]: ***Conceptualizarea*** este o *structură* <D**, R**>, unde D reprezintă *domeniul* conceptualizării, iar **R** este o *mulţime de relaţii* în D.

În această accepţiune, conceptualizarea[23] se referă la **relaţiile matematice** din D (numite **relaţiile *extensionale***), care descriu o stare particulară a problemei.

Deoarece este indicat ca *semnificaţia relaţiilor să fie independentă* de starea particulară care descrie domeniul, este necesară şi abordarea aspectului *intensional* al relaţiilor. Ca urmare, Guarino propune o altă accepţiune a termenului de conceptualizare, în care relaţiile **nu sunt** definite pe domenii concrete, ci într-un ***spaţiu al domeniului*** (definiţia 2.2.) şi sunt numite ***relaţii conceptuale*** şi sunt privite ca funcţii definite pe universuri posibile (definiţia 2.4.).

[23] La conceptualizarea definită prin 3.1. se referă Tom Gruber în definirea ontologiei (definiţia 2.2., *ontologia – specificarea unei conceptualizări*).

Definiția 2-2 [109]: **Spaţiul domeniului** este o structură <D, W>, unde D este un domeniu, iar W este *mulţimea stărilor relevante* (mulţimea maximală de stări a elementelor pentru domeniul D, numite şi *lumi posibile* sau *universuri posibile*).

Definiția 2-3 [109]: Dat fiind un spaţiu al domeniului <D, W>, **relaţia conceptuală** ρ^n de aritate n pe <D, W> este o funcţie ρ^n: W $\rightarrow$ $\mathbf{2}^{D^n}$ definită pe W (mulţimea stărilor), cu valori în mulţimea tuturor relaţiilor n-are pe D. Pentru o **relaţie conceptuală generică** ρ, mulţimea $\mathbf{E}_\rho$ ={ρ(w)|w$\in$ W}, conţine extensiunile admisibile pentru ρ.

Definiția 2-4 [107], [109]: O ***conceptualizare*** pentru D este definită ca tuptul **C**=< D, W, $\mathfrak{R}$ >, unde $\mathfrak{R}$ reprezintă mulţimea relaţiilor conceptuale definite pe spaţiul domeniului <D, W>. Conceptualizarea reprezintă, deci, o mulţime de relaţii conceptuale definite pe un spaţiu al domeniului [24].

Trebuie evidenţiat faptul că definirea terrmenului “conceptualizare” diferă (definiţiile 2.1. şi 2.4.). În accepţiunea lui *Genesereth* [159] ***conceptualizarea*** C=<D, **R**> referă o lume particulară (o stare a lucrurilor) şi constituie – mai degrabă - o ***structură a lumii, o structură a universului***. În accepţiunea lui *Guarino* [109], ***conceptualizarea*** **C**=<D, W, $\mathfrak{R}$ > defineşte ***mai multe astfel de structuri***, câte una pentru fiecare stare (lume particulară). Acestea pot fi numite ***structuri ale universului intensional corespunzător conceptualizăriii***.

Definiția 2-5 [109]: Fie C=<D, W, $\mathfrak{R}$ > o conceptualizare. Pentru fiecare univers posibil $w\in$W, structura lumii corespunzătoare sau structura intensională a lui w conformă cu C se defineşte ca: S_{wC} =<D, R_{wC} >, unde R_{wC} ={ρ(w)| ρ$\in\mathfrak{R}$ } reprezintă mulţimea extensiunilor (relative la w) ale elementelor lui $\mathfrak{R}$. Dacă notăm cu S_C mulţimea formată din toate structurile de lumi intensionale ale lui C, atunci:

$\mathbf{S}_C$ = { $\mathbf{S}_{wC}$ | $w\in$W } este ***mulţimea structurilor universului intensional al lui C***.

Definiția 2-6 [101]: Fie limbajul logic **L**, cu vocabularul V. Modelul pentru **L** este definit de structura <**S**, I>, în care **S**=<D, **R**> este o structură a lumii (conform definiţiei 2.1. a conceptualizării), iar funcţia I: V$\rightarrow$D$\cup$**R** este o interpretare care atribuie elementelor din D simboluri constante din V, iar elementelor din **R** simboluri predicative din V. Acest model permite o ***interpretare extensională a limbajului***.

Definiția 2-7 [101]: Numim *interpretare intensională* structura <C, $\mathfrak{I}$>, unde C=<D, W, $\mathfrak{R}$ > este o conceptualizare şi $\mathfrak{I}$: V$\rightarrow$D$\cup\mathfrak{R}$ este o funcţie de interpretare care asociază elementelor din D simboluri constante din V, iar elementelor din $\mathfrak{R}$ - predicate din V. Această *interpretare intensională* este numită *angajment ontologic* pentru L. Dacă

[24] În definiţii, simbolurile care desemnează structuri sau mulţimi de mulţimi sunt boldate.

K=<C, $\mathfrak{I}$ > este un *angajament ontologic* pentru L, se consideră că L *angajează pe* C *prin intermediul lui* K, iar C *este conceptualizarea de bază pentru* K.

Definiţia 2-8 [101]: Dat fiind limbajul **L** cu vocabularul **V** şi un angajament ontologicic **K**=<**C**,$\mathfrak{I}$> pentru **L**, un model **<S,** I**>** va fi ***compatibil*** cu **K** dacă:

a) **S** $\in$ $\mathbf{S}_C$;

b) Pentru fiecare constantă c, I(c) = $\mathfrak{I}$ (c);

c) Există un univers w astfel încât, fiecărui simbol predicativ p îi corespunde (pe baza funcţiei I) o extensie admisibilă din $\mathfrak{I}$(ρ). Cu alte cuvinte, există o relaţie conceptuală ρ astfel încât $\mathfrak{I}$(ρ) = ρ $\wedge$ ρ (w) = I(p). Mulţimea $\mathbf{I}_K$ (**L**) a tuturor modelelor lui **L** care sunt compatibile cu angajamentul ontologic **K** formează *mulţimea modelelor intensionale ale lui* **L**, *corespunzătoare lui* **K**.

Practic, ontologia **O** pentru limbajul **L** *aproximează o conceptualizare* **C** dacă există un angajament ontologic **K**=<**C,** $\mathfrak{I}$> astfel încât modelele intensionale ale lui **L** pentru angajamentul **K** sunt incluse în modelele lui **O** [109], [160] (figura 2-1).

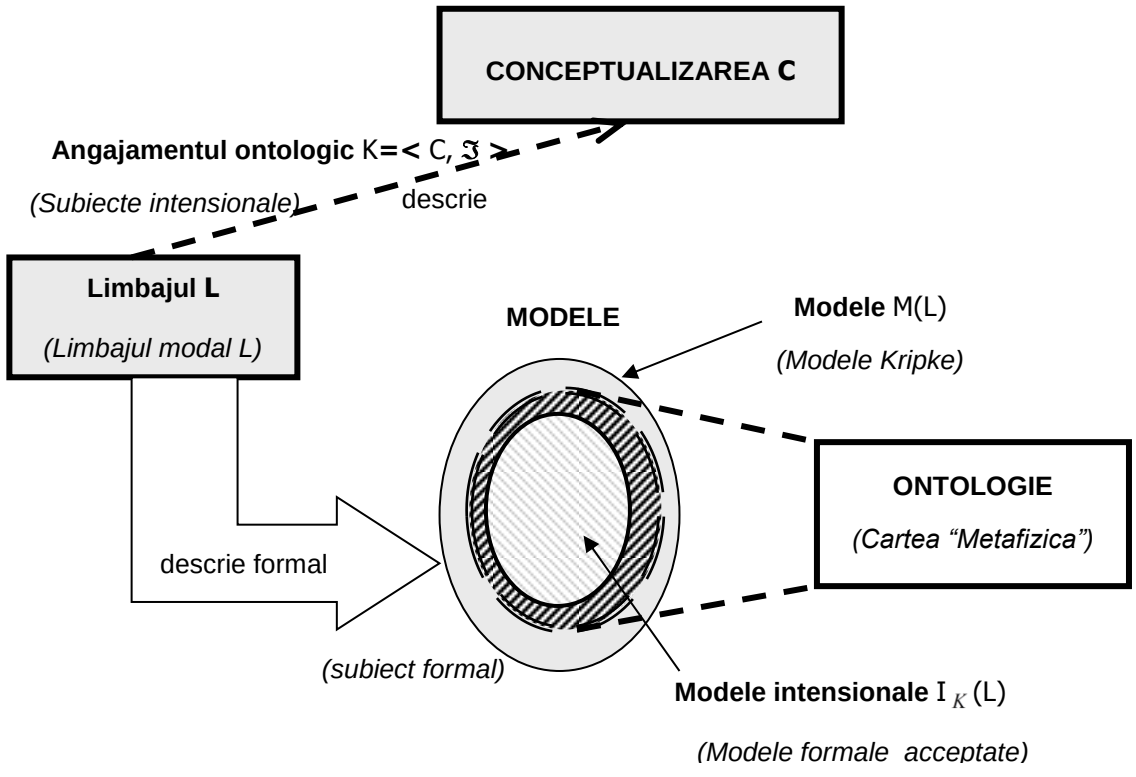


Figura 2-1 Relaţiile dintre vocabular, conceptualizare, angajament ontologic şi ontologie; Ontologie şi semnificaţie (adaptat după [109])

Definiţia 2-9 [109]: O ontologie este angajată la conceptualizarea C dacă.:

- A fost proiectată în vederea caracterizării lui **C**;
- Aproximează **C**.

Definiția 2-10 [107]: Se consideră că ***limbajul* L *este angajat la o ontologie* O**, dacă este angajat la anumite conceptualizări **C,** astfel încât **O** este în concordanţă cu **C**.

În accepţiunea lui Guarino, diferenţa dintre conceptualizare şi ontologie este fixată prin următoarea definiţie:

Definiția 2-11 [107]: O ontologie este o teorie logică destinată semnificaţiei intensionale a unui vocabular formal (vocabularul formal poate fi o parte a unui limbaj logic, un protocol de comunicaţie între agenţi, etc.), o *axiomatizare* (posibil incompletă) ***a conceptualizării*** (o axiomatizare a modelelor intensionale ale limbajului logic) [107], [109] constituie un angajament ontologic al unei conceptualizări particulare a universului.

Dat fiind un limbaj **L**, cu angajarea ontologică **K**, o ***ontologie pentru* L** este o mulţime de axiome proiectate astfel încât mulţimea modelelor sale aproximează cât mai bine posibil mulţimea modelelor intensionale ale lui **L** în concordanţă cu **K**.

Figura 2-2 ilustrează o particularizare pentru figura 2-1 în cazul unui sistem de instruire.

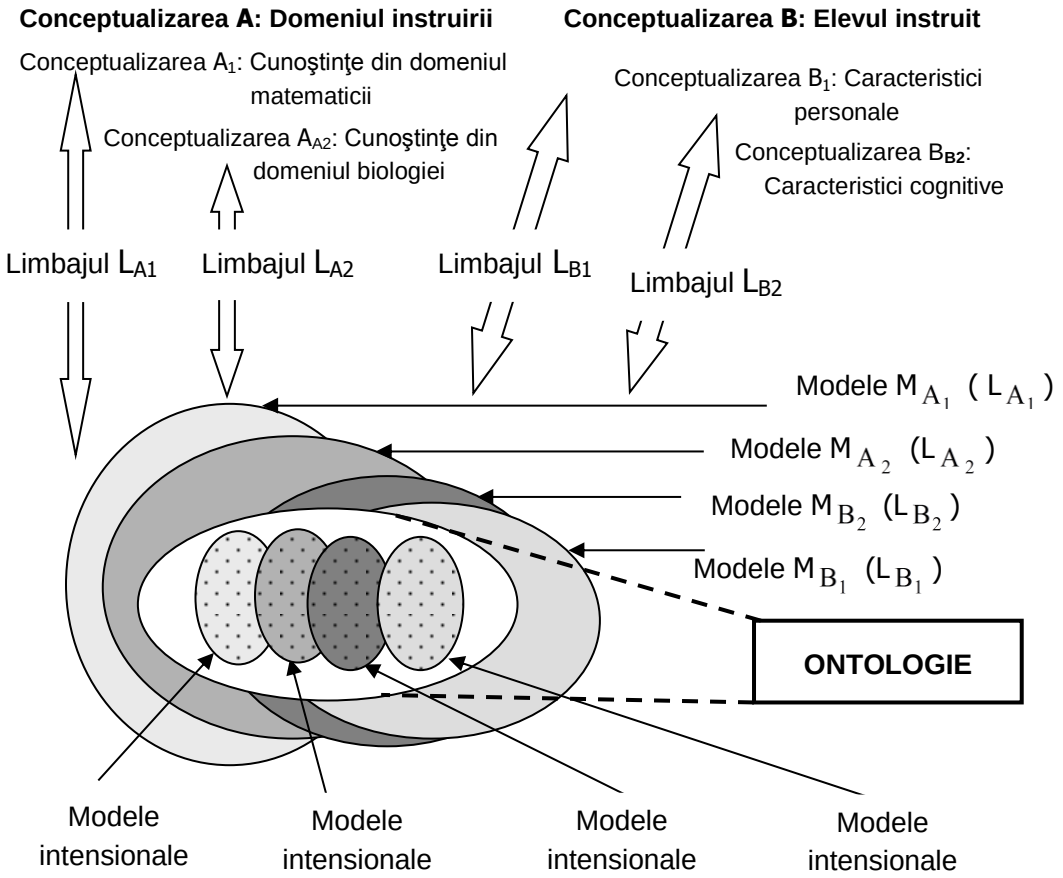


Figura 2-2 Ilustrarea relaţiilor dintre vocabularul, conceptualizarea şi ontologia (parţială) unui sistem de instruire

Observaţii:

1. Ontologia reprezintă o specificare a vocabularului utilizat pentru simbolurile nelogice (tipuri de entităţi, atribute şi proprietăţi, relaţii şi funcţii, constrângeri).
2. Ontologia este dependentă de limbaj.
3. Conceptualizarea este independentă de limbaj.

Așa cum se observă din definițiile anterioare, și noţiunea de **angajament ontologic** - extrem de importantă în cadrul ontologiilor pentru reprezentarea cunoştinţelor - este definită în mod diferit de către cercetători:

- Pentru *Guarino*, angajamentul ontologic (definiţiile 2.7.-2.10.) constituie utilizarea unui concept conform sensului său, deoarece conceptul trebuie să corespundă extensiunii obiectelor existente din universul interpretării.
- *Gruber* defineşte angajamentul ontologic ca reprezentând utilizarea unui vocabular partajat într-o manieră coerentă şi consistentă [103.]

Noţiunea de angajament ontologic apare (indirect) şi în lucrările lui *Bachimont* [122], deoarece existenţa obiectelor este prescrisă de sensul unui concept; angajamentul ontologic permite definirea unei ontologii formale (sau referenţiale). Ontologia[25] este definită ca latice a conceptelor formale caracterizate prin etichete ale căror semantică este definită prin extensiunea obiectelor. Fiecare concept formal este definit printr-un angajament ontologic care specifică ce obiecte trebuie sa existe în domeniu pentru a utiliza conceptul conform specificaţiei lui formale.

PRINCIPII DE BAZĂ ÎN REPREZENTAREA CUNOAŞTERII

Aşa cum se evidenţiază încă din 1993, ***reprezentarea cunoaşterii*** poate fi privită în termenii ***rolurilor*** pe care aceasta le poate avea într-un sistem (inteligent):

- Reprezentarea cunoştinţelor este un *substitut*: obiectele fizice, evenimentele şi relaţiile sunt reprezentate prin simboluri, în scopul memorării şi prelucrării de către un sistem informaţional;
- Reprezintă o *mulţime de angajamente ontologice*: permite reprezentarea explicită a conceptelor care există sau care ar putea exista în domeniul modelat. Conceptualizarea trebuie să fie partajată de către o comunitate de utilizatori (prin angajamentul ontologic);
- Este o *teorie fragmentară a raţionamentului inteligent*: teoria domeniului de aplicaţie poate fi exprimată prin *axiome explicite* sau poate fi *compilată în programe executabile*;
- *Mediu pentru calcul computaţional*;
- *Mediu pentru exprimarea umană* (dimensiunea interpretării: un bun limbaj de reprezentare a cunoştinţelor trebuie să uşureze comunicarea între agenţii umani şi/sau artificiali).

2.1.2 INGINERIA ONTOLOGICĂ

OBIECTIVELE INGINERIEI ONTOLOGICE

Pentru stabilirea unei metodologii clare după care trebuie să se desfăşoare cercetările orientate spre conţinut, cât şi pentru crearea unei punţi de legătură între teoriile de bază şi tehnologiile fundamentale, a apărut conceptul de ***inginerie ontologică***. Scopul cercetărilor orientate spre conţinut este acela de a diversifica aplicaţiile sistemelor bazate pe cunoştinţe şi de a permite ***reprezentarea cunoştinţelor independent de o anumită aplicaţie***.

[25] Definirea unei ontologii de reprezentare a cunoaşterii (conf. Bachimont) constă în definirea, pentru un domeniu dat şi o anumită problemă, a semnăturii relaţionale şi funcţionale a unui limbaj formal şi a semanticilor asociate.

Ingineria ontologică este definită ca "*mulţimea activităţilor implicate în procesul de dezvoltare a ontologiilor şi în ciclul de viaţă al acestora, metodologii, instrumente şi limbaje pentru construirea ontologiilor*" [161].

Ingineria ontologică furnizează baza de construire a modelelor [162]; furnizează tehnologii şi teorii pentru explicarea şi utilizarea proiectării raţionale a bazei de cunoştinţe, nucleul conceptualizării domeniului de interes, definiţii stricte ale semnificaţiilor conceptelor de bază, toate acestea conducând la "acumularea" cunoştinţelor şi interoperabilitatea lor, elemente absolut necesare unei modelări "eficiente" a lumii reale.

Ingineria ontologică urmăreşte stabilirea unor tehnologii şi teorii de bază (*aspecte prezentate detaliat în Anexa I*) pentru următoarea generaţie a ştiinţei cunoştinţelor. *Teoriile de bază* vizează semantica legăturilor, funcţia de meta-model a unei ontologii, modalităţile de construire a unor taxonomii cât mai corecte. *Tehnologiile de bază* vizează limbajele de reprezentare a ontologiei task-urilor, metodologiile de proiectare a ontologiei task-ului/domeniului, precum şi mediile de dezvoltare a acestora [163]. Obţinerea diferitelor tipuri de ontologii va permite folosirea acestora ca blocuri de construcţie în sistemele bazate pe cunoştinţe [145] deci şi în sistemele (inteligente) de instruire [164], [165], oferind astfel o soluţie de **reutilizare a reprezentărilor cunoştinţelor**. Se preconizează astfel apariţia *sistemelor educaţionale din generaţia a cincea*, sisteme educaţionale bazate pe ontologii, cu cunoştinţe din "exterior" (dintr-o "bază ontologică"). Din acest punct de vedere, diferitele tipuri de ontologii (prezentate în secțiunea 1.4.3. din capitolul 1) pot constitui pentru un sistem bazat pe cunoştinţe reprezentări conceptuale ale domeniului, specificarea modelului de raţionament sau suport în reprezentarea cunoştinţelor pentru cazul tratat de către sistem[26].

MODELE ŞI LIMBAJE DE REPREZENTARE A CUNOŞTINŢELOR UTILIZATE ÎN INGINERIA ONTOLGICĂ

Modelele de reprezentare a cunoştinţelor utilizate în ingineria ontologică pot fi grupate după paradigmele conceptuale reliefate:

- Modele bazate pe **cadre** - introduse de către Minsky [166] și propuse iniţial de către Gruber pentru reprezentarea ontologiilor – cu limbajele *KIF, OKBC, F-Logic, Cycl, Ontolingua* și *OCML.*
- Modele bazate e **logici de descriere**[27] – care combină reprezentările intensionale şi extensionale ale cunoştinţelor – cu limbajele *LOOM, CLASSIC, KL-ONE*;
- Modele bazate pe **grafuri conceptuale** – care combină aspectele cognitive şi cele computaţionale legate de reprezentarea cunoaşterii, furnizează un mod de *reprezentare grafică* a *cunoaşterii,* un *model bazat pe teoriile matematice ale grafurilor* [167] *şi ale logicii.*

Toate aceste formalisme permit specificarea cunoştinţelor despre obiectele din domeniu, despre relaţiile dintre concepte şi semantica primitivelor de modelare. Pentru fiecare model există unul sau mai multe **limbaje** care implementează *parţial* sau *total* un anumit model. Unele limbaje

[26] Ontologiile corespund modelelor cunoştinţelor, în sensul sistemelor bazate pe cunoştinţe: modelul conceptual al domeniului, modelul raţionamentului (în SBC-urile care au ca obiectiv operaţional rezolvarea de probleme) şi modelul conceptual al cazului tratat.

[27] LD (engl., "Description Logics")

sunt *operaţionale* (oferă mecanisme de raţionament), altele permit doar *specificarea declarativă* a cunoştinţelor.

În familia limbajelor de reprezentare a cunoştinţelor, *limbajelor "clasice"* li se alătură *limbajele adaptate web*, care utilizează sintaxa xml. În ceea ce privește limbajele de reprezentare a ontologiilor pentru web, urmărind modelul stratificat propus de Tim Berners-Lee [168], au fost studiate caracteriticile pentru XML, RDF/RDF(S), SHOE, XOL, OIL, OIL+DAML - bazate pe sintaxa XML/RDF, OWL **(**OWL LITE, OWL DL şi OWL FULL) și CL (cu dialectele CLIF, CGIF și XCL) [169].

Deoarece nu toate paradigmele şi limbajele studiate permit reprezentarea aceloraşi componente şi tipuri de raţionamente, oferind niveluri de expresivitate și servicii inferenţiale diferite, la implementarea unei ontologii puternic formalizate și alegerea limbajului de reprezentare trebuie luate în considerare cerinţele aplicaţiilor care se vor folosi ontologia.

METODOLOGII, METODE ŞI INSTRUMENTE ÎN INGINERIA ONTOLOGICĂ

Studiul detaliat al metodologiilor, metodelor și instrumentelor folosite în ingineria ontologică și principiile de bază în construirea ontologiilor este prezentat în ***Anexa I***. În această secțiune doar ***le vom enumera***, ***vom formula concluziile studiior comparative, evidențiind apoi elementele utile în demersul nostru.***

Fiind componente software, ontologiile trebuie privite ca obiecte tehnice evolutive, al căror ciclu de viaţă trebuie specificat. Activităţile din ingineria ontologică sunt: *activităţi de pre-dezvoltare*, de gestiune a proiectului (planificare, control, asigurarea calităţii); *activităţi de dezvoltare* (specificare, conceptualizare, formalizare, implementare); *activităţi de post-dezvoltare* (întreţinere, utilizare); *activităţi complementare* (validare, evaluare, documentare). Astfel, metodologiile şi instrumentele folosite în ingineria ontologică pot viza diverse aspecte din cele enumerate anterior; un studiu al metodologiilor şi instrumentelor folosite în ingineria ontologică poate fi găsit în [170].

Ne interesează în mod special *metodologiile*[28] *şi instrumentele pentru construirea ontologiilor (pornind de la zero)* şi *cele pentru evaluarea acestora.* În cadrul acestor metode/metodologii au fost urmărite etapele propuse pentru construirea/dezvoltarea ontologiei, existența unor ontologii dezvoltate pe baza metodei/metodologiei, domeniile de utilizare a ontologiilor identificate, aplicaţiile care folosesc ontologiile respective şi existenţa unor instrumente suport.

Tabelul 2-2 ilustrează o sinteză a metodelor şi metodologiilor pentru construirea ontologiilor.

[28] Conform standardelor (IEEE, 95) şi (IEEE, 90) o ***metodologie*** reprezintă o succesiune cuprinzătoare şi integrată de tehnici sau metode pentru creearea unei teorii generale despre modul în care ar putea fi realizată o categorie de activităţi; o ***metodă*** reprezintă un proces *ordonat* sau o procedură folosită în ingineria unui produs sau realizarea unui serviciu; o ***tehnică*** este o procedură tehnică şi managerială folosită pentru atingerea unui anumit obiectiv. Metodele şi tehnicile sunt componente ale unei metodologii; metoda necesită o ordine, tehnica – nu.

Tabelul 2-2 Studiu comparativ al metodologiilor și metodelor din ingineria ontologică

Criteriu		Cyc	Ushold& King	METHON TOLOGY	On-To-Knowledge	Grüninger& Fox	KAKTUS	SENSUS
Moştenire din ingineria cunoştinţelor		parţial	parţial, SBC	mare	necunoscut	mică	mare	nu
Propunerere ciclu de viaţă		da	-	da	da	da	da	nu
Dependenţa de aplicaţie		independentă	independente	independentă	dependentă	semi-dependentă	dependentă	semi-dependentă
Mod identificare concepte		top-down	middle-out	middle-out	toate	middle-out	top-down	nespecificat
Utilizează ontologii nucleu		da	nu	nu	posibil	nu	nu	da
Proiecte în care se aplică		1	1	mai multe	mai multe	1	1	1
Ontologii create, aplicaţii		1	1	mai multe	mai multe	1	1	1
Domenii		micro-teorii	1	mai multe	mai multe	1	1	1
Managementul proiectului		nu	nu	propus	descris	nu	nu	nu
Procese orientate dezvoltare	pre-dezvoltare	nu	nu	nu	nu	nu	propus	nu
	dezvoltare	propus parţial	propus parţial	descris	descris, propus	descris, propus	descris	descris, propus
	post-dezvoltare	nu	nu	parţial	nu	nu	descris	nu
Instrumente suport		da	nu	da	multe	nu	nu	nu
Gard detaliere		mic	mic	mare	mediu	mic	foarte mic	mediu
Recomandări formalizare		logică	nu	nu	reţele semantice	logică	nu	reţele semantice

Tabelul 2-3 Studiu comparativ al principalelor instrumente existente pentru construirea ontologiilor

Criteriu	OilED	OntoEdit	WebODE	Protégé	Ontolingua	OntoSaurus	WebOnto	DOE
Model al cunoştinţelor	LD	cadre	cadre	cadre	cadre	LD	cadre	entitate/relaţie
Limbaj de reprezentare	DAML+OIL	FOL	FOL	FOL	FOL	LOOM	FOL	nu
Axiome	DAML+OIL	FLOGIC	WAB	PAL	KIF	LOOM	OCML	nu
Ierarhii grafice	nu	nu	da	da	da	nu	da	da
Verificarea coerenţei	da	da	da	da	nu	da	da	nu
Motor inferenţă	FaCT	OntoBroker	Prolog	PAL	ATP (extern)	da	da	nu
Suport metodologic	nu	da	da	nu	nu	nu	nu	da

Cele mai cunoscute *medii/instrumente care permit construcția* ontologiilor, *fără a aplica o anumita metodologie (**Anexa I**)* sunt *OilED, OntoEdit, ODE/WebODE, Protégé-2000, Ontolingua Server, OntoSaurus, WebOnto* și *DOE.* Tabelul 2-3 prezintă rezultatele studiului comparativ al instrumentelor care permit construirea ontologiilor.

Au fost studiate, deasemenea, metodologiile, metodele și instrumentele pentru *evaluarea ontologiilor*: Metodologia lui *Goméz-Pérez*[29] *de evaluare a taxonomiilor* (instrumentul *ONE-T*), metodologia *Ontological Constraint Management* (instrumentul *OCM*) și *metodologia propusă de Gaurino*, care verifică constrângerile impuse prin meta-proprietăţile conceptelor (rigiditate, identitate, unitate şi dependenţă) (instrumentele *OntoCLEAN*, *OntoAnalyzer* şi *OntoGenerator*).

Tot în ***Anexa I*** sunt prezentate și *alte metodologii, metode şi instrumente studiate*, care nu fac vizează direct obiectivele acestei lucrări: metode şi instrumente *pentru reingineria ontologiilor*, pentru *construirea cooperativă a ontologiilor*, pentru *achiziţia automată (semi-automată) a cunoştinţelor domeniului* sau pentru *fuzionarea/alinierea ontologiilor*

2.2 OBIECTIVE

Studiul realizat în secțiunea 2.1.2, asupra unor aspecte din ingineria ontologică, a condus la o serie de concluzii, pe baza cărora am identificat cele două obiective ale prezentului capitol.

Concluziile analizei metodologiilor/metodelor de construire a ontologiilor sunt:

1. Niciuna dintre metodologii nu acoperă toate procesele recomandate de IEEE 1075-1995; majoritatea abordărilor vizează doar procesul de dezvoltare a ontologiilor, ignorând de cele mai multe ori activitatea de evaluare sau ciclul de viaţă al ontologiilor.
2. Fiecare grup propune propria metodologie, neexistând un cadru unificator. Metodologia SENSUS este complet diferită de toate celelalte.
3. În momentul de faţă, aceste metodologii pot fi considerate doar puncte de plecare în propunerea unei metodologii unanim acceptate sau standardizate.

Aceste concluzii au condus la un prim obiectiv: **propunerea unei metode integrate de inginerie ontologică (MIIO)**, în subcapitolul 2.3.

În ceea ce privește modelele și limbajele de reprezentare a cunoștințelor, ambele se caracterizează prin *expresivități și capacități inferențiale diferite*:

- ***Expresivitate***:
  - ***Conceptele*** (ierarhii), ***relaţiile binare*** şi ***instanţele*** sunt *singurele componente ontologice* care pot fi reprezentate *în **toate** limbajele prezentate*. Limbajele Ontolingua, LOOM, OCML, OIL, DAML+OIL şi OWL sunt cele mai expresive, deoarece permit crearea unor taxonomii exhaustive[30], cu subclase disjuncte.
  - ***Relaţiile n-are*** sunt admise doar în limbajele Ontolingua şi SHOE, în celelalte limbaje putând fi reprezentate doar prin descompunerea lor în relaţii binare.

[29] În cadrul capitolului următor, ontologiile vor fi evaluate aplicând metodologia lui Gómez-Pérez.

[30] Într-o astfel de taxonomie, o subclasă este definită ca partiţie a unei clase; clasa părinte reprezintă reuniunea tuturor claselor copil. Clasificare poate să nu fie completă, în sensul în care pot exista instanţe ale clasei părinte care să nu fie incluse în orice subclasă a clasei părinte.

- ***Funcţiile*** pot fi uşor definite în Ontolingua, LOOM, OCML, OIL, DAML+OIL şi OWL. În Flogic ele pot fi definite printr-o relaţie şi o axiomă suplimentară care restricţionează numărul valorilor pe care le poate avea relația.
- ***Axiomele formale*** sunt cele mai puternice mijloace de reprezentare a cunoaşterii în ontologii, putând fi definite în Ontolingua, LOOM, OCML şi Flogic.
- ***Regulile*** pot fi definite doar în LOOM şi OCML.
- ***Procedurile*** pot fi definite doar în Ontolingua (chiar dacă nu pot fi executate), LOOM şi OCML.

- ***Mecanismele inferenţiale:*** Cu excepţia motorului de inferenţă din OIL (FaCT), toate celelalte motoare permit doar deducerea unor noi cunoştinţe sau verificarea inconsistenţei cu ajutorul axiomelor formale. Mecanismele de inferenţă din LOOM şi OIL permit clasificarea automată a conceptelor.

Figura 2-3 sintetizează relaţiile dintre principalele metode şi metodologii, instrumente şi limbaje analizate (studiul detaliat în Anexa I) și utilizate în ingineria ontologică.

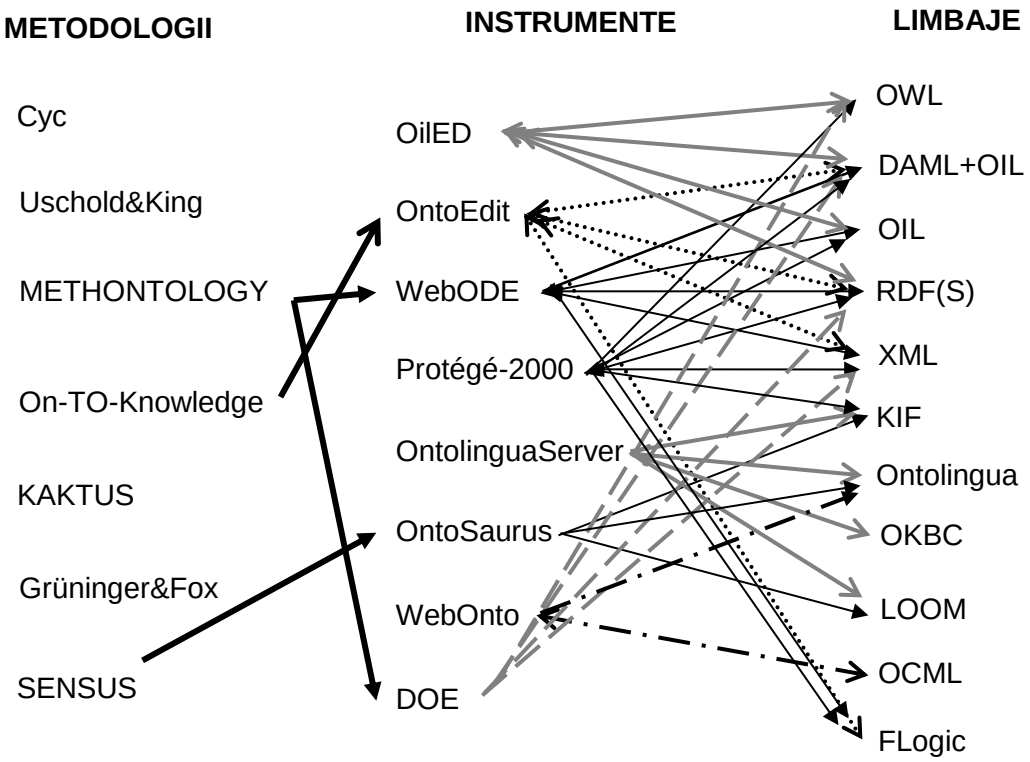


Figura 2-3 Sinteză: Metodologii, instrumente şi limbaje pentru ontologii

La concluziile formulate anterior, putem adăuga:

1. Singurele corespondenţe între metodologiile de construire a ontologiilor şi instrumente sunt cele dintre MERTHONTOLOGY/WebODE şi On-To-Knowledge/OntoEdit. Deoarece nu există suport tehnologic pentru majoritatea metodologiilor propuse, cea mai mare parte a instrumentelor vizează doar câteva activităţi (proiectare şi implementare) ale ciclului de viaţă al ontologiei.
2. Puţine dintre instrumentele de construire a ontologiilor prezintă similarităţi; de aceea ele nu sunt, de obicei, capabile de interoperabilitate. Ca urmare, vor trebui rezolvate

probleme importante de integrare şi aliniere a ontologiilor şi a limbajelor în care acestea sunt reprezentate.

3. Limbajele de reprezentare a ontologiilor bazate pe web sunt încă în faza de dezvoltare, evoluând continuu, ceea ce conduce la dificultăţi în actualizarea tehnologiilor pentru managementul ontologiilor.
4. În privinţa instrumentelor, principala problemă identificată este lipsa unor medii integrate de dezvoltare a ontologiilor. Cu mici excepţii (OntoEdit, Protégé 2000 sau WebODE), majoritatea instrumentelor sunt module "izolate", care rezolvă un anumit tip de probleme.

Ca urmare, OBIECTIVUL 2 al studiului din acest capitol este *propunerea şi validarea teoretică şi experimentală a unei o metode şi a unui instrument de creare şi operaţionalizare a unei ontologii puternic formalizate, care să poată fi utilizată în sistemele bazate pe cunoștințe într-un mod cât mai independent de scopul operațional al acestor sisteme. Acest obiectiv constituie subiectul capitolului următor al lucrării.*

2.3 CONTRIBUȚII PRIVIND O METODĂ INTEGRATĂ DE INGINERIE ONTOLOGICĂ (MIIO)

Având în vedere cercetările din ingineria ontologică, prezentate în sumar în secțiunea 2.1.2 și detaliat în Anexa I, în special cele legate de metodologiile şi instrumentele existente şi concluziile formulate anterior, consider că ***procesul de construire a unei ontologii poate fi integrat în ciclul de viaţă al ontologiei*** în modul prezentat în figura *2*-4. Ingineria ontologică este compusă în special din fazele de modelare şi operaţionalizare a ontologiei (etapele 2, 3), însă acestea depind într-o mare măsură de fazele 1, 4 şi 5.

Ca urmare, *propun o **M**etodă **I**ntegrată de **I**nginerie **O**ntologică* (**MIIO**), caracterizată prin:

1. ***Integrează o serie de elemente şi norme*** (prezentate detaliat în Anexa I) ***importante, prezente în alte metodologii***.
2. Procesul de construire a ontologiilor se bazează pe ***principiile ingineriei ontologice*** (enunțate și analizate (identificarea fazelor în care se aplică principiile și entitățile vizate) în Anexa I).
3. Respectă ***standardul IEEE 1075-1995*** [171] pentru dezvoltarea produselor software.

Metodologia propusă (numită MIIO), pe care o vom aplica la dezvoltarea ontologiei din capitolele următoare, ia în considerare (caracteristica 1) următoarele elemente și norme:

a) Considerarea *ciclului de viaţă al ontologiilor* (propuse prin On-To-Knowledge şi METHONTOLOGY);
b) Strategiile de *identificare a conceptelor (propusă de Uschold&King*);
c) Întrebările de *competenţă (din Grüninger&Fox)*;
d) Criteriile de evaluare din *METHONTOLOGY şi Goméz-Pérez*.

Metoda MIIO constă în cinci faze, fiecare dintre acestea cuprinzând, la rândul lor, mai multe etape. Fazele popuse sunt:

I. Studiul fezabilităţii şi al mediului;
II. Modelarea ontologică (specificare, conceptualizare, formalizare şi implementare);
III. Operaţionalizarea (utilizarea) ontologiei;
IV. Evaluarea, constând în verificarea şi validarea ontologiei şi testarea aplicaţiei care foloseşte ontologia;
V. Documentarea.

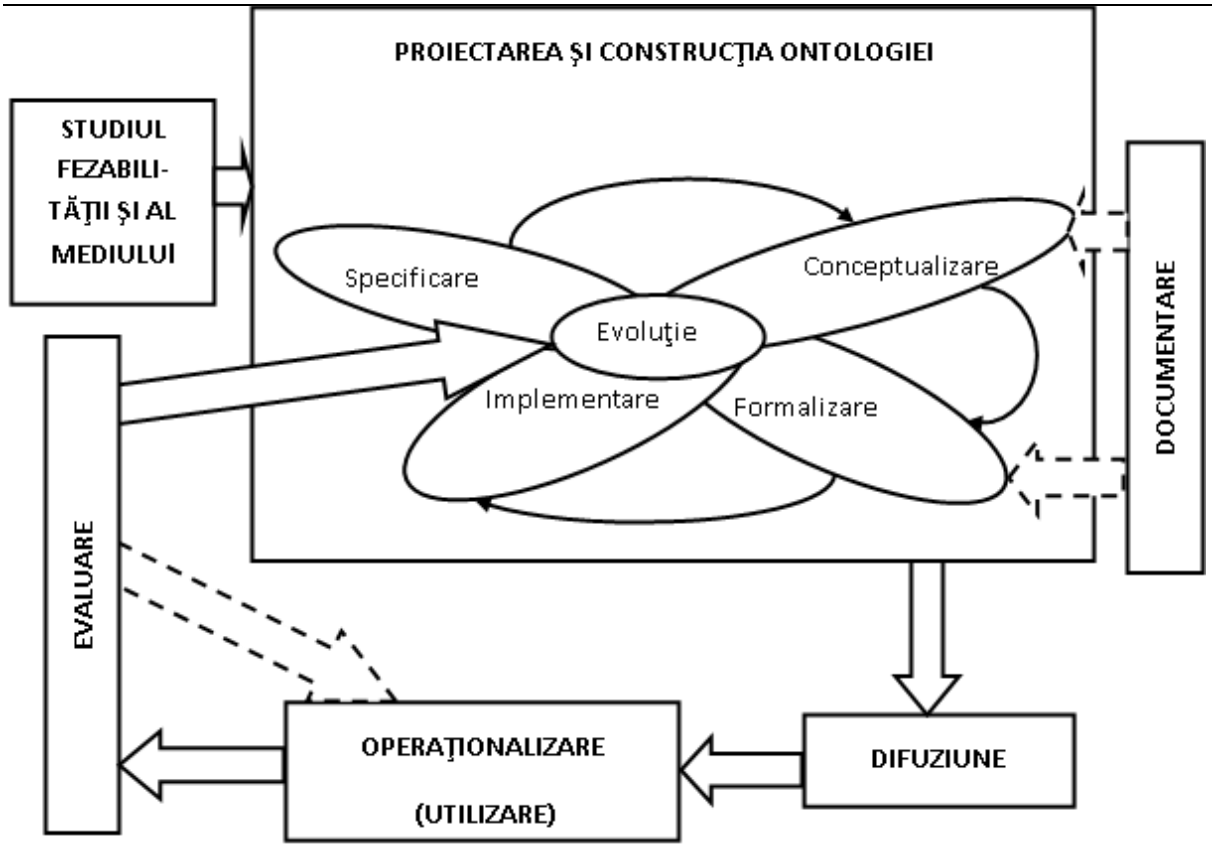


Figura 2-4 Ciclul de viaţă a unei ontologii

***Faza I*: Studiul de fezabilitate şi al mediului**

Studiul[31] este realizat în faza de pre-dezvoltare, înaintea modelării ontologice. Studiul de fezabilitate urmăreşte să răspundă la întrebări cum sunt cele legate de posibilităţile reale de construire a ontologiei, de oportunitatea construirii acesteia; studiul mediului vizează problemele legate de platformele pe care ar putea fi utilizată ontologia şi de aplicaţiile în care aceasta ar putea fi utilizată.

***Faza a II-a*: Modelarea ontologică**

În toate procesele constituente ale acestei faze se aplică principiile de dezvoltare a ontologiilor (prezentate în Anexa I), cât şi cele prezentate în tutorialul ”Ontology 101” [172]: (1) nu există un singur model ”corect” al unui domeniu; (2) dezvoltarea ontologiei este obligatoriu un proces iterativ; (3) conceptele incluse în ontologie trebuie să fie cât mai aproape de obiectele (fizice sau logice) şi de relaţiile din domeniul de interes.

Etapa I-a: Captura nevoilor (cerinţe-specificaţii de modelare ontologică) presupune delimitarea cât mai precisă a domeniului cunoştinţelor, care va permite extragerea termenilor de descriere a cunoştinţelor domeniului, a cunoştinţelor de raţionament şi a cunoştinţelor de nivel înalt (comune mai multor domenii). Etapa constă în:

- Definirea domeniului care va fi reprezentat prin ontologie: se va aplica principiul angajamentelor ontologice minimale (P6);
- Definirea contextului de utilizare a ontologiei;
- Definirea scopului aplicaţiei care va utiliza ontologia;

[31] Studiul este propus în METHONTOLOGY şi On-To-Knowledge

- Definirea caracteristicilor aplicaţiei care va utiliza ontologia (necesară pentru alegerea strategiei de identificare a conceptelor ontologiei: bottom-up, top-down, middle-out);
- Definirea directivelor de concepţie a ontologiei: convenţii de nume, mărimea ontologiei, etc. (conf. Metodei Ushold&King);
- Identificarea surselor de informare (experţi în domeniu, identificarea unor ontologii sau a unor baze de cunoştinţe în vederea reutilizării, publicaţii ştiinţifice, etc.);
- Stabilirea categoriilor de utilizatori şi a scenariilor de utilizare a ontologiei;
- Definirea întrebărilor de competenţă informală (conform metodologiei Gruninger&Fox întrebările de competenţă sunt cele exprimate în limbaj natural, la care trebuie să răspundă ontologia creată; conform metodologiei On-To-Knowledge întrebările de competenţă sunt utile în realizarea documentului de specificaţii şi la evaluarea ontologiei);

Etapa a II-a: Analiza surselor de informare în vederea stabilirii termenilor care vor fi incluşi în ontologie.
Ponind de la specificaţii, se analizează întrebările de competenţă pentru găsirea termenilor adecvaţi care vor fi incluşi în ontologie, se completează această listă cu termeni care apar în alte surse (scenarii de utilizare, documente, interviuri, etc.) şi se reţin termenii candidaţi în funcţie de potenţialul acestora de a reprezenta domeniul.

Etapa a III-a: Conceptualizarea ontologiei, constând în:
- Construirea ontologiei conceptuale, printr-un proces de modelare informală sau semi-formală, prin elaborarea unei taxonomii de bază a conceptelor şi a relaţiilor, adăugarea rolurilor/atributelor, a restricţiilor şi a relaţiilor netaxonomice.
- Reutilizarea şi/sau integrarea unor ontologii sau modele ale cunoştinţelor existente (facultativ).
- Semi-formalizarea ontologiei (facultativ).

Etapa a IV-a: Evaluarea ontologiei conceptuale
- Verificarea ontologiei, realizată la nivelul conceptual de către experţii în modelare şi experţii în domeniu.
- Corectarea ontologiei, pe baza recomandărilor experţilor, dar şi a principiilor ingineriei ontologice P1, P2, P3, P4, P5, P7, P9, P10 (Anexa I, figura AI - 16).

Verificarea şi corectarea conceptualizării au loc iterativ, până la obţinerea unui nivel satisfăcător de completitudine, coerenţă şi precizie.

Etapa a V-a: Documentarea ontologiei conceptuale

Etapa a VI-a: Formalizarea ontologiei constă în structurarea şi formalizarea conceptualizării (pentru a construi o ontologie specificând terminologia şi semantica domeniului cu ajutorul unui model dotat cu o semantică formală). În această etapă poate fi utilizat un editor de ontologii. Etapa de formalizare poate fi completată cu una de integrare, în care în cadrul ontologiei construite sunt importate una sau mai multe ontologii formale existente.

Etapa a VII-a: Evaluarea ontologiei formale, constând în:
- Validarea ontologiei formale (**în cazul nostru, va fi realizată automat, cu ajutorul instrumentului OntL_CG, propus în subcapitolul 3.6**)

- Rafinarea ontologiei formale, necesitând satisfacerea principiului P5, al extinsibilităţii monotonice maxime.

Aceste procese sunt iterative.

Etapa a VIII-a: Documentarea ontologiei formale

Faza a III-a: Utilizarea ontologiei în cadrul unor aplicaţii

Etapa a IX-a: Operaţionalizarea ontologiei

Operaţionalizarea ontologiilor se poate realiza în două moduri:

- Prin reutilizarea unor componente informatice predefinite, în sensul în care structurilor din model li se asociază componente informatice reutilizabile (task-uri generice); fiecărui task îi corespunde un o comandă (shell) care pune în practică o metodă de rezolvare a unei probleme specifice, iar operaţionalizarea constă în asamblarea diferitelor componente corespunzătoare task-urilor identificate în model;
- Prin utilizarea limbajelor de operaţionalizare, în care cunoştinţele expertului sunt exprimate în sintaxa propusă prin limbaj; această abordare aduce mai multă flexibilitate.

Etapa a X-a: Testarea modului de utilizare a ontologiei operaţionale și testare/experimentare a ontologiei operaţionale. Activitățile implicate în această etapă pot fi:

- Proiectarea unei interfeţe de comunicare ontologie-aplicaţie
- Utilizarea ontologiei operaţionale (utilizarea ontologiei în aplicație)
- Analiza rezultatelor obținute
- Analiza unor indicatori de inteligență ai aplicației (în cazul ontologiilor propuse în această lucrare, indicatorii ar putea fi adaptabilitatea și utilitatea ontologiei).

Faza a IV-a: Evaluarea ontologiei

Cuprinde validarea și verificarea ontologiei, cu elemente din METHONTOLOGY şi Goméz-Pérez: criterii de coerență, completitudine și precizie și detectarea erorilor de inconsistență, incompletitudine și de redundanță (prezentate detaliat în Anexa I).

Faza a V-a: Documentarea ontologiei, care include (conform standardului OWL):

- Crearea dicționarului de clase;
- Crearea dicționarului de proprietăți;
- Crearea dicționarului de axiome;
- Crearea dicționarului de instanțe.

Modul de aplicare (parțială) a metodologiei propuse, MIIO, va fi ilustrat în capitolul 3 al lucrării (validarea ontologiei) și capitolul 4 al lucrării, într-un sistem autor.

2.4 CONCLUZII ŞI CONTRIBUŢII

Reprezentarea şi prelucrarea cunoaşterii sunt elementele esenţiale în obţinerea comportamentului inteligent al unui sistem. Prezentul capitol aduce o serie de contribuții metodologice la modelarea ontologică a cunoașterii.

Studiul preliminar (subcapitolul 2.1) evidențiază rolul nivelului ontologic în reprezentarea cunoașterii (secțiunea 2.1.1), abordările formale existente pentru conceptualizare, ontologie și angajament ontologic (secțiunea 2.1.2) și formulează o serie de concluzii asupra unor aspecte din ingineria ontologică (secțiunea 2.1.2.). Menționăm că studiul detaliat (al obiectivelor ingineriei ontologice, al modelelor și limbajelor de reprezentare a cunoștințelor, precum și a metodologiilor, metodelor și instrumentelor din ingineria ontologică) care stă la baza acestor concluzii se găsește în *Anexa I* a lucrării.

Interesul nostru pentru ingineria ontologică s-a concentrat în jurul a trei poli: *concepția, formalizarea* și *operaționalizarea* unei ontologii, deoarece o ontologie de proiectare pedagogică (mai precis a strategiilor și tehnicilor pedagogice) are un rol extrem de important *în cadrul unui sistem autor bazat pe cunoștințe*.

Concluziile analizei metodologiilor/metodelor de construire a ontologiilor au condus la un prim obiectiv (tratat în acest capitol al lucrării): **propunerea unei metode integrate de inginerie ontologică (MIIO)**, bazate pe *principiile ingineriei ontologice, pe metodele ingineriei ontologice* (intergrează elemente precum considerearea ciclului de viață al ontologiilor din On-To-Knowledge și METHONTOLOGY, strategiile de identificare a conceptelor propuse de Uschold&King, întrebările de competență din Grüninger&Fox și criteriile de evaluare din *METHONTOLOGY şi Goméz-Pérez)* și *pe standardele din ingineria software*.

Trebuie menționat că metoda pe care am propus-o, MIIO, a fost aplicată în capitolele 3 (validarea ontologiei) și 4 ale acestei lucrări. Metoda nu a putut fi validată în totalitate. Cu toate acestea, având în vedere că integrează o serie de elemente din alte metode și metodologii validate deja în decursul mai multor ani, putem considera că metoda MIIO este validă.

Concluziile asupra modelelor, limbajelor și instrumentelor de reprezentare a cunoașterii ontologice au condus la un al doilea obiectiv: *propunerea şi validarea teoretică şi experimentală a unei o metode şi a unui instrument de creare şi operaţionalizare a unei ontologii puternic formalizate, care să poată fi utilizată în sistemele bazate pe cunoștințe într-un mod cât mai independent de scopul operațional al acestor sisteme. Acest obiectiv constituie subiectul capitolului următor al lucrării.*

"Mathematics is the language with which God has written the universe."

"Matematica este limbajul în care Dumnezeu a descris universul."

(Galileo Galilei, 1564-1642)

3. CONTRIBUŢII PRIVIND FORMALISMUL MODELĂRII CONCEPTUALE A DOMENIULUI

3.1 OBIECTIVE

Într-un sistem inteligent bazat pe cunoştinţe sunt necesare două categorii de cunoştinţe:

1. Cunoştinţe despre realităţile obiective din domeniul de interes (obiecte, relaţii, evenimente, stări, etc., din domeniu);
2. Cunoştinţele despre modul în care poate fi atins un tip de obiective, cunoştinţele despre modul în care va fi rezolvată problema vizată.

Dacă iniţial, termenul de *cunoştinţe ale domeniului* (cadrul CommonKADS) era folosit doar pentru *cunoaşterea factuală*, mai nou, termenul mixează cunoştinţele factuale cu cele de rezolvare a problemelor, deoarece sistemele bazate pe cunoştinţe utilizează lanţuri de inferenţă structurate în metode de rezolvare a problemei. Acest aspect este prezentat în figura 3-1, care ilustrează o schemă a procesului de reprezentare a cunoştinţelor.

Aşa cum s-a ilustrat în figura 3-1, dar şi în cadrul capitolului 1 al acestei lucrări, ***sistemele (educaţionale inteligente) bazate pe cunoştinţe*** au o arhitectură modulară (figura 3-2), care separă *cunoştinţele declarative* ale domeniului de *mecanismele de raţionament*. Acest lucru conduce la ideea de a putea folosi mecanisme de raționament diferite, în funcție de obiectivul vizat, asupra acelorași reprezentări ale cunoștințelor.

În cazul ***sistemelor de gestiune a cunoştinţelor***, mecanismul de manipulare a cunoştinţelor este **serverul de cunoştinţe**, care efectuează cereri asupra unei baze de cunoştinţe şi transmite răspunsul utilizatorului (figura 3-3). În acest context, ontologiile sunt privite ca şi **componente**[32] (interne sau externe) ale **sistemelor bazate pe cunoştinţe**, care permit o reprezentare conceptuală şi explicită a cunoştinţelor şi asigură independenţa între această reprezentare şi mecanismele de raţionament *din cadrul sistemului bazat pe cunoştinţe*. Utilizarea ontologiilor garantează modularitatea, deoarece *fiecare motor de inferenţă poate fi dezvoltat independent de domeniu, fiind însă determinat de scenariul de utilizare vizat***.**

[32] Aspectul este ilustrat şi în Figura 3-1

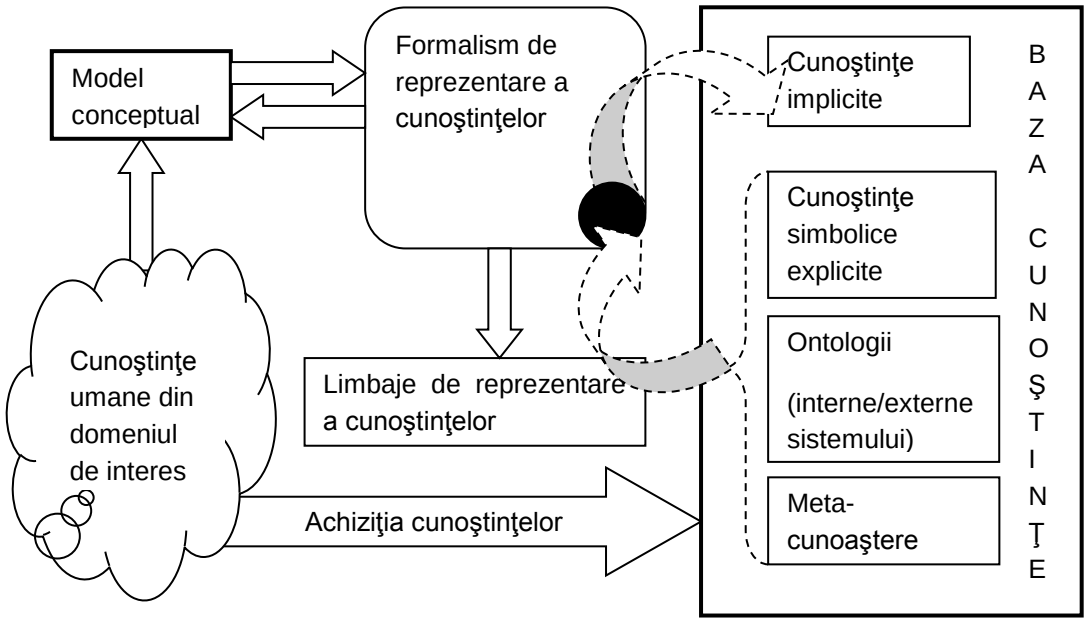


Figura 3-1 Procesul de reprezentare a cunoştinţelor

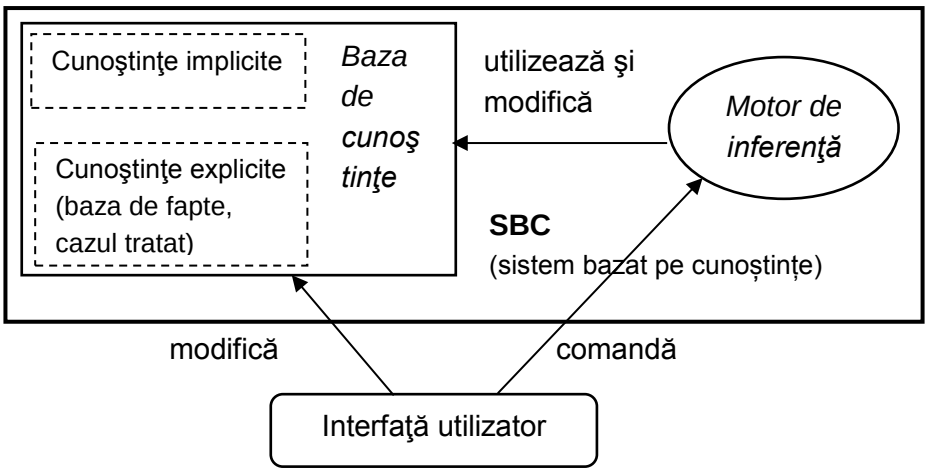


Figura 3-2 Arhitectura generală a unui SBC pentru rezolvarea de probleme

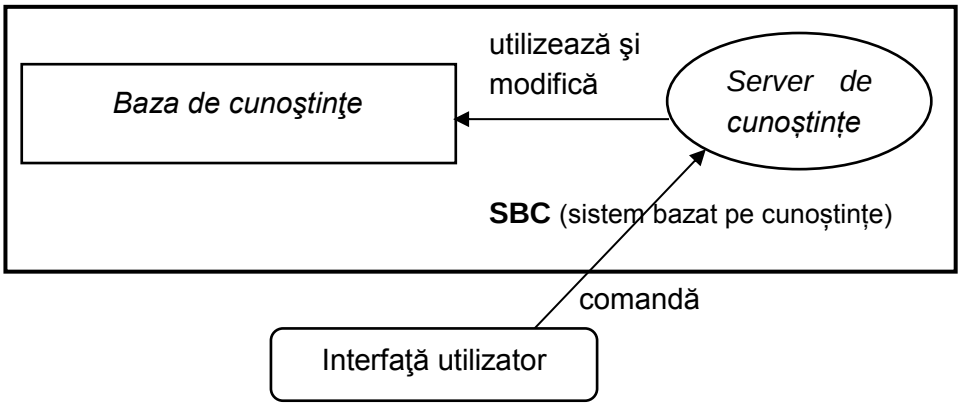


Figura 3-3 Arhitectura generală a unui SBC pentru gestiunea cunoştinţelor

Aspectele prezentate anterior și concluziile studiului comparativ asupra instrumentelor existente la ora actuală (tabelul 2-3) în ingineria ontologică au condus la **obiectivul acestui capitol al lucrării**: **propunerea şi validarea teoretică şi experimentală a unei o metode şi a unui instrument de creare şi operaţionalizare a unei ontologii a domeniului, puternic formalizate, care să poată fi utilizată în sistemele bazate pe cunoștințe într-un mod cât mai independent de scopul operațional al acestor sisteme.**

Utilizarea ontologiei și operaționalizarea acesteia în cadrul unui SBC vizează utilizarea operațională a unei reprezentări a cunoștințelor conceptuale ***neutre față de aplicația care o va utiliza***.

Vom propune o ***metodă generală de operaționalizare a ontologiei domeniului***, bazată pe specificarea unui ***scenariu de utilizare operațional***. Metoda propusă permite automatizarea procesului de operaționalizare independent de limbajul folosit pentru reprezentarea cunoștințelor. Cu toate acestea, pentru operaționalizare a fost ales formalismul grafurilor conceptuale.

În raport cu natura ontologiilor - Obiectivul vizează modul de construire a unei ontologii "grele" (puternic formalizate) care să sprijine raționamentul, indiferent de forma acestuia (interogarea unei baze de cunoștințe, inferență asupra unor fapte sau testarea coerenței bazei de cunoștințe) și de axiomele care exprimă semantica domeniului.

În raport cu ciclul de viață al ontologiilor – Modul în care va fi operaționalizată reprezentarea conceptuală a cunoștințelor determină limbajul în care va fi reprezentată ontologia (ontologia este dependentă de limbaj (pag. 55), iar așa cum am evidențiat în subcapitolul 2.1, limbajele de reprezentare a ontologiilor au expresivități și capacități inferențiale diferite). Alegerea limbajului de exprimare a ontologiei îngrădește posibilitatea utilizării acesteia în orice SBC. În cadrul celorlalte instrumente de inginerie ontologică, ontologiile bazate pe alte proprietăți conceptuale în afară de subsumare necesită specificarea axiomelor direct la nivel operațional, sub o formă fixă.

În raport cu reprezentarea cunoștințelor – Construirea ontologiilor necesită intervenția experților domeniului, care nu sunt nici experți în reprezentarea cunoștințelor, nici programatori. În *Anexa I* au fost prezentate cele trei modele de reprezentare a cunoștințelor. Dintre acestea, modelul grafurilor conceptuale este singurul care oferă o sintaxă grafică. În această lucrare vom folosi sistemul formal al grafurilor conceptuale nu doar ca notație grafică, ci ca model operațional de reprezentare, care oferă mecanisme de raționament (raționament bazat pe proiecție).

În raport cu modelul grafurilor conceptuale - Operația fundamentală a modelului CG, proiecția, poate fi ușor înțeleasă de către un expert în domeniu, nespecialist în informatică și/sau în limbajul de reprezentare și îi permite acestuia să urmărească, pas cu pas, raționamentele realizate, cu scopul de a valida reprezentarea.

3.2 CONTRIBUȚII TEORETICE PRIVIND LIMBAJUL DE REPREZENTARE OntL_CG

Având în vedere faptul că o ontologie trebuie să includă cel puțin un vocabular al termenilor, în funcție de gradul formalizării, aceasta poate lua o varietate de forme: **o ontologie grea**, puternic formalizată, (figura 3-4) trebuie poată exprima nu doar cunoştinţele statice din domeniu, dar şi proprietăţile acestora, constrângerile, axiomele şi regulile utilizate în mecanismele de inferenţă. Ontologiile "grele" conţin termenii utilizaţi pentru conceptele şi relaţiile domeniului, proprietăţile acestora şi pentru orice alte cunoştinţe necesare descrierii semantice a domeniului (specificarea semnificaţiei termenilor prin definiţii, modul de relaţionare a conceptelor care determină diferite structuri, constrângeri asupra interpretărilor posibile)[33].

Numeroși cercetători au subliniat faptul că semantica diferențială dintr-o ontologie este aproape "goală" dacă nu există posibilitatea de specificare a axiomelor: Pe de o parte, reprezentarea şi raţionamentul asupra conceptelor, taxonomiilor şi a relaţiilor binare nu sunt, de obicei, suficiente pentru crearea *ontologiilor grele* şi realizarea unor raţionamente complexe; de cealaltă parte, translatările existente între diferitele limbaje de reprezentare a ontologiilor nu sunt destul de precise, conducând adesea la pierderea de informaţii [173].

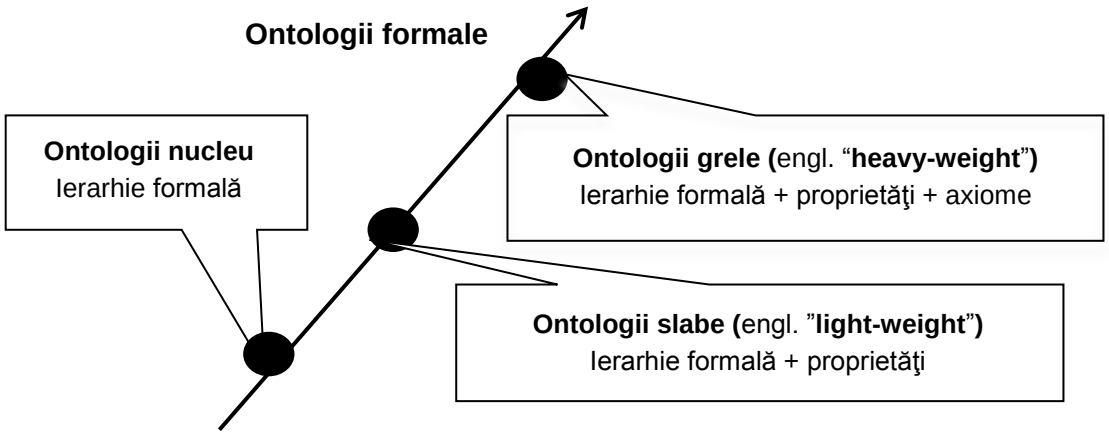


Figura 3-4 Clasificarea ontologiilor formale după gradul de formalizare

Obiectivul vizat în această parte a lucrării constă în **propunerea unui limbaj de reprezentare a cunoașterii ontologice bazat pe paradigma Entitate/Relație** (introdusă de Chen [174]), asemănător celui pe care se bazează logicile de descriere, care să permită o

[33] *Ontologiile nucleu* - simple taxonomii, ierarhii ale conceptelor din domeniul modelat

Ontologiile slabe (constând în termeni de descriere a conceptelor, a relaţiilor domeniului şi a unor proprietăţi semantice minimale) pot fi suficiente pentru dezvoltarea unor aplicaţii (cum ar fi aplicaţiile de căutare) care folosesc în raţionament doar relaţia de subsumare.

reprezentare cât mai largă a cunoștințelor unui domeniu, care să ofere posibilități computaționale și să fie ușor de înțeles și folosit de către un utilizator care nu are cunoștințe de informatică. Limbajul va permite reprezentarea cunoștințelor și proprietăților care apar într-o ontologie și va putea fi translatat într-un formalism operațional într-un mod atractiv și intuitiv.

3.2.1 OntL_CG – BAZAT PE PARADIGMA ENTITATE-RELAȚIE

Paradigme Entitate-Relație unifică modelul rețelelor, modelul relațional și teoria mulțimii entităților. Paradigma se bazează pe (1) teoria mulțimilor, (2) algebra modernă, (3) logică și (5) teoria laticilor. În paradigma Entitate-Relație, conceptele și relațiile sunt utilizate pentru construirea grafurilor (în care conceptele sunt noduri, iar relațiile sunt arce). Paradigma Entitate-Relație descrie cunoașterea în termeni de concepte și relații dintre ele, în timp ce în modelul cadrelor relațiile apar ca proprietăți ale conceptelor. Spre deosebire de modelul relațional al datelor, în care o relație reprezintă produsul cartezian al unor valori (ale atributelor entităților participante în relație), în modelul Entitate-Relație, relația reprezintă produsul cartezian al entităților.

Limbajul propus, numit **OntL_CG (Ont**ological **L**anguage based on **C**onceptual **G**raphs), se bazează pe modelul grafurilor conceptuale și extensiile acestuia (SCG cu reguli și constrângeri), propuse de către Chein și Mugnier (prezentate formal în Anexa III).

3.2.2 LIMBAJUL OntL_CG

Limbajul de reprezentare a cunoștințelor, numit OntL_CG, propus pentru construirea unei ontologii puternic formalizate a domeniului, constituie o parte a modelului grafurilor conceptuale, ***restrânsă la nivelul ontologic de reprezentare a cunoștințelor***. În plus, modelul nostru integrează proprietățile asupra tipurilor de concepte și a tipurilor de relații (folosite în ontologii), care nu sunt explicit integrate în modelul grafurilor conceptuale (CG). Limbajul propus este diferit de cel al CG, dar compatibil cu acesta, având la bază aceeași paradigmă. **Limbajul va permite crearea și validarea unei ontologii a domeniului puternic formalizate, care va putea fi utilizată apoi într-un sistem bazat pe cunoștințe.**

Sintaxa grafică a limbajului OntL_CG este cea din grafurile conceptuale, care permite construirea unei ontologii puternic formalizate într-un mod ușor și intuitiv. Modelul grafurilor conceptuale constituie limbajul operațional cel mai potrivit pentru operaționalizarea unei ontologii (**operaționalizarea axiomelor din ontologie**) exprimate în OntL_CG, dar, evident, este posibilă și operaționalizarea într-un alt limbaj.

NIVELUL TERMINOLOGIC

Nivelul terminologic permite specificarea vocabularului conceptual al domeniului prin concepte, relaţii şi instanţe ontologice. Atât conceptele, cât relațiile pot fi organizate în ierarhii (structurate prin sub-sumare). Menționăm că folosim noțiunile de concept/relație adoptate în ingineria ontologică, similare cu cele de tip de concept/tip de relație din formalismul CG. Comparativ cu modelul CG, este similar suportului.

Definiția următoare este unanim acceptată:

Definiția 3-1 La nivel terminologic, ontologia domeniului este un vocabular, $\mathcal{V}$, definit prin tuplul ($\mathcal{C}$, $\mathcal{R}$, $\mathcal{I}$), în care:

a) $\mathcal{C}$ este mulțimea conceptelor, care include conceptul Universal (T).

b) $\mathcal{R}$ este mulțimea relațiilor, care include relația de diferență (diff).

c) $\mathcal{I}$ este mulțimea instanțelor ontologice.

Observații:

1. Elementele mulțimilor $\mathcal{C}$ și $\mathcal{R}$ se determină în faza de conceptualizare și corespund claselor obiectelor din domeniul modelat și relațiilor dintre acestea.
2. Instanțele ontologice apar doar în cazuri rare, fiind absolut necesare în exprimările axiomelor domeniului (de exemplu, într-o ontologie a trigonometriei, numărul π (pi)).
3. Instanțele (numite și indivizi sau individuali) constituie definiţii extensionale ale ontologiei, obiecte care particularizează cunoştinţele din ontologie.
4. Relația diff ([175]) (figura 3-13):
   - Este necesară pentru a exprima că două concepte generice sunt diferite.
   - Are semnătura (T, T) și este simetrică.
   - Prezența relației determină adăugarea constrângerii că două noduri concept co-identice nu pot fi legate prin relația diff.

NIVELUL AXIOMATIC

Axiomele reprezintă cunoștințele neterminologice, intensiuni[34] ale conceptelor și ale relațiilor (mulțimi de atribute, proprietăți și caracteristici), fiind elementele care deosebesc o ontologie puternic formalizată de un simplu vocabular al termenilor. Ele exprimă modul în care sunt folosite cunoștințele terminologice ale domeniului, deci semantica domeniului.

Propunem două categorii de axiome:

1. Axiome-schemă;
2. Axiome ale domeniului.

Definiția 3-2 Numim *axiomă-schemă* o axiomă prototip care caracterizează proprietăţile clasice ale tipurilor de concepte şi ale tipurilor de relaţii.

Definiția 3-3 Numim *axiomă a domeniului* o axiomă care exprimă proprietăți specifice domeniului modelat.

Operaționalizarea axiomelor se va realiza cu ajutorul modelului CG. Poate fi ales și un alt limbaj operațional, însă alegerea acestui model a fost determinată de avantajul ***reprezentării grafice a axiomelor***. Axiomele *vor fi reprezentate sub formă de reguli, însă în locul unui cuplu de λ-*

[34] **Intensiunea unui concept/relație** reprezintă mulțimea atributelor, caracteristicilor și proprietăților conceptului/relației. **Extensiunea unui concept/relație** reprezintă mulțimea instanțelor conceptului/relației. Bachimont consideră că intensiunea furnizează semantica diferențială, iar extensiunea – semantica referențială. Folosește termenii de context semantic pentru intensiune și context formal pentru extensiune.

abstracțiuni care exprimă graful ipotezei, respectiv al concluziei regulii, vom folosi reprezentarea grafică, sub forma bicoloră (detalii în Anexa III, definițiile AIII-25, 26 și 27).

A. Axiomele-schemă

Axiomele-schemă au o formă predefinită, care va fi instanțiată cu una sau două primitive conceptuale. De exemplu, proprietatea de simetrie a unei relații binare este exprimată în formalismul CG sub forma axiomei din figura 3-10. Instanțierea acestei proprietăți presupune înlocuirea în axioma-schemă a relației - din domeniul de aplicație - care are proprietatea de simetrie și a conceptelor care participă în relație.

Se propun axiome-schemă pentru:

1. Concepte;
2. Relații;
3. Instanțe.

1) Axiome-schemă pentru concepte

- *Subsumarea conceptelor* (legăturile IS-A între două concepte, folosite pentru structurarea taxonomiilor de concepte în arbori sau în latici): Conceptul c_2 este subconcept al conceptului c_1 dacă orice instanță a lui c_2 este și instanță a lui c_1.
- *Abstractizarea unui concept*: Conceptul c este abstract dacă orice instanță a lui c este, deasemenea, instanță a unuia dintre subconceptele sale. Precizăm că în propunerea noastră, termenul de concept abstract nu este folosit în sensul de concept fără extensiune, ci de concept care poate fi adăugat la ontologie pentru a ajuta structurarea reprezentării.
- *Disjuncția între două concepte*: Conceptele c_1 și c_2 sunt disjuncte dacă nu au instanțe comune.

2) Axiome-schemă pentru relații

- *Subsumarea relațiilor* (legăturile IS-A între două relații): Relația r_2 este subrelație a lui r_1 dacă orice instanță a a relației r_2 este și instanță a relației r_1.
- *Semnătura unei relații:* Precizează conceptele cele mai specifice pe care le poate lega. Formal, semnătura unei relații r este imaginea acesteia prin aplicarea funcției $\sigma: \mathcal{R} \rightarrow C^n$, unde n este aritatea relației. Semnătura unei relații trebuie să fie conformă cu semnătura relației părinte (aceeași aritate, conceptele legate prin r trebuie să fie subconcepte ale conceptelor legate prin relația părinte).
- *Incompatibilitatea a două relații cu aceeași semnătură:* Două relații cu aceeași semnătură sunt incompatibile dacă ele nu pot lega aceleaşi instanţe ale conceptelor. Două relații cu semnături diferite sunt incomparabile.
- *Exclusivitatea a două relații cu aceeași semnătură:* Două relații cu aceeași semnătură sunt incompatibile dacă ele nu pot lega aceeași mulțime de instanțe. Două relații cu semnături diferite sunt incompatibile.
- *Proprietățile algebrice ale relațiilor binare (pentru relația R: C → CxC)*
  - Simetria: ∀x∈C, (x, x)∈R
  - Tranzitivitatea: ∀x, y, z ∈C, dacă (x, y)∈R și (y, z)∈R, atunci (x, z)∈R
  - Reflexivitatea: ∀x, y ∈C, dacă (x, y)∈R, atunci (x, y)∈R

- Anti-reflexivitatea: O relație nu poate fi în același timp anti-reflexivă și reflexivă
- Anti-simetria: ∀x, y ∈C, dacă (x, y)∈R și (y, x)∈R, atunci x=y
- Celelalte proprietăți algebrice ale relațiilor pot fi exprimate prin combinații ale celor anterioare:
  - O relație asimetrică este o relație anti-simetrică și anti-reflexivă.
  - O relație de echivalență este o relație reflexivă, simetrică și tranzitivă.
  - O relație de ordine parțială este o relație reflexivă, anti-simetrică și tranzitivă.

3) Axiome-schemă pentru instanțe

Tipul unei instanțe este tipul cel mai specific căreia aceasta îi aparține.

B. Axiomele domeniului

Sunt exprimate grafic, bicolor, în funcție de domeniul modelat și se azează tot pe λ-abstracțiuni.

3.2.3 SEMANTICA FORMALĂ A LIMBAJULUI OntL_CG

Semantica formală a limbajului OntL_CG este cea din teoria mulțimilor, care *permite constrângerea interpretării axiomelor-schemă și a axiomelor domeniului* pentru ontologia domeniului, dar *nu impune un mod de utilizare operațională a acestora*.

Definiția 3-4 Interpretarea δ a ontologiei $O = (\mathcal{C}, \mathcal{R}, \mathcal{I})$ pe domeniul $\mathcal{D}$ este:

- ∀ c ∈ $\mathcal{C}$, δ(c) ⊆ $\mathcal{D}$. Se consideră că δ(T) = $\mathcal{D}$
- ∀ r ∈ $\mathcal{R}$, δ(r) ⊆ $\mathcal{D}^n$, unde n este aritatea relației
- ∀ i ∈ $\mathcal{I}$, δ(r) ∈ $\mathcal{D}$

Definiția 3-5 Semantica formală a axiomelor-schemă:

- Sub-sumarea a două concepte (c_2 este subconcept al lui c_1) are ca semantică formală $\delta(c_2) \subseteq \delta(c_1)$
- Abstractizarea unui concept c: $\left(\bigcup_{i=1..n} \delta(c_i)\right) = \delta(c)$, unde c_i sunt concepte-fii ale lui c
- Disjuncția conceptelor c_1 și c_2: $\delta(c_2) \cap \delta(c_1) = \emptyset$
- Disjuncția a două relații (r_2 este sub-relație a lui r_1): $\delta(r_2) \subseteq \delta(r_1)$
- Semnătura unei relații r: $\sigma(r) = (c_1, \dots, c_n) \Longrightarrow \delta(r) \subseteq \delta(c_1) \boldsymbol{x} \dots \boldsymbol{x} \delta(c_n)$. Semnătura relației *diff* este $\sigma(diff) = (Universal, Universal)$
- Incompatibilitatea între relațiile r_1 și r_2, cu aceeași semnătură: $\delta(r_1) \cap \delta(r_2) = \emptyset$
- Exclusivitatea între relațiile r_1 și r_2, cu aceeași semnătură: $\sigma(r_1) = \sigma(r_2) = (c_1, .., c_n)$, are ca semantică formală: $\forall(i_1, .., i_n) \in \delta(c_1) \times \dots \times \delta(c_n), (i_1, .., i_n) \in \delta(r_1) \Leftrightarrow (i_1, .., i_n) \notin \delta(r_2)$
- Simetria unei relații binare are semantica formală: $\forall(i_1, i_2) \in \delta(r) \Longrightarrow (i_2, i_1) \in \delta(r)$
- Tranzitivitatea unei relații binare: $\forall (i_1, i_2) \in \delta(r), (i_2, i_3) \in \delta(r) \Longrightarrow (i_1, i_3) \in \delta(r)$
- Reflexivitatea unei relații binare cu semnătura $\sigma(r) = (c, c)$, are semantica formală: $\forall\ i \in \delta(c) \Longrightarrow (i, i) \in \delta(r)$
- Anti-Reflexivitatea unei relații binare: $\forall\ i \in \mathcal{D} \Longrightarrow (i, i) \notin \delta(r)$

- Antisimetria-unei relații binare r are semantica formală: $\forall (\mathrm{i}_1, \mathrm{i}_2) \in \delta(\mathrm{r}) \Longrightarrow (\mathrm{i}_2, \mathrm{i}_1) \notin \delta(\mathrm{r})$
- Cardinalitatea maximă, de valoare c_{max}, asupra conceptului de pe poziția i din semnătura $\sigma(\mathrm{r}) = (\mathrm{c}_1, .., \mathrm{c}_\mathrm{n})$ a relației r are ca semantică formală: $\forall\, i_i \in \delta(\, c_i)$ există cel mult c_{max} n tuple diferite $(i_1, .., i_{1-1}, i_{1+1}, .., i_n) \in \delta(\mathrm{c}_1) .. \times \delta(\mathrm{c}_{\mathrm{i}-1}) \times \delta(\mathrm{c}_{\mathrm{i}+1}) \times .. \times \delta(\mathrm{c}_\mathrm{n})$ astfel încât $(\mathrm{i}_1, .., \mathrm{i}_\mathrm{n}) \in \delta(\mathrm{r})$.
- Cardinalitatea minimă, de valoare c_{min}, asupra conceptului de pe poziția i din semnătura $\sigma(\mathrm{r}) = (\mathrm{c}_1, .., \mathrm{c}_\mathrm{n})$ a relației r are ca semantică formală: $\forall\, i_i \in \delta(\, c_i)$ există cel mult c_{max} n tuple diferite $(i_1, .., i_{1-1}, i_{1+1}, .., i_n) \in \delta(\mathrm{c}_1) .. \times \delta(\mathrm{c}_{\mathrm{i}-1}) \times \delta(\mathrm{c}_{\mathrm{i}+1}) \times .. \times \delta(\mathrm{c}_\mathrm{n})$ astfel încât $(\mathrm{i}_1, .., \mathrm{i}_\mathrm{n}) \in \delta(\mathrm{r})$.
- Tipul c al unei instanțe i are semantica formală: $\delta(\mathrm{i}) \in \delta(\mathrm{c})$.

Semantica formală a axiomelor este cea a regulilor CG (exprimată în limbaj natural: *Dacă ipoteza axiomei este adevărată, atunci concluzia este adevărată*). Reprezentarea axiomei în ontologie nu precizează dacă ea va fi folosită într-un raționament cu înlănțuire înainte sau înapoi.

Semantica formală a axiomelor domeniului este definită ca un *cuplu de λ-abstracțiuni*, reprezentând ipoteza, respectiv concluzia regulii (Anexa III, secțiunea 4).

3.3 CONTRIBUȚII TEORETICE PRIVIND ONTOLOGIILE LIMBAJULUI DE REPREZENTARE OntL_CG

În acest capitol propunem o ontologie de reprezentare (în sensul celei din subcapitolul 1.4.3, din figura 1-12 Clasificarea ontologiilor în funcţie de diferite criterii și din figura 1-13 Diferite tipuri de ontologii) *bazată pe limbajul de reprezentare OntL_CG* (propus în 3.2). Este o ontologie de meta-nivel, numită **OntL_CG– MetaOnto**.

Definiția 3-6

Ontologia OntL_CG-MetaOnto (care descrie elementele folosite în reprezentare) este definită formal ca tuplul $(\mathcal{H}_{\mathcal{CR}}, \mathcal{H}_{\mathcal{RR}}, \mathcal{A}_s, \mathcal{A}_{\mathcal{D}})$, în care:

- $\mathcal{H}_{\mathcal{CR}}$ - Ierarhia conceptelor care descriu primitivele utilizate în limbajul OntL_CG (figura 3-5);
- $\mathcal{H}_{\mathcal{RR}}$ – Ierarhia relațiilor pentru reprezentarea legăturilor între primitivele limbajului OntL_CG (figura 3-6);
- $\mathcal{A}_s$ - Axiomele-schemă folosite în principal pentru descrierea proprietăților conceptelor și a relațiilor;
- $\mathcal{A}_{\mathcal{D}}$ - Axiomele domeniului.

Primitivele limbajului OntL_CG sunt concepte, relații și proprietăți (figura 3-5). Proprietățile pot fi cele legate de un concept (abstractizare pentru conceptele abstract) sau de proprietățile algebrice ale relațiilor (simetrie, tranzitivitate, etc.). Conceptele și relațiile se pot afla în consecventul sau antecedentul unei reguli.

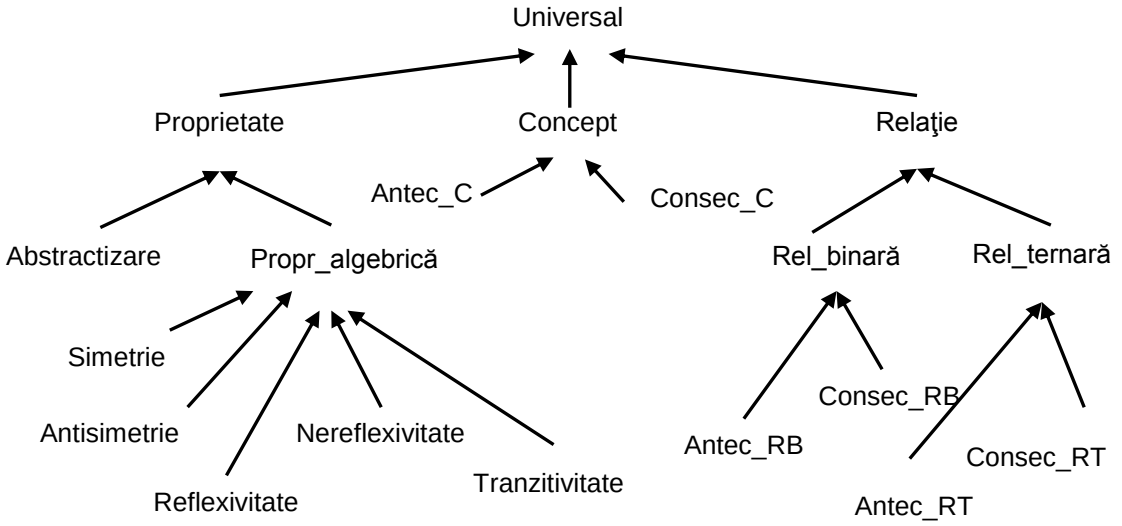


Figura 3-5 Conceptele din ontologia de reprezentare OntL_CG-MetaOnto

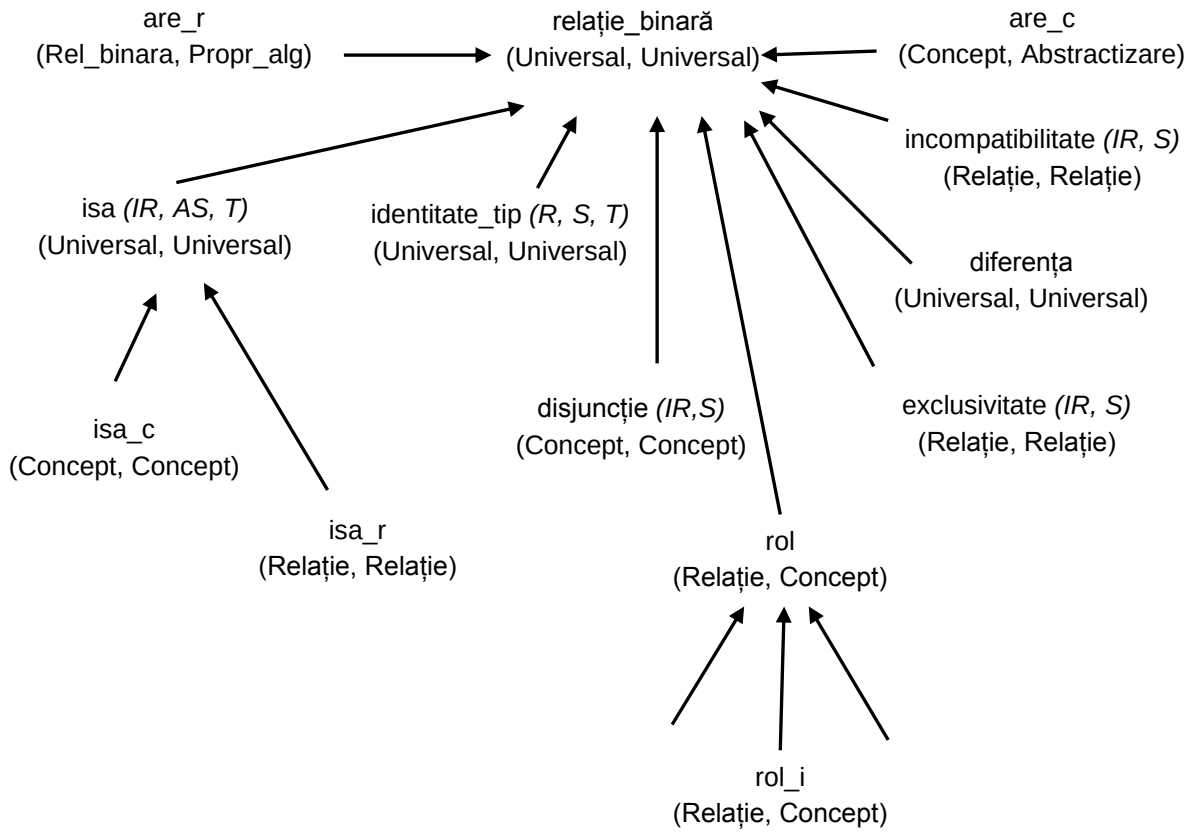
Observație:

Chiar dacă în figura 3-5 au fost specificate doar relațiile binare și ternare, în mod similar, pot fi utilizate relații de orice aritate.

Relațiile de reprezentare a legăturilor dintre primitivele limbajului (figura 3-6) sunt cele care *exprimă proprietățile algebrice ale unei relații* (are_r), *relația de subsumare* (isa) între două concepte (isa_c) sau între două relații de aceeași aritate (isa_r), *identitatea tipului* de concepte sau de relații (identitate_tip), *exclusivitatea și incompatibilitatea* a două relații (exclusivitate, incompatibilitate), *disjuncția* a două concepte (disjuncție), semnătura și cardinalitatea unei relații și sunt exprimate prin axiome-schemă.

O instanță OntL_CG-MetaOnto (practic, un graf) poate reprezenta atât ontologia domeniului, cât și grafuri-fapte. Graful MetaOnto reprezină ontologia care conține o reprezentare a ierarhiei de concepte, reprezentarea ierarhiei de relații și reprezentarea axiomelor.

Implementarea ontologiei **OntL_CG-MetaOnto** este realizată în **sistemul OntL_CG**, propus în capitolul 3.6. În Anexa II sunt prezentate detalii de implementare (figura AII - 3, figura AII - 7, figura AII - 9, figura AII - 10, figura AII - 14).



Legendă: **T** – tranzitivitate **AS** - antisimetrie

R – reflexivitate **IR** - nereflexivitate **S** – simetrie

Figura 3-6 Relațiile din ontologia de reprezentare OntL_CG-MetaOnto

3.4 CONTRIBUȚII TEORETICE PRIVIND O METODĂ DE OPERAȚIONALIZARE A UNEI ONTOLOGII A DOMENIULUI

Operaționalizarea unei ontologii constă într-un sistem bazat pe cunoștințe în rezolvarea de probleme constă în (figura 3-7):

1. Specificarea semanticii operaționale adăugate ontologiei, care descrie mecanismele de raționament din baza de cunoștințe care utilizează ontologia;
2. Cum am evidențiat și în metoda MIIO (subcapitolul 2.3):
    a. Asamblarea unor componente informatice predefinite, corespunzătoare task-urilor identificate în model;

b. Transcrierea reprezentării conceptuale a ontologiei într-un limbaj operațional, exprimând cunoștințele expertului în sintaxa limbajului de operaționalizare[35].

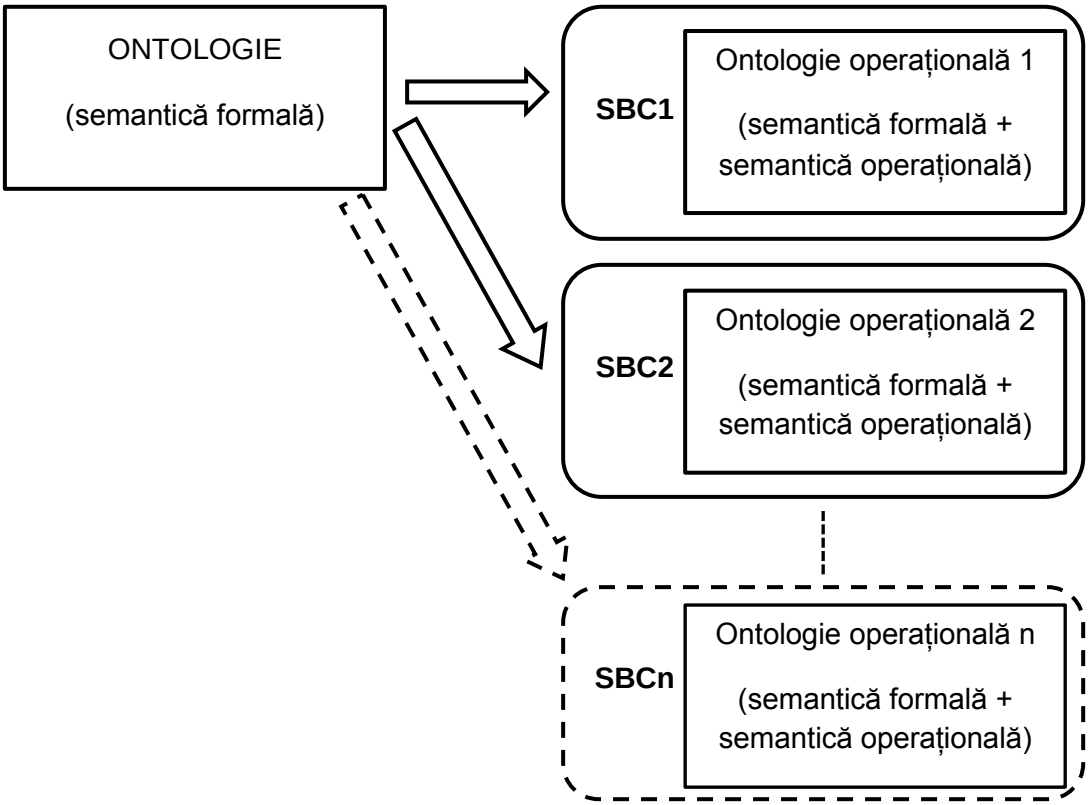


Figura 3-7 Schema generală de operaționalizare a unei ontologii în vederea utilizării acesteia în cadrul unui SBC

Scopul nostru este acela de a oferi utilizatorului un mod de specificare intuitiv și simplu a scopurilor și modului în care cunoștințele din ontologie vor fi folosite într-o aplicație, deci a scenariului de utilizare a ontologiei (în metoda MIIO (subcapitolul 2.3), faza a II-a, etapa I).

Observații:

1. Cunoștințele terminologice (cele din ontologie, sau suport dacă folosim terminologia din CG) rămân aceleași indiferent de scopul sau modul în care vor fi utilizate în baza de cunoștințe: reprezentarea unui tip de concept sau unui tip de relație este aceeași și în cazul în care baza de cunoștințe este validată, și în cazul în care, plecând de la faptele existente, se obțin noi fapte care sunt adăugate la baza de cunoștințe sau în cazul unor interogări. Ca urmare, reprezentarea cunoștințelor terminologice nu este influențată de semantica operațională a acestora.
2. **Doar reprezentările operaționale ale axiomelor sunt determinate de scopul aplicației.**

[35] În programarea logică, procesul de operaționalizare a unei reprezentări a cunoștințelor pentru un anumit obiectiv operațional este numit “compilarea cunoștințelor” (sau “compilarea regulilor”). Constă în transcrierea unei mulțimi de formule logice într-un limbaj formal.

Ca urmare, în sensul metodei de operaționalizare propuse în acest capitol, definim noțiunile de scenariu de utilizare și context de utilizare. Schema detaliată de operaționalizare propusă este reprezentată în figura 3-8.

Definiția 3-7*:* Numim *scenariu de utilizare* descrierea modului în care axiomele (axiomele–schemă și axiomele domeniului) vor fi utilizate în raționamentul din sistemul bazat pe cunoștințe.

Definiția 3-8: Numim context de utilizare rolul pe care o axioma îl va juca în raționamentul din baza de cunoștințe

Definiția 3-9*:* Formal, un *scenariu de utilizare* este $\cup(\mathcal{A}_i, C_j)$ - reuniunea tuplelor în care $\mathcal{A}_i \in \mathcal{A}_s \cup \mathcal{A}_\mathcal{D}$, $i = 1..n$ ($\mathcal{A}_i$ este o axiomă), iar C_j, $j = 1..4$ este contextul de utilizare al axiomei.

Pentru fiecare axiomă va fi specificat contextul de utilizare (cum va fi folosită și aplicată axioma). Astfel, o axiomă poate fi folosită pentru a deduce noi cunoștințe sau pentru a verifica adecvarea cunoștințelor în raport cu semantica domeniului considerat.

Contextele de utilizare *identificate* și *selectate*[36] pentru metoda de operaționalizare propusă sunt: *contextul inferențial și cel de validare*, *contextul implicit și cel explicit*. Prin combinarea[37] acestora, se obțin următoarele contexte de utilizare:

1. Contextul de utilizare inferențial și explicit – utilizatorul aplică axioma asupra bazei de fapte existente, în vederea obținerii de noi fapte;
2. Contextul de utilizare inferențial și implicit – axioma este aplicată de către sistem, asupra bazei de fapte existente, în vederea obținerii de noi fapte;
3. Contextul de validare explicit – utilizatorul aplică axioma pentru a verifica dacă baza de fapte este în concordanță cu semantica domeniului;
4. Contextul de validare implicit – axioma este aplicată de către sistem, pentru verificarea concordanței bazei de fapte cu semantica domeniului;

Figura 3-9 prezintă o posibilă schemă de aplicare a axiomelor. Cele patru etape sunt:

I. Utilizatorul adaugă fapte în baza de cunoștințe
II. Raționament la cererea utilizatorului
III. Etapa de inferențe automate
IV. Etapa de validare semi-automată sau automată în funcție de utilizarea sau neutilizarea unei axiome în contextul de validare explicit.

În *scenariile de validare pură*, cunoștințele ontologice sunt utilizate doar pentru validarea bazei de fapte în raport cu semantica domeniului.

În *scenariile de inferență implicită* cunoștințele ontologice sunt utilizate pentru a produce noi cunoștințe (ca în cazul sistemelor expert) conforme cu semantica domeniului, *fără a fi necesară validarea*.

[36] Aceste contexte de utilizare apar în multe alte sisteme bazate pe ontologii.

[37] Majoritatea sistemelor combină mecanismele inferențiale cu cele de validare.

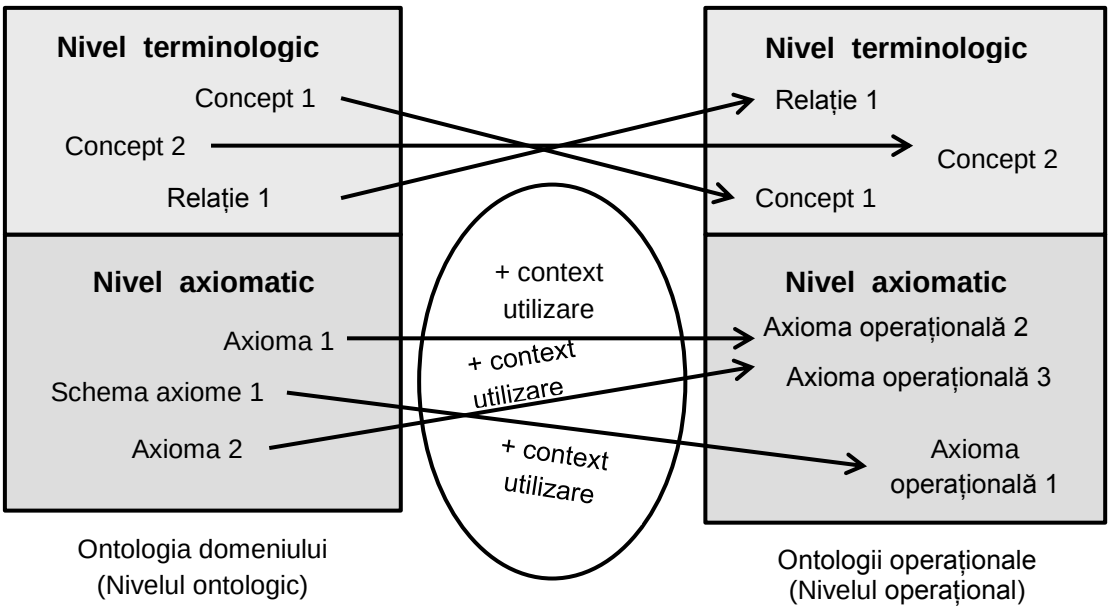


Figura 3-8 Schema detaliată de operaționalizare a unei ontologii în vederea utilizării acesteia în cadrul unui SBC

Limbajul de operaționalizare ales poate fi *oricare*, deoarece toate axiomele (și axiomele-schemă și cele ale domeniului) sunt bazate pe λ-abstracțiuni.

Am propus ca limbajul de operaționalizare să fie cel al grafurilor conceptuale datorită posibilității de exprimare grafică a axiomelor.

Dacă, de exemplu, ca limbaj de operaționalizare ar fi fost ales unul logic, de exemplu logica predicatelor de ordinul I, formele operaționale ale unei axiome, în funcție de contextul de utilizare, ar fi fost:

- *Context inferențial și implicit*: axioma este operaționalizată printr-o regulă corespunzătoare formulei logice $\forall x_1, \dots, x_n H \Rightarrow \exists y_1, \dots, y_m r_1(..) \wedge .. \wedge r_p(..)$;
- *Context inferențial și explicit*: axioma este operaționalizată printr-o regulă corespunzătoare formulei logice $\forall x_1, \dots, x_n H \Rightarrow \exists y_1, \dots, y_m r_1(..) \wedge .. \wedge r_p(..)$ iar p este constrângerea negativă corespunzătoare formulei $\forall x_1, \dots, x_n H\ (\wedge r_i(..))_{i=1..p, i \neq j}$, nu poate exista $r_j'(..), j = 1..p$, unde relațiile r_j' și r_j sunt exclusive. Pentru orice relație din ontologie, exclusivă cu r_j , constrângerea corespunzătoare este înlocuită cu q constrângeri negative care corespund formulei $\forall x_1, \dots, x_n H\ (\wedge r_i(..))_{i=1..p, i \neq j}$, nu poate exista $r_{jk}'(..), k = 1..q$, unde toate r_{jk}' sunt incompatibile cu r_j .
- *Context de validare (implicit, respectiv explicit)*: axioma este operaționalizată prin p constrângeri implicite (respectiv explicite), de forma $\forall x_1, \dots, x_n H\ (\wedge r_i(..))_{i=1..p, i \neq j} \Rightarrow r_j$, $j = 1..p$, unde r_j' și r_j sunt relații exclusive din ontologie. Pentru orice relație din ontologie, exclusivă cu r_j , contrângerea corespunzătoare este înlocuită cu q

constrângeri negative care corespund formulei logice $\forall x_1, \ldots, x_n H\ (\wedge r_i(..))_{i=1..p, i\neq j}$, nu poate exista $r'_{jk}\ (..), k = 1..q$, unde toate r'_{jk} sunt incompatibile cu r_j .

În cazul operaționalizării în limbajul predicatelor de ordinul I (FOL), formalizarea pentru incompatibilitatea și exclusivitatea a două primitive ar fi fost mult mai simplu de exprimat decât în cazul operaționalizării prin limbajul grafurilor conceptuale, deoarece modelul folosit în lucrare nu conține negația. În cazul operaționalizării într-un limbaj FOL:

- Incompatibilitatea între două primitive, P_1 și P_2 este formalizată prin $\neg (P_1 \wedge P_2)$.
- Exclusivitatea între două primitive, P_1 și P_2 este formalizată prin $\neg P_1 \Rightarrow P_2$.

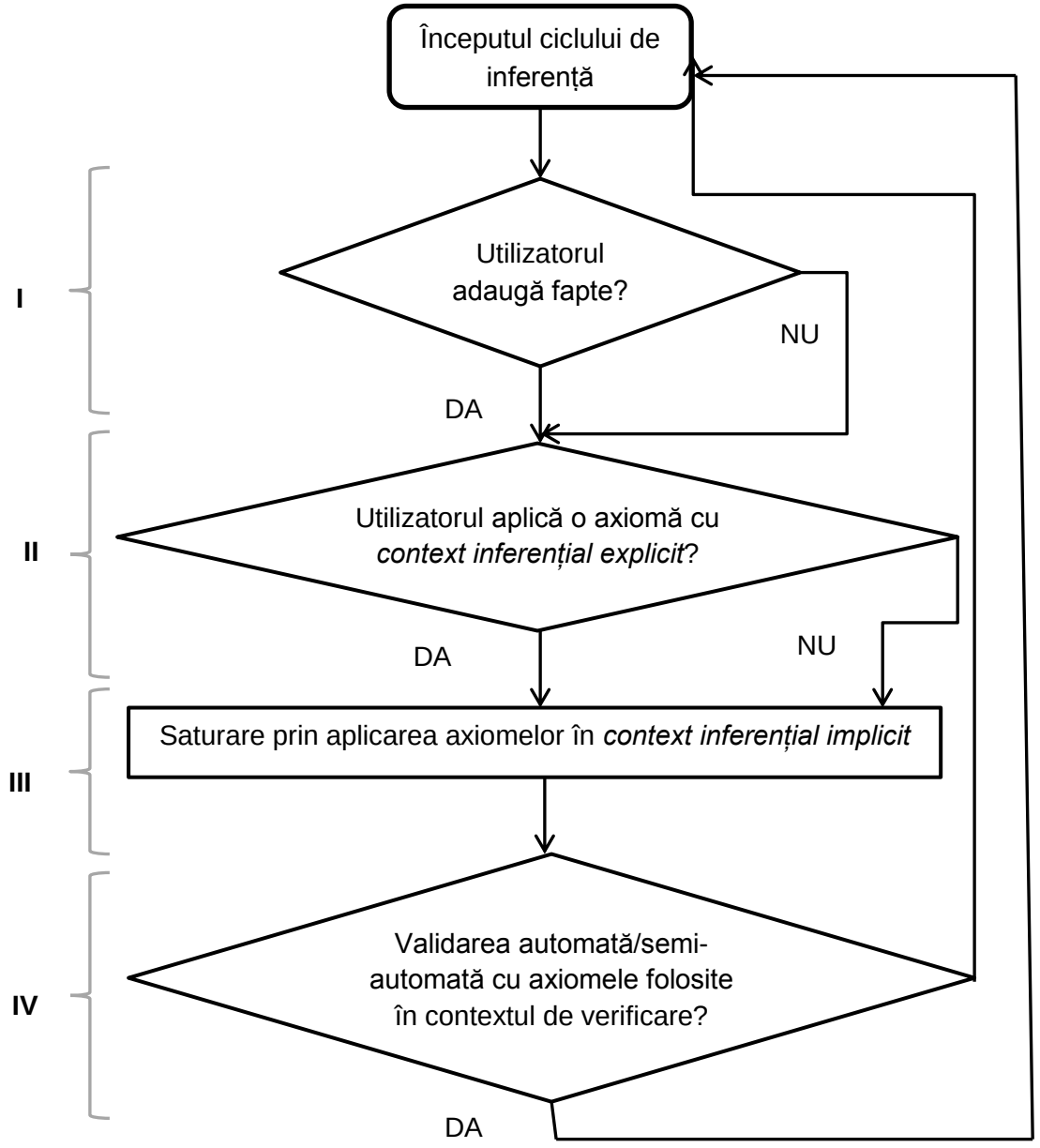


Figura 3-9 Ciclul de inferență pentru ontologia operațională

Testele de validare a ontologiei (validare realizată cu ajutorul constrîngerilor) sunt în mare parte cele propuse de către Gómez-Pérez (prezentate în Anexa I, dar și în subcapitolul 3.6).

3.5 OPERAȚIONALIZAREA BAZATĂ PE MODELUL GRAFURILOR CONCEPTUALE

Limbajul ales pentru operaționalizare este cel al grafurilor conceptuale.

Așa cum am evidențiat în Anexa I, în privința limbajului de reprezentare CG, există două direcții:
1) CG sunt folosite doar ca notație grafică, raționamentul fiind unul logic
2) GC sunt primitive ale limbajului, raționamentul se bazează pe algoritmi ai grafurilor (operații interne, proiecții), au o semantică consistentă și completă și păstrează corectitudinea și completitudinea în raport cu limbajul predicatelor de ordinul I (FOL).

Aceste aspecte sunt prezentate în Anexa I, iar Anexa III prezintă formalismul grafurilor conceptuale folosit în această lucrare.

Limbajul ales prezintă avantajul că semantica operațională a unor proprietăți este determinată prin mecanismul de proiecție a grafurilor: relația de subsumare (a conceptelor sau a relațiilor) și semnătura unei relații de aceeași aritate.

Practic, rămâne doar operaționalizarea axiomelor, iar acest lucru s-a realizat prin **reguli** și **constrângeri** (formalismul utilizat, mai precis SCG extinsă cu reguli și contsrângeri, este prezentat în Anexa III):
- Regulile permit obținerea de noi cunoștințe în mod automat sau la cererea utilizatorului
- Constrângerile permit validarea ontologiei domeniului.

Axiomele din subcapitolul 3.2.2 sunt exprimate sub formă de **reguli și *constrângeri***. Axiomele-schemă au o formă predefinită, fixă, iar axiomele domeniului vor fi exprimate de către utilizator, pe baza sintaxei grafice. Sunt prezentate în figurile următoare (de la figura 3-10 până la figura 3-23) reprezentările grafice ale axiomelor-schemă.

Deoarece modelul folosit nu este cel al grafurilor conceptuale complete (FCG, engl., “Full Conceptual Graphs”), deci **NU include negația**, a fost nevoie să simulăm prezența negației printr-o relație de exclusivitate între două relații. Practic, dacă într-o axiomă este necesară folosirea negației unei relații, pentru relația respectivă se adaugă la ontologie o relație exclusivă celei care trebuie negate.

Proprietățile de incompatibilitate și exclusivitate au fost simulate prin introducerea unei mulțimi de reguli și constrângeri. Pentru incompatibilitatea/exclusivitatea a două relații s-a folosit o *constrângere negativă implicită* care interzice prezența celor două relații între aceleași două concepte (figura 3-23) și o *mulțime de reguli implicite* pentru diferența conceptelor care apar în relație. În figura 3-23 e prezentat cazul incompatibilității a două relații binare, *rel1* și *rel2*.

În cazul general: pentru două relații incompatibile, de aritate n, care leagă mulțimea de n-1 concepte, conceptele trebuie să difere două câte două. Pentru incompatibilitatea regulilor de aritate n din domeniu, va fi prezentă o tot o singură constrângere, dar va crește numărul regulilor (n reguli pentru concepte).

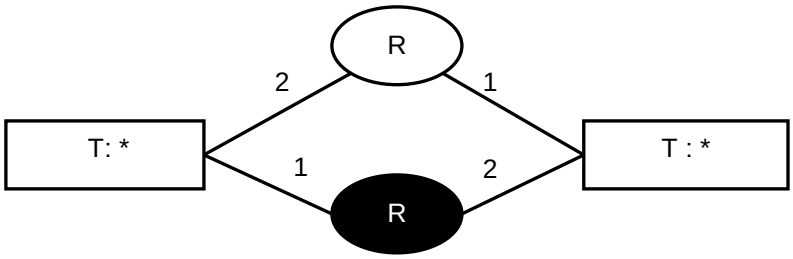


Figura 3-10 Axioma-schemă bazată pe sintaxa CG pentru simetria unei relații binare
(Reprezentarea în sintaxa CG a simetriei relației R)

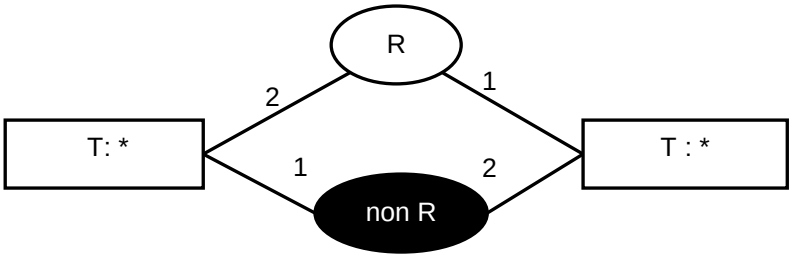


Figura 3-11 Axioma-schemă pentru anti-simetria relației R

Reflexivitatea relației R

Anti-Reflexivitatea relației R
(non-R – relație exclusivă)

Figura 3-12 Reflexivitatea / anti-reflexivitatea relației R

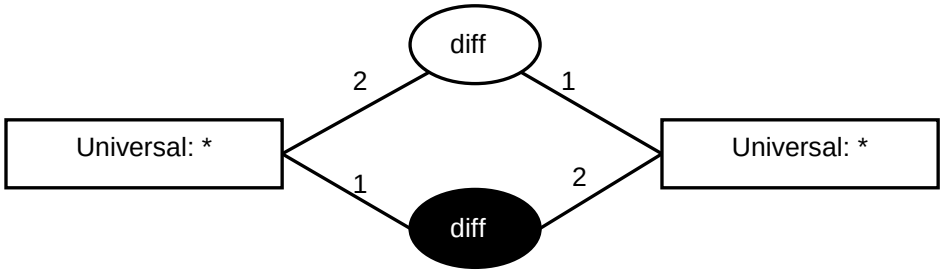


Figura 3-13 Relația diff, pentru exprimarea diferenței a două concepte

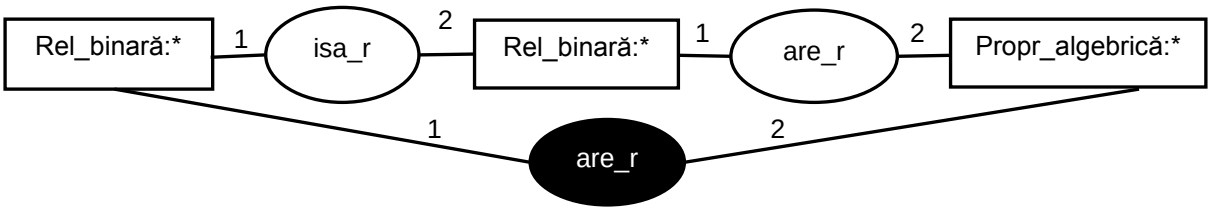


Figura 3-14 Axioma schemă pentru moștenirea proprietăților algebrice

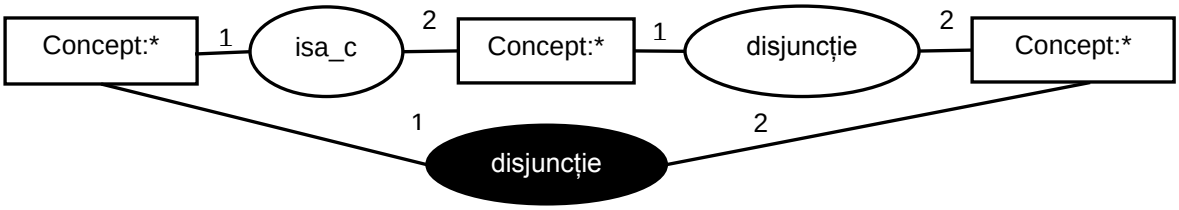


Figura 3-15 Axioma schemă pentru moștenirea disjunctă

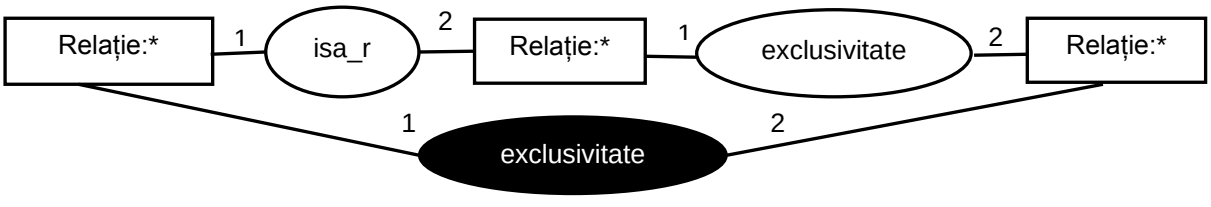


Figura 3-16 Axioma schemă pentru moștenirea exclusivității

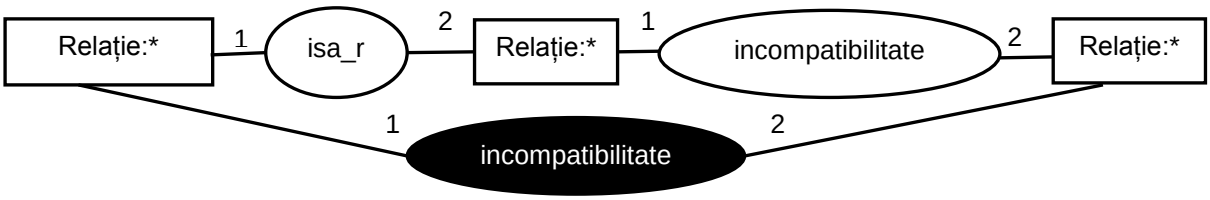


Figura 3-17 Axioma schemă pentru moștenirea incompatibilității

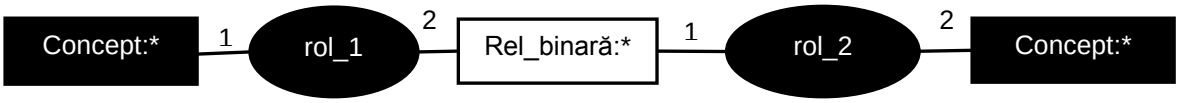


Figura 3-18 Axioma despre semnătura binară a unei relații

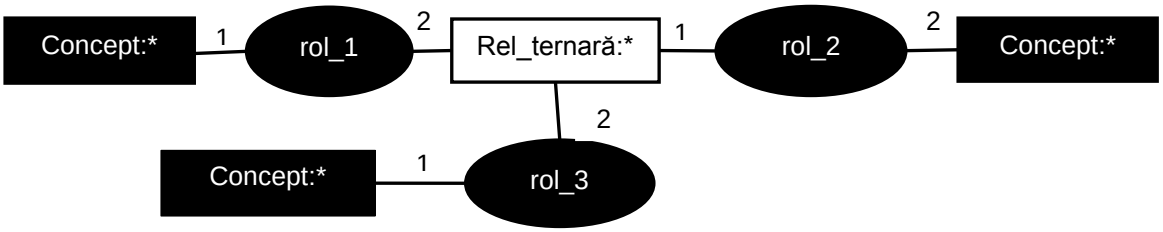


Figura 3-19 Axioma despre semnătura ternară a unei relații

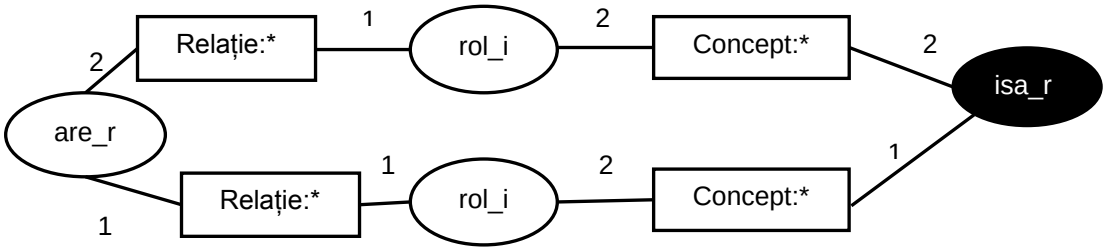


Figura 3-20 Conformitatea semnăturii (în rol_i, i=1, 2 sau 3)

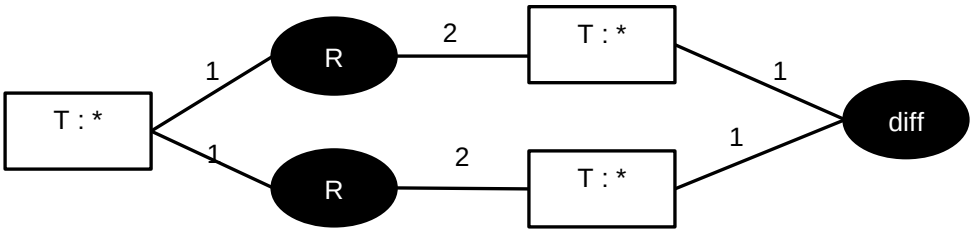


Figura 3-21 Cardinalitatea minimă 2 asupra primului element din semnătura unei relații binare prin cuplu de λ-expresii

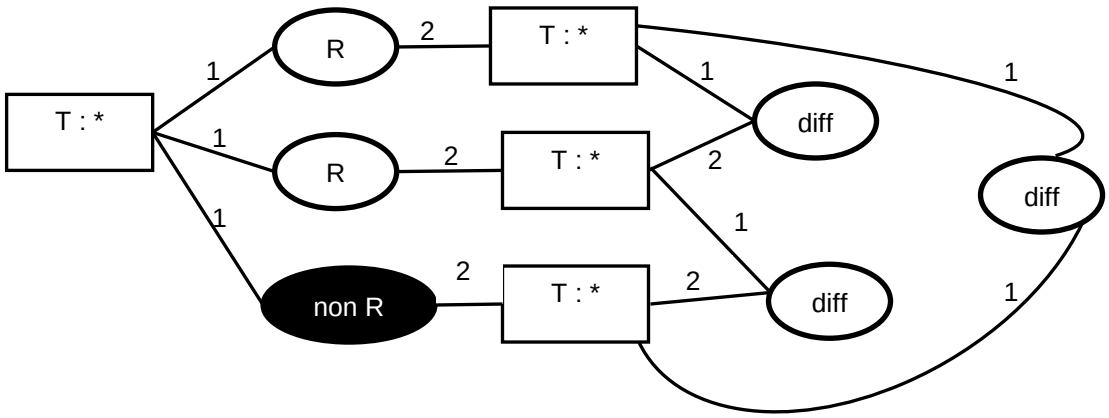


Figura 3-22 Cardinalitatea maximă 2 asupra primului element din semnătura unei relații binare prin cuplu de λ-abstracțiuni

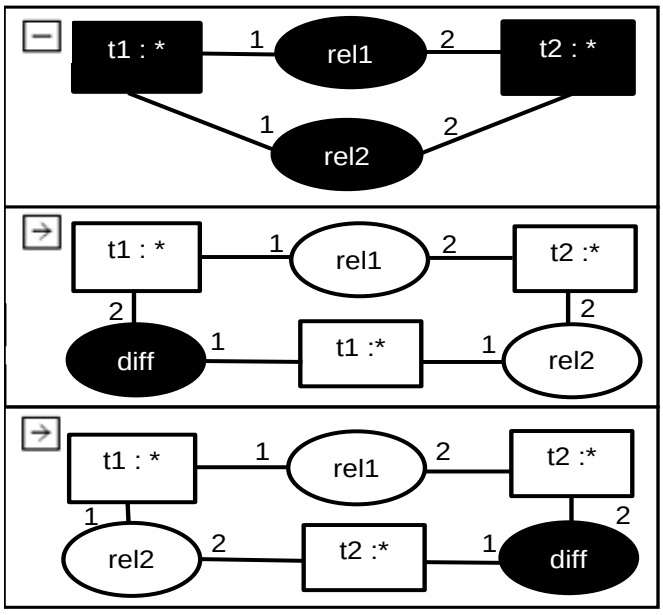


Figura 3-23 Incompatibilitatea între relațiile rel1(t_1, t_2) și rel2(t_1, t_2)

3.6 CONTRIBUȚII APLICATIVE – Sistemul OntL_CG

Aplicația numită OntL_CG (ca și limbajul propus în subcapitolul 3.2.2) permite obținerea unei ontologii operaționale, a unei baze de cunoștințe a domeniului, folosind formalismul grafurilor conceptuale, într-o manieră grafică. Astfel, construcția și testarea ontologiei pot fi realizate nu numai de către utilizatorul care cunoaște sintaxa limbajului în care sunt definite axiomele (de obicei, inginerul de cunoștințe), ci și de către un expert al domeniului modelat (pedagog/profesor autor) care nu deține cunoștințe de informatică. Aplicația translatează automat o reprezentare ontologică a cunoașterii domeniului într-o reprezentare operațională, conform scenariului de utilizare definit de către utilizator, demonstrând modul de aplicare a contribuțiilor teoretice din acest capitol.

Aspectele ***care diferențiază aplicația OntL_CG de alte aplicații existente*** (o sinteză a caracteristicilor instrumentelor existente a fost prezentată în tabelul 2-3 din capitolul 2):

1. Structurarea ontologiei se bazează pe paradigma entitate-relație, spre deosebire de majoritatea instrumentelor existente (OILEd, OntoEdit, Protége), bazate pe paradigma cadrelor.
2. Se bazează pe modelul grafurilor conceptuale, care furnizează atât nivelul terminologic al ontologiei, cât și mecanisme de raționament bazate pe homomorfismul grafurilor.
3. Majoritatea instrumentelor existente furnizează un mod textual de specificare a vocabularului și a axiomelor (în OntoEdit axiomele sunt specificate folosind sintaxa F-logic). Câteva instrumente (WebOnto) pun la dispoziție o interfață grafică care permite doar lucrul asupra vocabularului conceptual, nu și asupra axiomelor.
4. Permite reprezentarea și *operaționalizarea oricărui tip de axiomă*, nu numai a celor predefinite.
5. Posibilitatea de a utiliza ontologia pentru raționament, aspect deosebit de important în web-ul semantic: pentru serviciile web bazate pe ontologii nu sunt suficiente doar ierarhiile de primitive terminologice, reprezentarea axiomelor este o cerință indipensabilă pentru raționament.

Componentele principale ale aplicației sunt:

1. Un *editor grafic al ontologiilor "grele"* (puternic formalizate), având la bază paradigma ER (entitate-relație);
2. Un *mecanism de operaționalizare* care folosește modelul grafurilor conceptuale;
3. Funcționalități de *verificare și validare* a ontologiei;
4. *Motor de inferență* care ne permite să utilizăm ontologiile operaționale pentru a raționa asupra bazelor de cunoștințe.

3.6.1 CAZURILE DE UTILIZARE

OntL_CG oferă o interfață care permite crearea/modificarea unei ontologii (incluzând vocabularul și axiomele), specificarea contextului în care va fi utilizată fiecare axiomă și operaționalizarea (parțială) ontologiei. Motorul de inferență ***folosește biblioteca CoGITaNT*** și realizează ciclul de inferență prezentat în figura 3-9. Interfața client este implementată în Java. Server-ul este unul CoGITaNT (implementat în C++). Comunicarea server-client se realizează prin fișiere .xml.

Funcționalitățile aplicației sunt prezentate în figura *3-24*, prin cazurile de utilizare (UML).

3.6.2 PREZENTAREA GENERALĂ A APLICAȚIEI

Aplicația OntL_CG permite utilizatorului să construiască și să opereționalizeze o ontologie a domeniului. Utilizatorul se conectează la server-ul CoGITaNT (din linie de comandă sau selectând opțiunea din meniul Options (figura AII - 19, figura AII - 1). Interfața este multi-lingvală (în momentul actual, engleză și română (parțial)).

Editorul de ontologii permite utilizatorului să definească termenii pentru conceptele și relațiile domeniului și să vizualizeze grafic ierarhia conceptelor și ierarhia relațiilor, în care apar și axiomele schemă sub forma unor simboluri grafice.

Pentru concepte, săgețile (→) indică legătura de subsumare, conceptele abstracte sunt reprezentate prin dreptunghiuri fără contur, iar simbolul ⊗ leagă două concepte disjuncte. Proprietățile unui concept (termen, abstractizare, disjuncție) pot fi specificate într-o fereastră separată.

Pentru relații, săgețile (→) indică legătura de subsumare, simbolul ⊗ leagă două relații incompatibile, iar simbolurile **S, T, R, I, A, C-**sau **C+** sunt folosite pentru relațiile simetrice, tranzitive, reflexive, nereflexive, antisimetrice, de cardinalitate minmă, respectiv maximă. Modificarea proprietăților unei relații se face, ca și în cazul conceptelor, în cadrul unei ferestre de editare (figura AII - 14).

Pentru o axiomă-schemă, reprezentată grafic, se generează automat un cuplu de λ-abstracțiuni (cazul proprietăților algebrice ale relațiilor și a cardinalității acestora), iar axioma este adăugată la lista axiomelor-schemă din ontologie.

Utilizatorul poate construi în mod grafic axiomele domeniului (orice fel de axiomă), selectând opțiunea Axioms (figura AII - 9). Ipoteza axiomei este colorată în galben, iar concluzia – în gri (formalismul grafurilor bicolore).

Ontologia poate fi salvată în format .cgxml[38] sau .owl.

Utilizatorul definește *scenariul de utilizare* a ontologiei, specificând pentru fiecare axiomă *contextul de utilizare* (figura AII - 5). Implicit, axiomele-schemă și axiomele domeniului sunt operaționalizate în context inferențial explicit. Utilizatorul poate salva scenariul de utilizare, poate defini pentru aceeași ontologie mai multe scenarii sau poate folosi un scenariu definit anterior (figura AII - 5, figura AII - 6). În urma scenariului considerat, ontologia operațională este generată în cadrul modelului grafurilor conceptuale (suportul corespunzător nivelului terminologic al ontologiei, regulile și constrângerile corespunzătoare axiomelor operaționalizate).

[38] Bazat pe sintaxa .xml, exprimată într-un DTD aflat la adresa http://cogitant.sourceforge.net/cogitant_html/cogxml.html; permite reprezentarea suportului, a grafurilor, a regulilor CGs și a grafurilor imbricate.

Motorul de inferență permite utilizarea ontologiilor operaționale în raționamentele asupra unei baze de cunoștințe. Motorul de inferență se bazează pe mecanismele de raționament din grafurile conceptuale, mai precis, pe proiecția între grafurile conceptuale. Mecanismele de raționament sunt implementate în CoGITaNT, iar aplicația OntL_CG, pentru calculul proiecțiilor, apelează server-ul CoGITaNT (comunicarea client-server se realizează prin fișiere .xml).

Raționamentul are loc *la nivelul instanțelor domeniului*, prin aplicarea regulilor și constrângerilor din ontologia operațională generată asupra bazei de cunoștințe.

Un aspect important al raționamentului este *verificarea/validarea* (prin selectarea opțiunii *Tests* din meniul principal, figura AII - 2). Așa cum am precizat în studiul detaliat din Anexa I (secțiunea 3.2) și în metoda MIIO, evaluarea unei ontologii (care presupune verificarea și validarea) se realizează prin *teste interne* (teste independente de domeniu, definite în raport cu sintaxa și semantica formală a limbajului de reprezentare, asupra reprezentării cunoștințelor conținute în ontologie) și prin *teste externe* (teste dependente de domeniu realizate prin întrebările de competență ale lui Grüninger). Verificarea/validarea ontologiei în aplicația OntL_CG se axează pe **testele interne**, la care se adaugă o serie de elemente din metodologia lui **Goméz-Pérez** de testare a taxonomiilor (așa cum am propus și în metoda MIIO).

În urma testării, pot fi emise mesaje de eroare (caz în care ontologia nu poate fi operaționalizată) sau de atenționare (figura AII - 18). Deasemenea, interfața gestionează unele constrângeri.

Testarea ierarhiei de concepte:

- Erori pentru situațiile în care:
  a) Există tipuri conceptuale care nu sunt descendenți (direcți sau indirecți) ai conceptului "Universal".
  b) Un tip conceptual nu poate avea doi părinți disjuncți.
  c) Ierarhia prezintă circularități.
- Atenționări
  a) Două tipuri conceptuale nu pot avea aceeași etichetă.
  b) Un tip conceptual poate avea un singur fiu.
- Gestionate prin interfață
  a) Trebuie să existe tipul "Universal".
  b) Un tip conceptual nu poate fi disjunct cu un alt tip conceptual aflat pe aceeași ramură a ierarhiei.
  c) Un tip conceptual nu poate deține mai multe relații de subsumare cu un alt tip conceptual.

Testarea mulțimii instanțelor:

- Erori:
  a) Două instanțe de tipuri disjuncte și diff nu pot avea aceeași etichetă.
  b) Două instanțe nu pot avea același tip și aceeași etichetă.
- Atenționări:
  a) Două instanțe de tipuri nedisjuncte și diff nu pot avea aceeași etichetă.
- Gestionate prin interfață:
  a) Un tip conceptual abstract nu poate avea instanțe.
  b) Un tip poate avea eticheta “diff”.

Teste asupra ierarhiei tipurilor de relații:

- Erori:
  a) Ierarhia tipurilor de relații prezintă circularități.
  b) Trebuie specificată semnătura fiecărui tip de relație.
  c) Semnătura fiecărui tip de relație trebuie să fie coerentă cu semnăturile părinților.
- Atenționări:
  a) Două tipuri de relații nu pot avea aceeași etichetă.
- Gestionate prin interfață:
  a) Semnătura fiecărui tip relație trebuie să fie aceeași cu cele ale tipurilor de relații exclusive sau incompatibile.
  b) Două tipuri relaționale exclusive sau incompatibile nu pot fi părinți pe aceeași ramură a ierarhiei.
  c) Proprietățile trebuie să fie coerente cu semnătura tipurilor de relații.
  d) Un tip relațional nu poate deține mai multe legături de subsumare cu un alt tip relațional.

Teste asupra fiecărei axiome:

- Erori
  a) Fiecare concept care compune axioma (atât din partea de ipoteză a axiomei, cât și din partea de concluzie a acesteia) trebuie să aibă un tip.
  b) Fiecare relație care compune axioma (atât din partea de ipoteză a axiomei, cât și din pertea de concluzie a acesteia) trebuie să aibă un tip.
  c) Numărul de concepte relaționate trebuie să fie egal cu aritatea relației respective iar tipurile conceptelor trebuie să fie compatibile cu semnătura tipului de relație.
  d) Toate relațiile din partea de ipoteză a axiomei trebuie să lege conceptele din partea de ipoteză a axiomei.
- Atenționări
  a) Nu pot exista două concepte identice (cu același tip și acelasi referent, sau cu același tip și ambele generice, sau cu relația “diff” între ele).
  b) Nu pot exista în aceeași axiomă două relații exclusive sau incompatiile.
  c) Relația “diff” nu poate lega două concepte identice.
  d) Pentru fiecare axiomă, partea de ipoteză și cea de concluzie nu pot fragmenta constrângerile induse de către alte axiome.

În Anexa II sunt ilustrate ecranele din ontologia de reprezentare OntL_CG-MetaOnto (ontologia de meta-nivel).

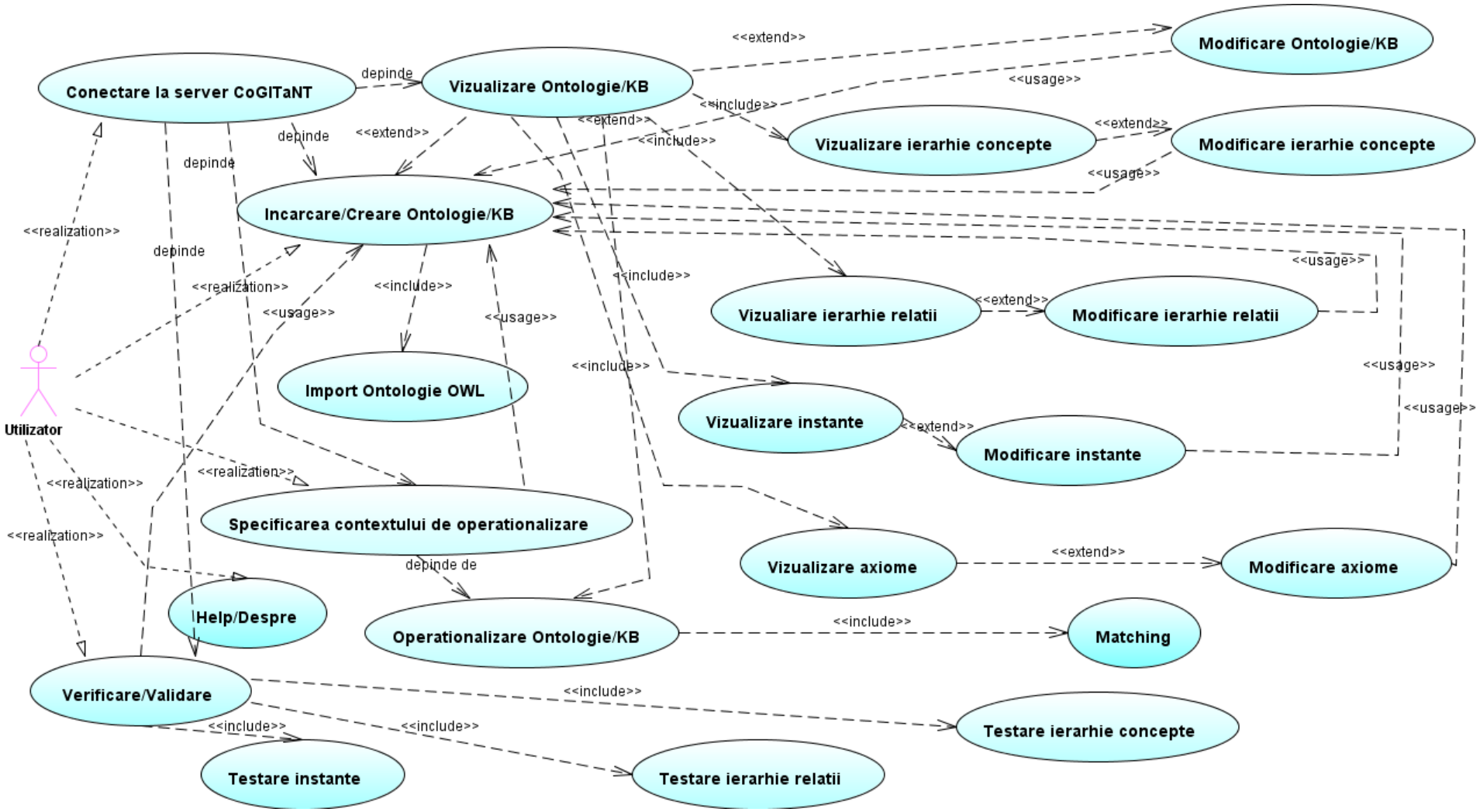


Figura 3-24 OntL_CG – Cazuri de utilizare

3.6.3 STRUCTURA APLICAȚIEI CLIENT OntL_CG

Aplicația este structurată în 6 pachete. In figura 3-25 sunt reprezentate pachetele și legăturile dintre ele. Așa cum se observă, nu există circularități în dependența pachetelor.

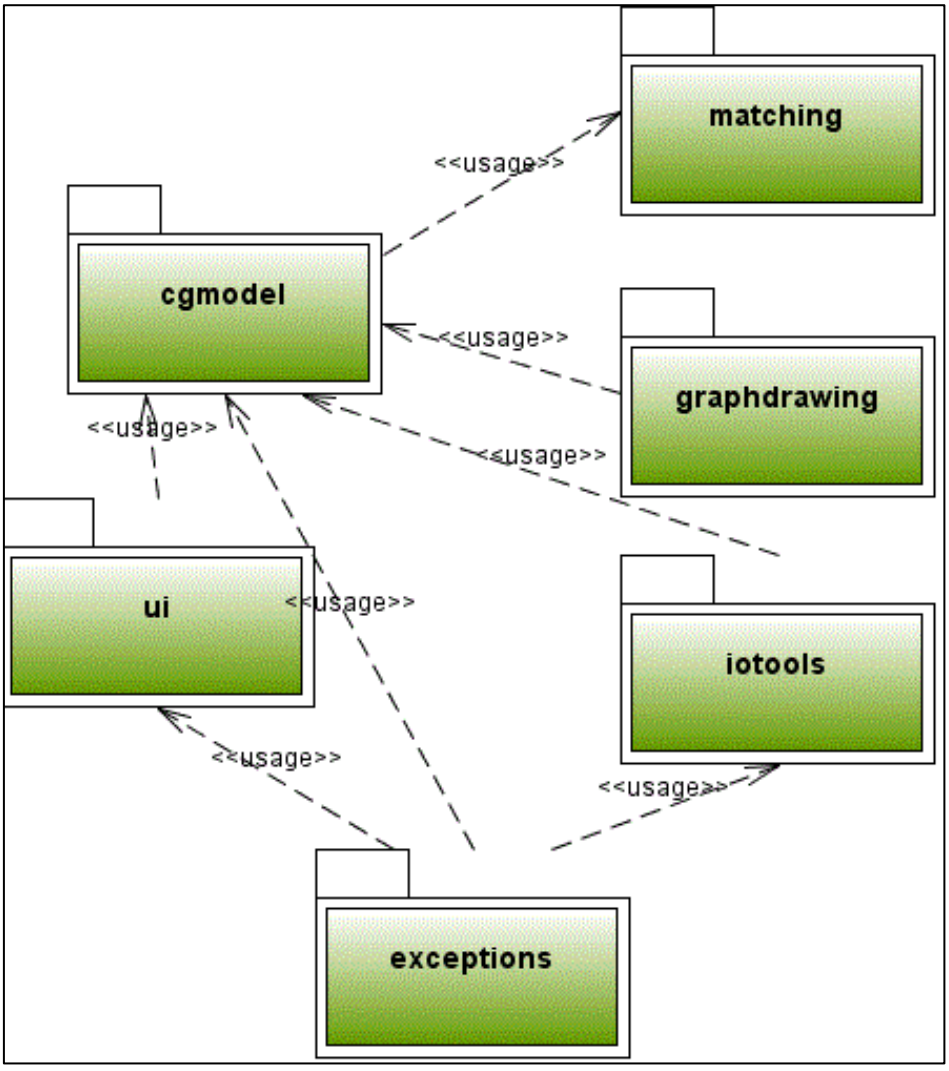


Figura 3-25 Pachetele din aplicație

Pachetul cgmodel

Pachetul **cgmodel** este pachetul principal, care conține clasele care implementează modelul propus, bazat pe grafurile conceptuale.

Clasele conținute în acest pachet sunt ilustrate în figura 3-26.

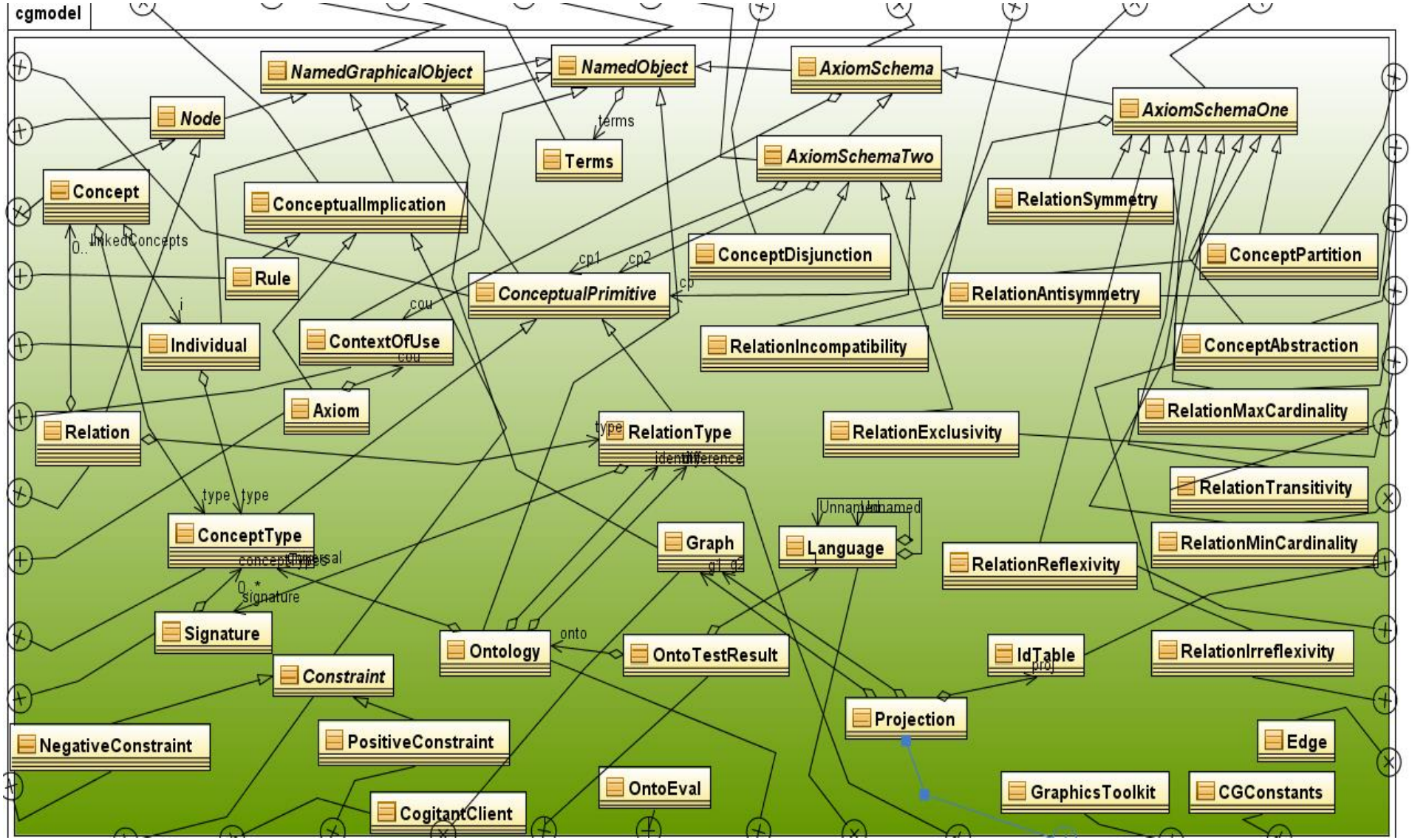


Figura 3-26 Clasele din pachetul *cgmodel*

Sunt prezentate sumar clasele care implementează modelul și rolul acestora.

Clasa **CogitantClient** - Reprezintă clientul care comunică cu serverul Cogitant și furnizează metodele de creare și manipulare a obiectelor conceptuale pe server (figura 3-27).

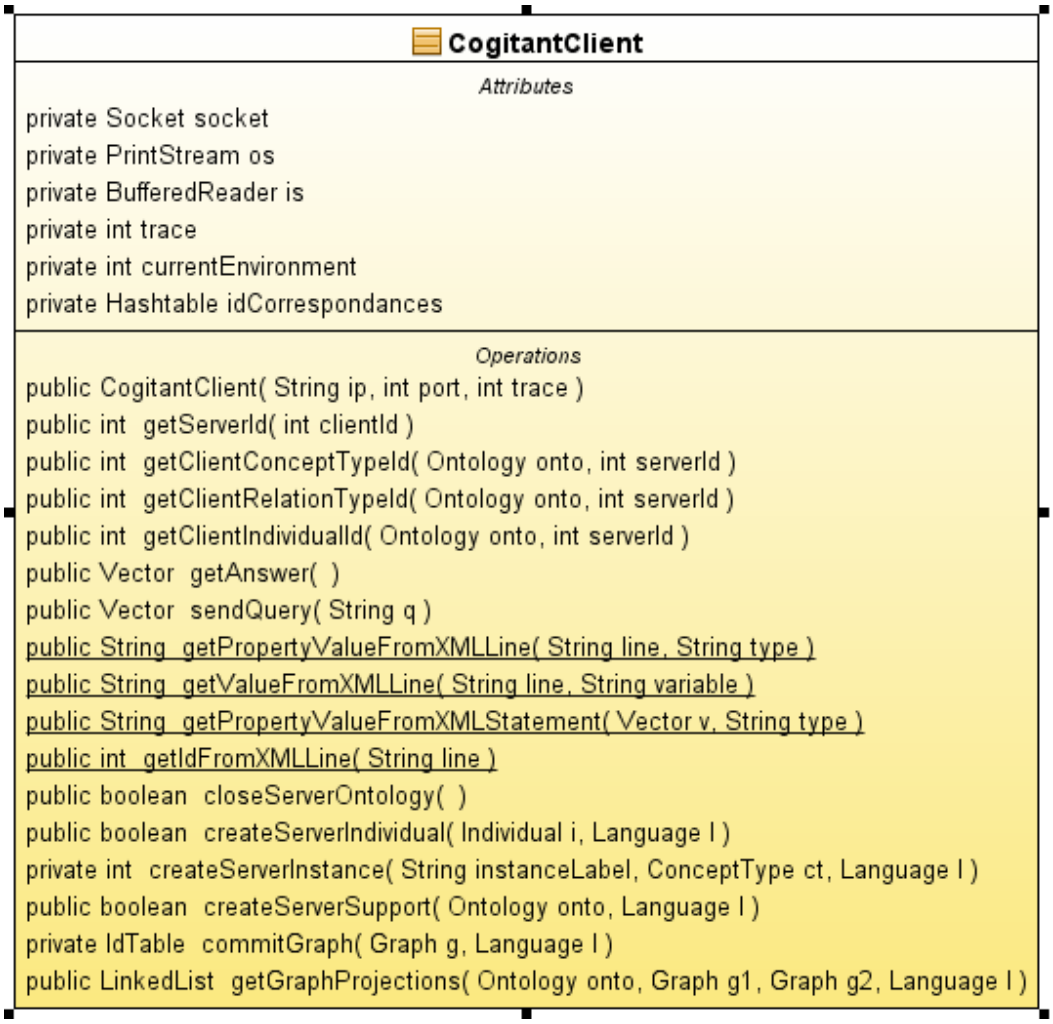


Figura 3-27 Clasa *CogitantClient*

Clasa **CGConstants** – Definește constantele utilizate în pachet.

Clasa **Language** – Oferă funcționalități de internaționalizare, prin posibilitatea de a specifica termenii din ontologie în mai multe limbi (engleză și română în momentul actual, cu posibilități de extindere în viitor). Fiecărei limbi i s-a atribuit o etichetă.

Clasa **Terms** – Permite specificarea termenilor - folosiți pentru un obiect din ontologie - în diferite limbi. Fiecare termen este legat de un anumit limbaj. Limbile disponibile în acest moment sunt engleza și româna.

Clasa **GraphicsToolkit** – Furnizează metodele grafice pentru desenarea primitivelor conceptuale (concepte și relații), a grafurilor și a axiomelor.

Clasa **IdTable** – Clasa pentru tabela care conține cuplurile de id-uri (int/int) găsite prin operația de proiecție între două grafuri (figura 3-28).

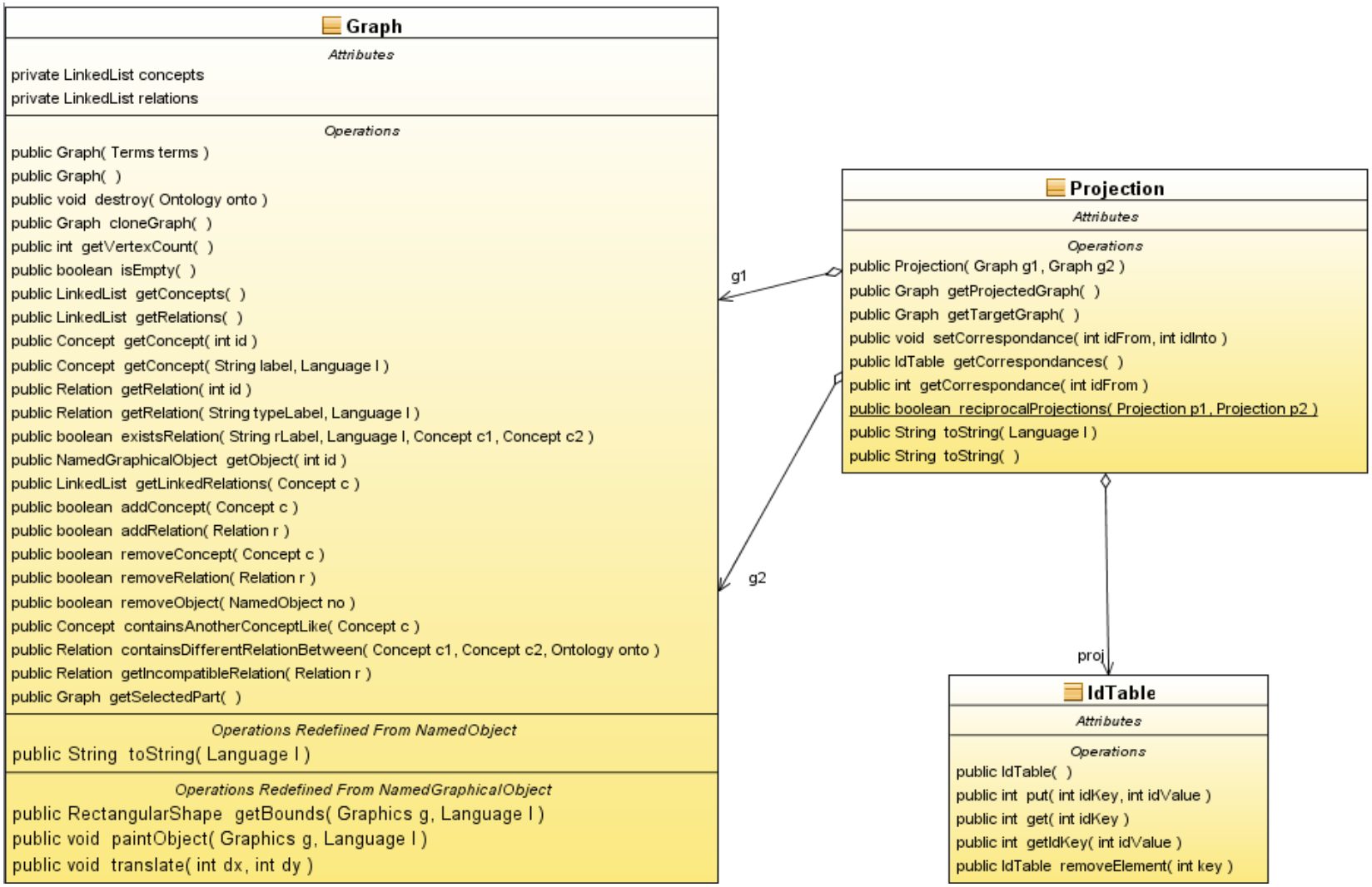


Figura 3-28 Clasele *Graph, Projection* și *idTable*

Clasa **Ontology** – Clasa pentru reprezentarea ontologiei. Ontologia conține ierarhia tipurilor de concepte, ierarhia tipurilor de relații, mulțimea axiomelor și graful ontologiei. Pentru crearea unui nou tip de concept/relație sau a unui nou graf se utilizează metoda new(Object), unde Object este numele clasei obiectului conceptual care va fi construit. Clasa conține, deasemenea, metode pentru construirea axiomelor. Sunt folosite cele două primitive conceptuale particulare: *tipul de concept universal* (T) și *tipul de relație pentru diferență (diff)*.

Clasa **OntoTestResult** – Clasă pentru rezultatele testării ontologiei.

Clasa **OntoEval** – Permite verificarea și validarea ontologiei și vizualizarea barei de progres.

Clasa **ConceptualPrimitive** – Reprezintă primitivele conceptuale (tipurile de concepte și tipurile de relații) din ontologie, structurate prin relația *is-a*.

Clasa **ConceptType** – Pentru reprezentarea tipurilor de concepte din ontologie.

Clasa **Individual** – Pentru marcatorii individuali din ontologie.

Clasa **RelationType** – Pentru reprezentarea tipurilor de relații din ontologie. Fiecare tip de relație are o semnătură și o serie de proprietăți (figura 3-29).

Clasa **Concept** – Reprezintă conceptele din ontologie. Într-o axiomă, un concept poate juca rolul de ipoteză sau de concluzie. Deasemenea, un concept poate fi o instanță ontologică (metoda getIndividual() returnează instanța, nenulă) sau nu (metoda getIndividual() returnează null, dar metoda getTerms() returnează termenii care nu sunt egali cu "").

Componentele ontologiei (clasa *Ontology, clasele ConceptType, Concept* pentru concepte*),* clasa *Individuals* pentru instanțe*,* clasele *RelationType, Relation* pentru relații, clasa *Signature* pentru semnătura relațiilor*),* clasa *Ontology* și clasa *ConceptualPrimitive* sunt prezentate în figura 3-29, figura 3-30, respectiv figura 3-31.

*Clasele **Ontology**, **OntoTestResult**, **Language**, **ConceptType**, **Concept** și **Individual)***

Clase pentru concepte și proprietățile acestora:

Clasa **ConceptAbstraction** – Reprezintă abstractizarea unui concept.

Clasa **ConceptDisjunction** – Reprezină disjuncția a două concepte (nu pot avea individuali comuni).

Clasa **ConceptPartition** – Reprezintă partiția unui concept (disjuncția între toți fiii conceptului).

Clase pentru relații și proprietățile acestora:

Clasa **Relation** – Clasa pentru relațiile într-un graf/axiomă din ontologie.

Clasa **Signature** – Clasa pentru semnătura unei relații (1 e primul index al semnăturii).

Clasele **RelationMinCardinality** și **RelationMaxCardinality** – Clasele pentru reprezentarea cardinalității minime, respectiv maxime a unei relații.

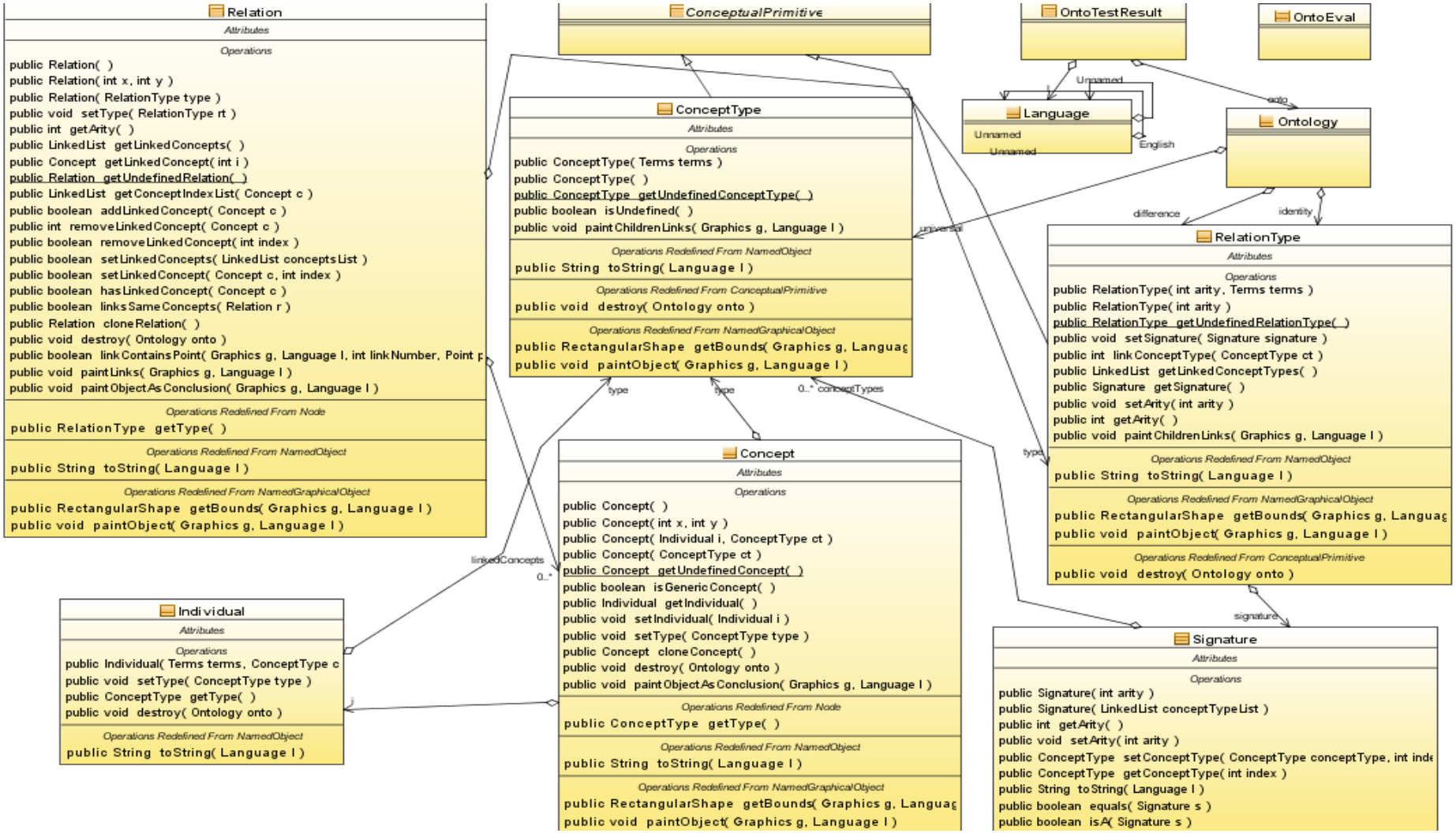


Figura 3-29 Componentele ontologiei

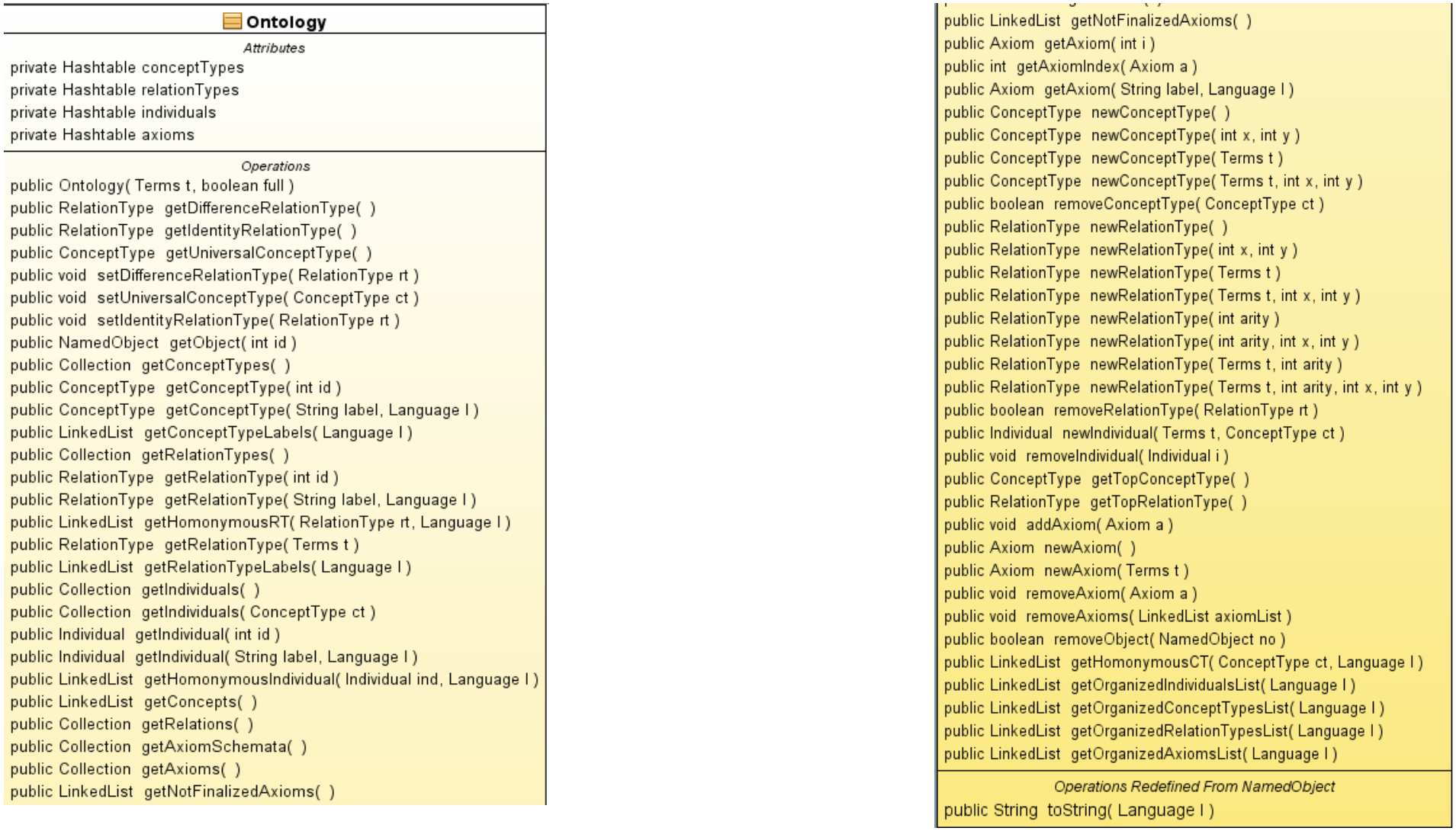


Figura 3-30 Clasa Ontology

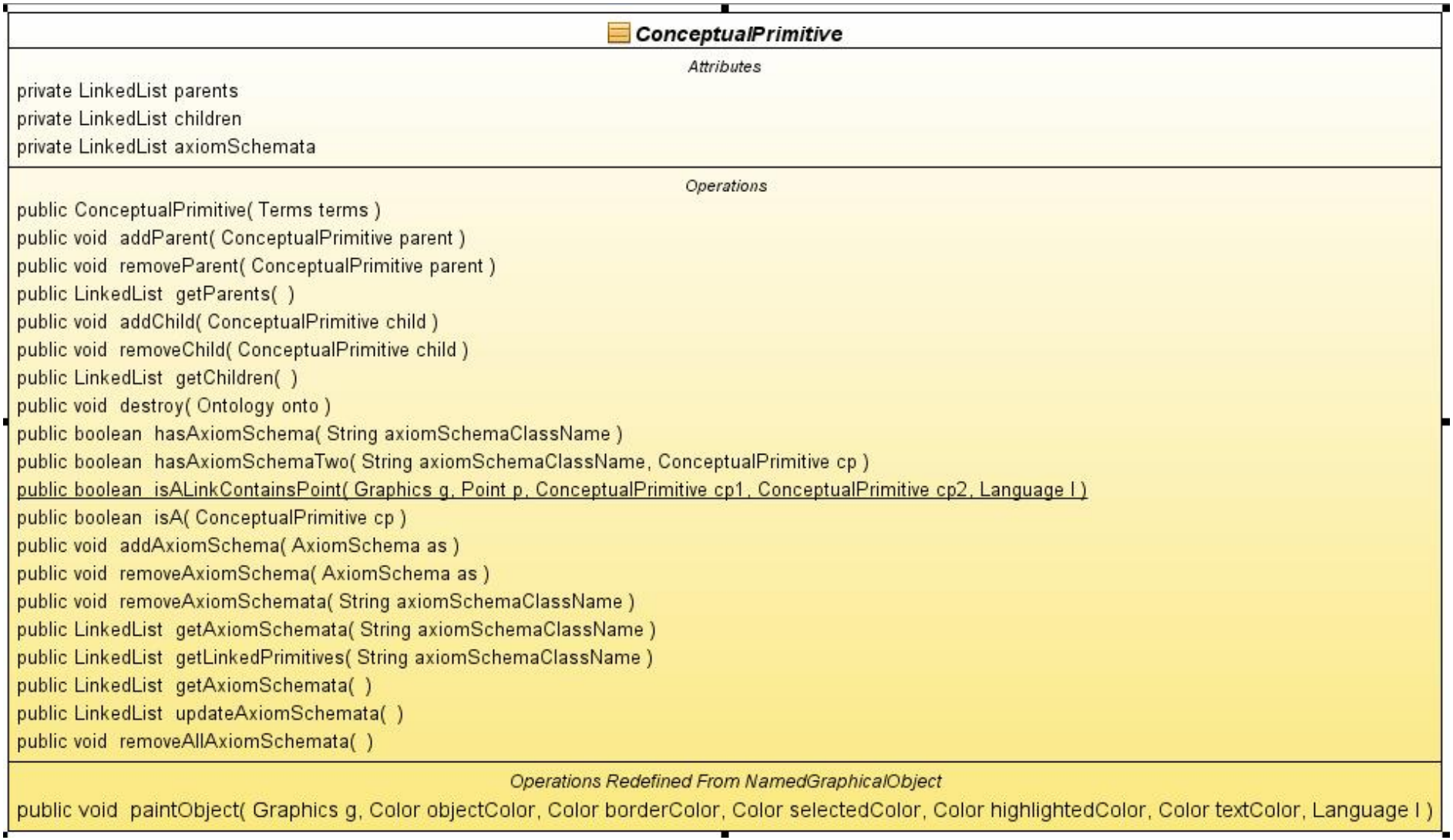


Figura 3-31 Clasa *ConceptualPrimitive*

Clasele **RelationRreflexivity** și **RelationIrreflexivity** – Clasele pentru reprezentarea reflexivității, respectiv a nereflexivității unei relații.

Clasa **RelationExclusivity** – Clasă pentru reprezentarea exclusivității dintre două relații (o relație poate exista două concepte, cealaltă – nu).

Clasa **RelationIncompatibility** – Reprezintă incompatibilitatea dintre două relații (nu pot lega aceleași concepte).

Clasa **RelationTransitivity** – Reprezintă tranzitivitatea unei relații.

Clasa **NamedGraphicalObject** – Reprezintă numele generic al unui obiect grafic. Fiecare obiect este descris prin mai mulți termeni și o poziție. Aceste obiecte pot fi selectate.

Clasa **NamedObject** – Reprezintă un obiect “multilingual”. Include id-ul (generat automat la crearea obiectului) pentru identificarea obiectului în sistem. Constructorii tuturor subclaselor lui NamedObject trebuie să apeleze constructorul acestei clase cu parametrul Terms (metoda super(Terms)).

Clasa **Graph** – Reprezintă grafurile conceptuale (grafuri bipartite, cu noduri-concept și noduri-relație).

Clasa **Node** – Reprezintă un nod (dintr-un graf sau dintr-o axiomă). Extinde clasa NamedGraphicalObject prin adăugarea atributelor (dx, dy) pentru implementarea algoritmului pentru di-grafuri. Un alt atribut determnină dacă nodul este fixat (este mutat de către utilizator și nu trebuie mutat prin algoritm).

Clasa **Edge** – Reprezintă un arc într-un graf și este utilizată de către clasa GraphDrawing. Atributele clasei (figura 3-32) sunt originea (from), vârful (to) și lungimea arcului – len (în pixeli).

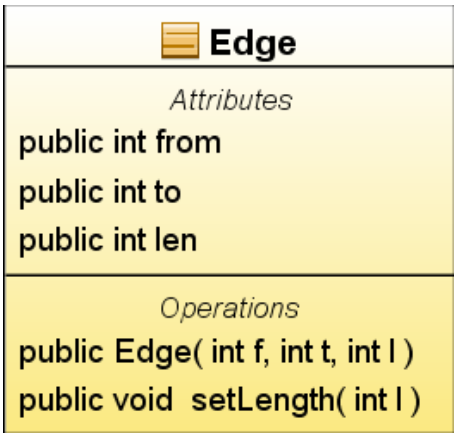


Figura 3-32 Clasa *Edge*

Clasa **Projection** – Calculează proiecția unui graf în alt graf.

Clasa **Axiom** – Reprezintă axiomele din ontologie, deci partea de ipoteză și cea de concluzie. Axioma poate fi “hard” sau “soft” (implicit) și poate avea un context de utilizare care specifică modul în care va se vor realiza inferențele. O axiomă “finală” nu poate fi modificată de către

utilizator, exceptând contextul de utilizare (de obicei, axiomele care reprezintă proprietățile algebrice ale tipurilor de relații, nu pot fi modificate).

Clasa **ContextOfUse** – Reprezintă contextul de utilizarea a axiomelor în ontologie. Contextul poate fi *inferențial/explicit, inferențial/implicit, verificare/implicit și verificare/explicit* (figura 3-33).

Clasa **AxiomSchema** – Reprezintă schema axiomelor din ontologie (figura 3-34).

Clasa **AxiomSchemaOne** – Reprezintă schema axiomelor cu o singură primitivă (figura 3-35).

Clasa **AxiomSchemaTwo** – Clasa pentru schema axiomelor cu două primitive (figura 3-34, figura 3-36).

Clasa **Constraint** – Clasă derivată din ConceptualImplication: conține metodele pentru exprimarea constrângerilor asupra grafurilor conceptuale ().

Clasa **PositiveConstraint** – Clasa pentru constrângerile pozitive asupra CG.

Clasa **NegativeConstraint** - Clasa pentru constrângerile negatve asupra CG.

Clasa **Rule** – Reprezintă regulile CG (figura 3-33).

Clasele pentru raționament (axiome – clasa *Axiom*, constrângeri, cu clasele derivate constrângeri pozitive și costrângeri negative - clasele *Constraint*, *PositiveConstraint* și *NegativeConstraint*; reguli – clasa *Rule*) sunt derivate din clasa *ConceptualImplication* (figura 3-33).

Pachetul matching

Clasele din pachetul matching sunt ilustrate în figura 3-37.

Clasa **Matching** – Reprezintă punerea în corespondență a primitivelor conceptuale.

Clasa **MatchingAlgorithm** – Conține parametri folosiți în algoritmul de punere în corespondență.

Clasa **MatchingList** – Construiește lista punerii în corespondență a primitivelor conceptuale.

Clasa **MatchingResultForm** – Reprezintă fereastra grafică în care va fi afișat rezultatul punerilor în corespondență găsite în ontologie.

Pachetul graphdrawing

Pachetul conține clasele pentru desenarea grafurilor (figura 3-38).

Clasa **TreeDrawing** – Afișează arborele în Graphics (lista primitivelor conceptuale ale clasei).

Clasa **Walker** – Afișează arborele bi-dimensional folosind algoritmul optimizat al lui Walker (o versiune optimizată a algoritmului Reingold-Tilford).

Clasa **TreeNode** – Conține structura arborelui de afișat.

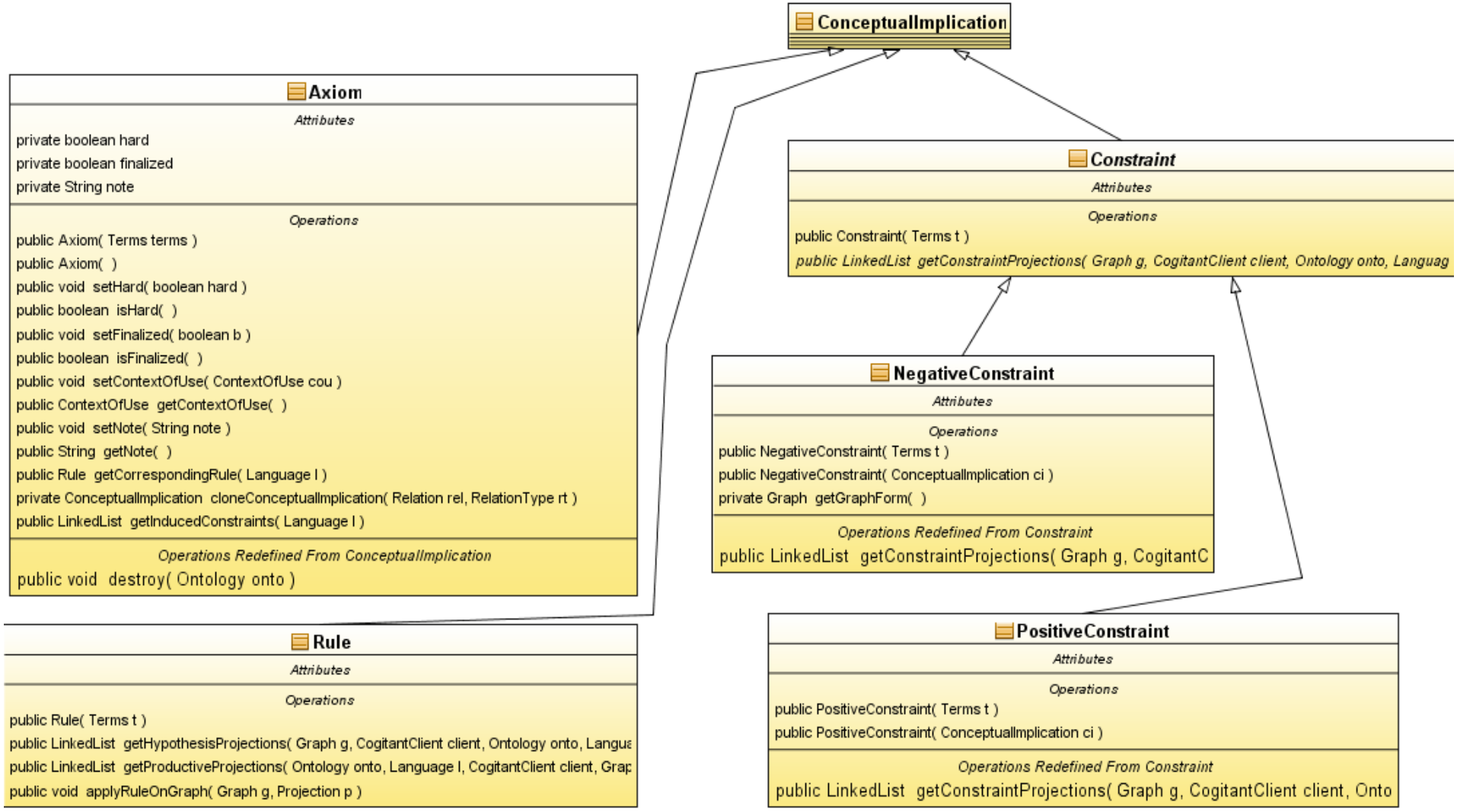


Figura 3-33 Clase pentru raționament (axiome, constrângeri pozitive și negative, reguli)

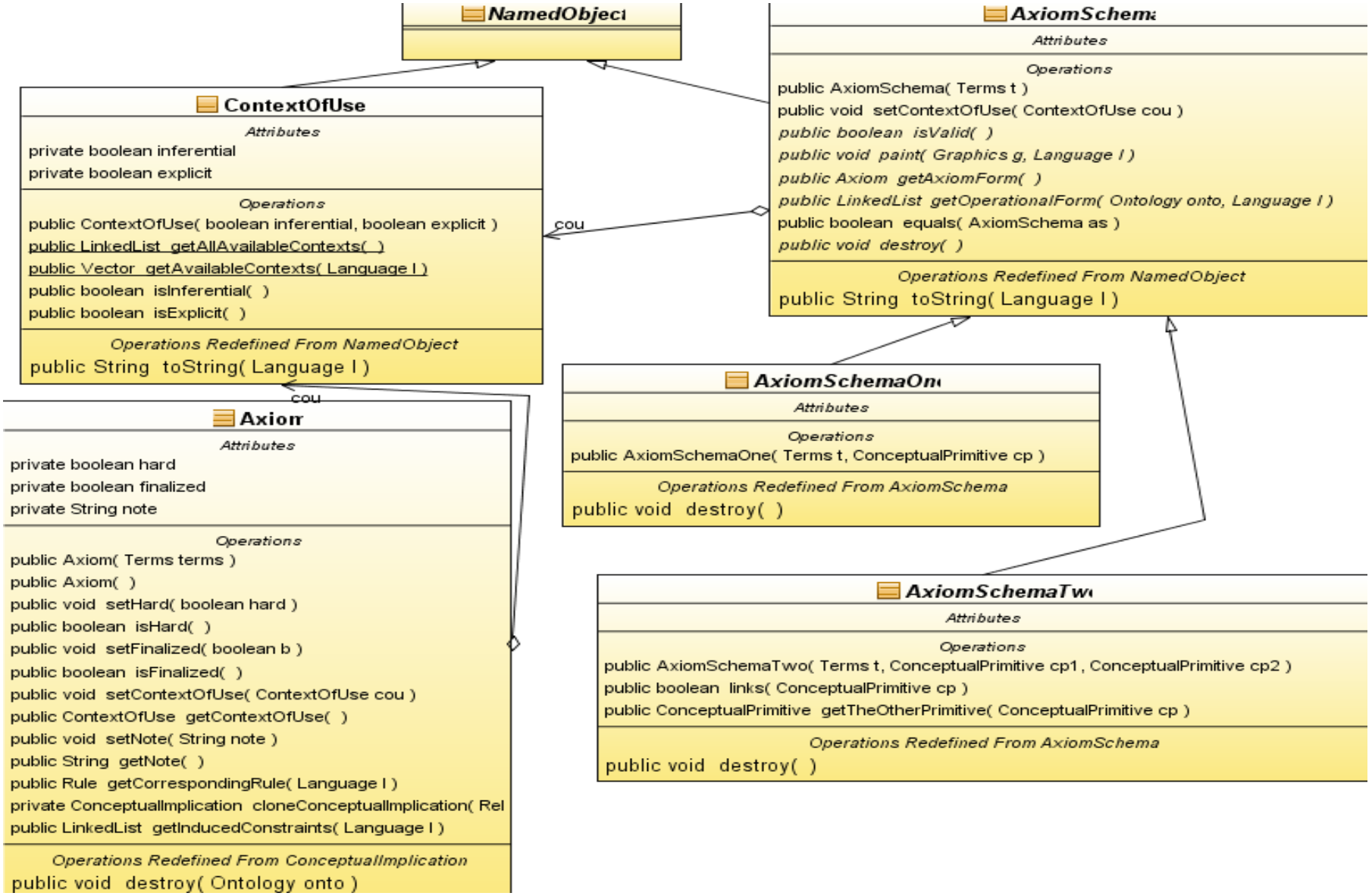


Figura 3-34 Axiome-schemă și contextul de utilizare

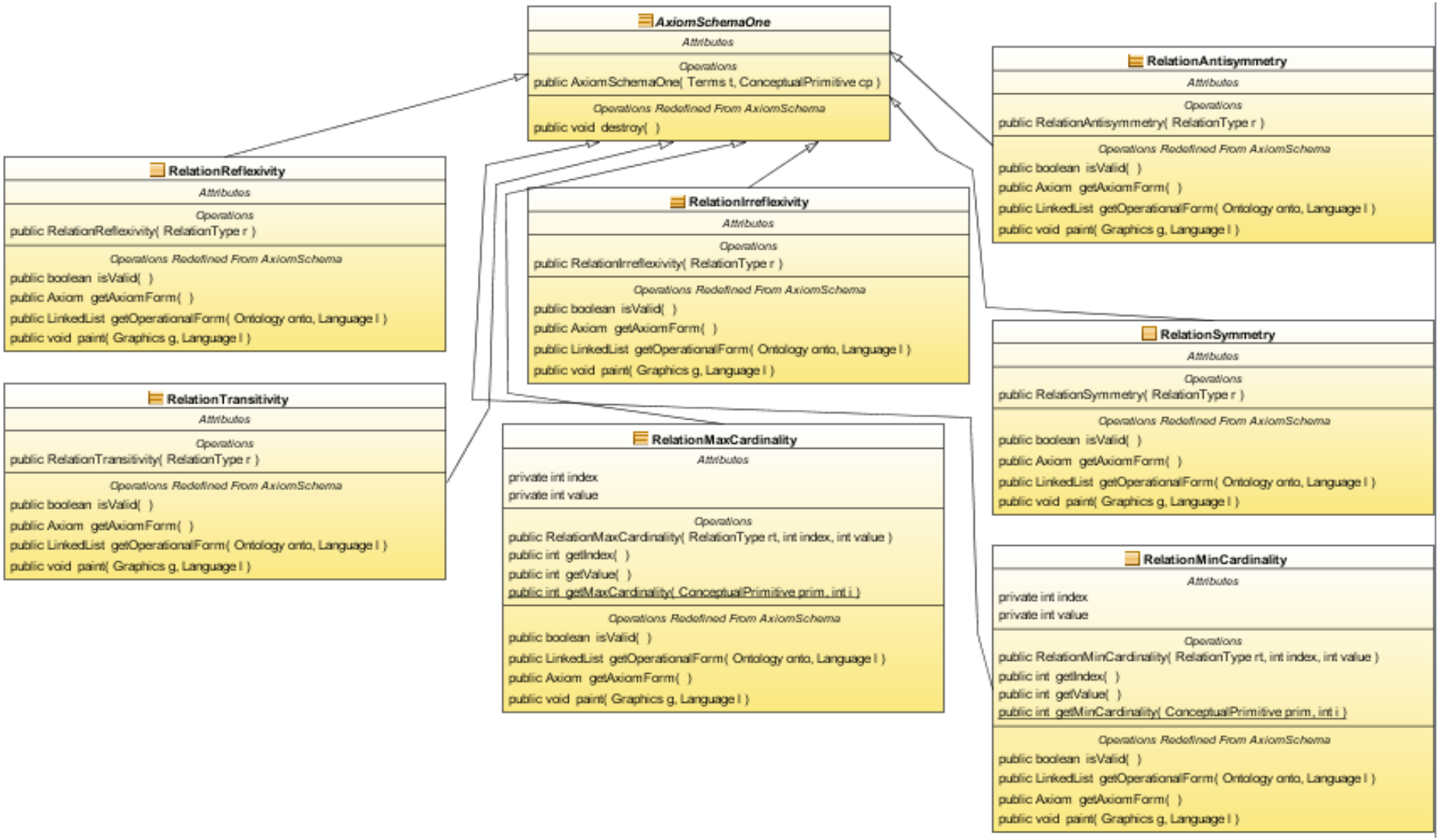


Figura 3-35 Axiome-schemă cu o singură primitivă (*AxiomSchemaOne*) și clasele pentru proprietățile relațiilor

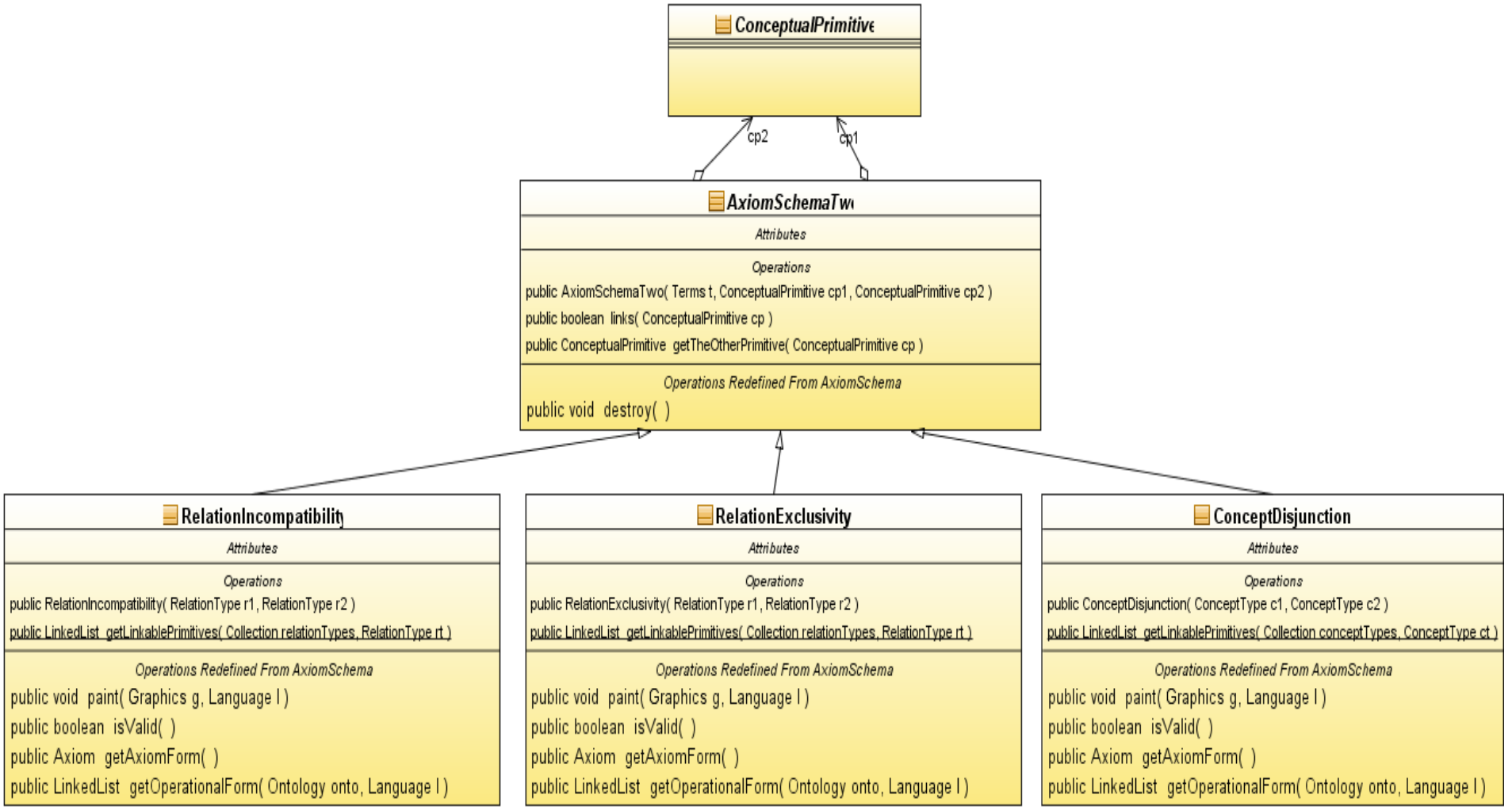


Figura 3-36 *A*xiome-schemă (*AxiomSchemaTwo*) cu două primitive

Pachetul iotools

Clasele din acest pachet sunt reprezentate în figura 3-39 și conțin clasele care **implementează operațiile de intrare/ieșire pentru fișierele în formatele *.cgxml* și *.owl*.**

Clasa **IOInterface** – Interfața pentru operațiile de intrare/ieșire.

Clasa **OntologyFileFilter** – Filtrul de fișiere pentru fișierele care conțin ontologia în format .cgxml/.owl.

Clasa **OperationalizationFileFilter** – Filtrul pentru fișierele care specifică modul de operaționalizare (extensia *.ope*).

Clasa **CGXMLFileFilter** - Filtrul de fișiere pentru fișierele *.cgxml*.

Clasa **CGXMLParser** – Clasa conține metodele pentru parsarea și salvarea fișierelor *.cgxml*.

Clasa **OWLFileFilter** - Filtrul de fișiere pentru fișierele *.owl*.

Clasa **OWLParser** - Clasa conține metodele pentru parsarea și salvarea fișierelor *.owl*.

Pachetul exceptions

Pachetul exceptions conține o serie de clase pentru tratarea excepțiilor. Diagrama claselor din acest pachet este prezentată în figura 3-40.

Clasa **OntologyFileLoadingException** - Excepțiile în cazul încărcării unui fișier care conține ontologia.

Clasa **BadFileSyntaxException** - Excepțiile în cazul unui tip de fișier care nu respectă sintaxa .cgxml/.owl/.ope.

Clasa **ConceptualPrimitivePropertyException** – Reprezintă excepțiile în cazul unei primitive conceptuale cu proprietate incorectă.

Clasa **NoExclusiveRelationTypeException** – Excepțiile în cazul unuor relații exclusive.

Clasa **RelationTypePropertyException** – Excepțiile în cazul unui tip de relație cu o proprietate incorectă.

Clasa **WrongSignatureException** - Excepțiile în cazul unui tip de relație cu semnătură incorectă.

Pachetul ui

Pachetul conține 35 de clase pentru interfața cu utilizatorul. Diagrama claselor din acest pachet este prezentată în figura 3-41.

Clasa **Constants** – Conține constantele aplicației, incluzându-le pe cele de la lansarea acesteia (directorul curent de lucru, rezoluția ecranului, etc.).

Clasa **InterfaceIcon** – Furnizează metodele de creare a icon-urilor pentru interfață.

Figura 3-37 Pachetul *matching* și clasele din pachet

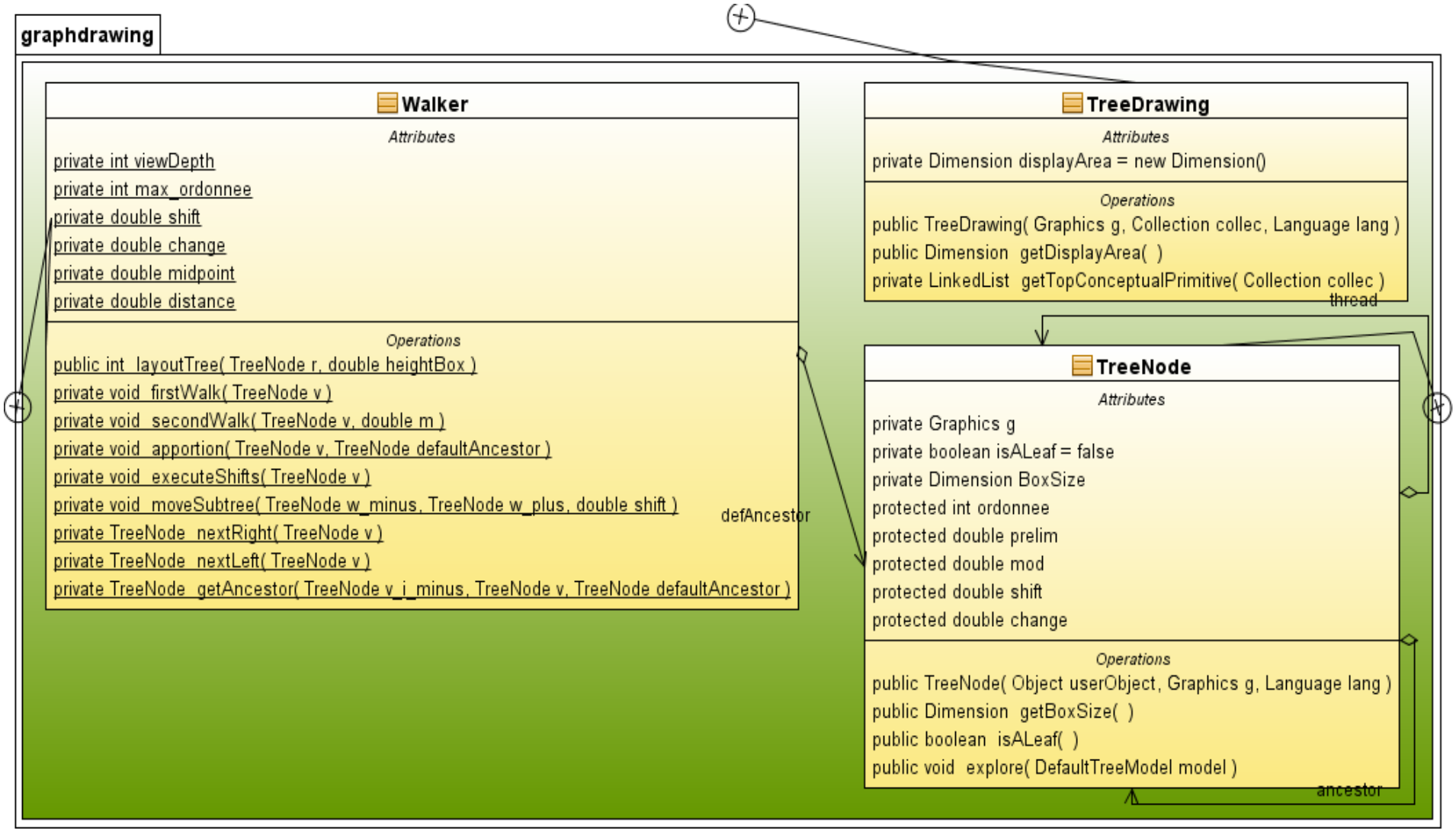


Figura 3-38 Pachetul *graphdrawing* și clasele din pachet

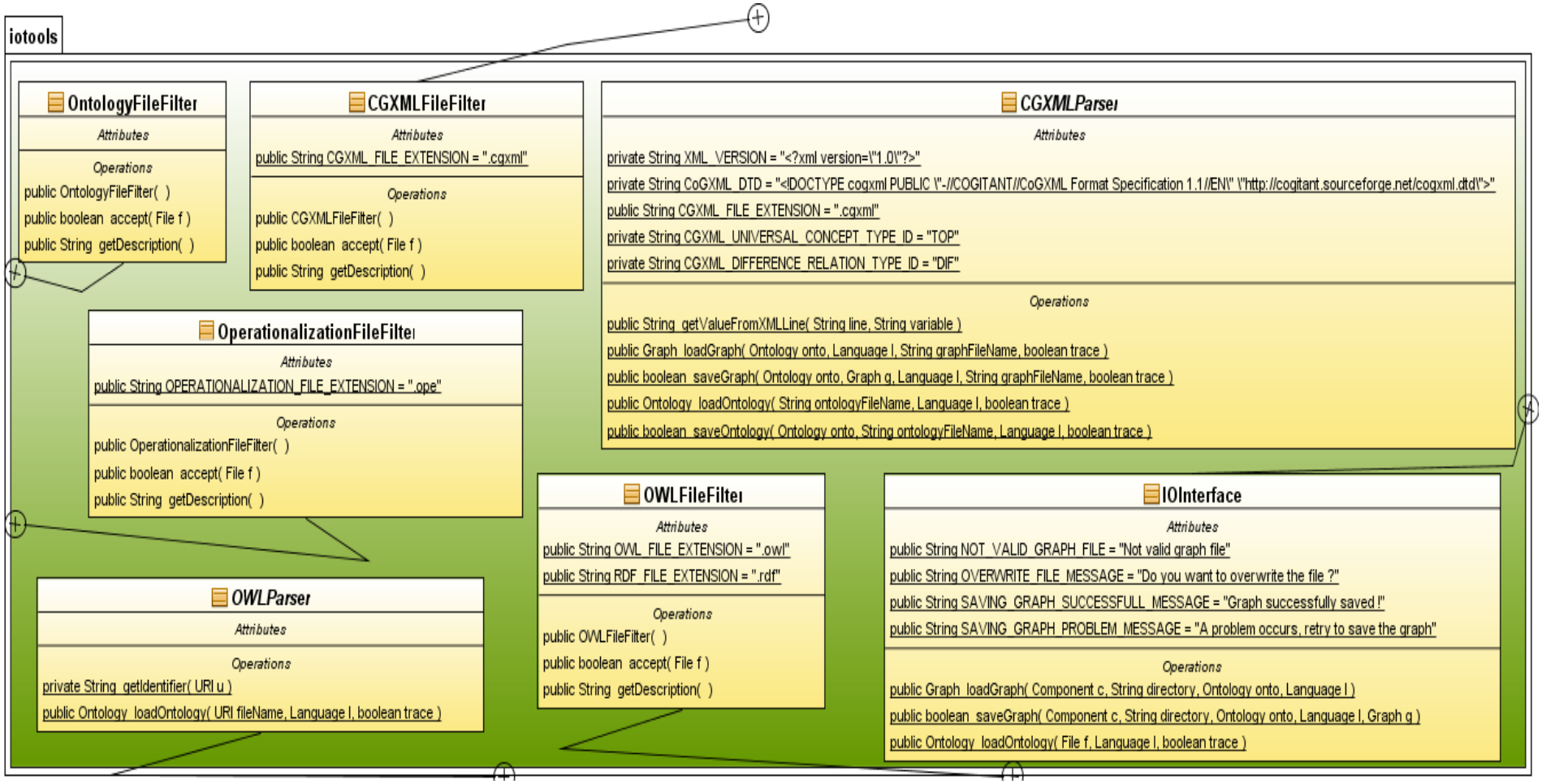


Figura 3-39 Pachetul *iotools* și clasele din pachet

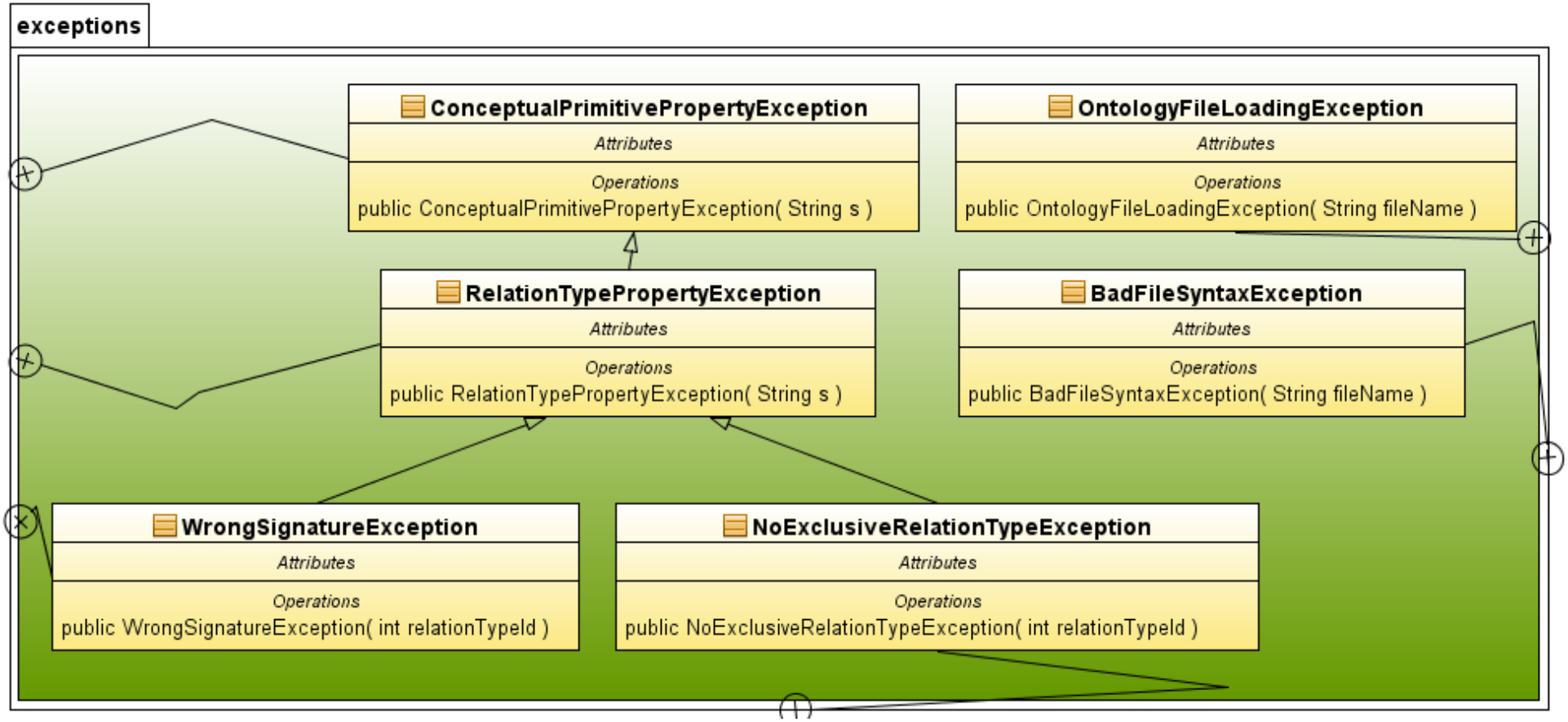


Figura 3-40 Pachetul *exceptions* și clasele din pachet

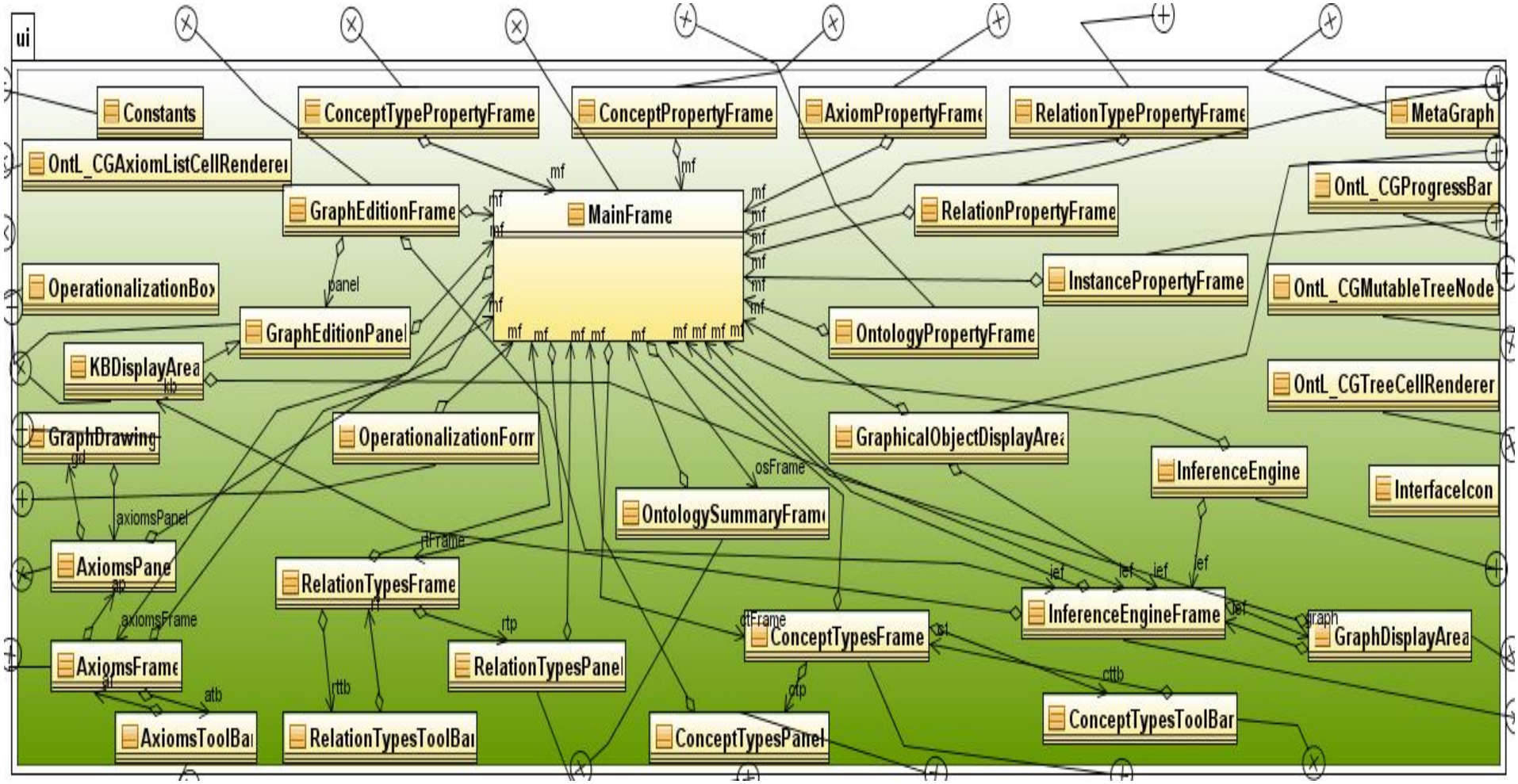


Figura 3-41 Pachetul *ui* pentru interfața cu utilizatorul și clasele din pachet

Clasa **MainFrame** – Fereastra grafică principală a aplicației (în figura 3-42 sunt clasele pentru ferestrele grafice ale aplicației).

Clasa **Metagraph** – Clasă pentru construirea metagrafului ontologiei.

Clasa **KBDisplayArea** – Fereastra grafică de afișare a bazei de cunoștințe.

Clasa **OntologyPropertyFrame** – Fereastra grafică pentru specificarea proprietăților ontologiei.

Clasa **OntologySummaryFrame** – Fereastra grafică de prezentare și accesare a obiectelor ontologiei. Furnizează funcții de manipulare a obiectelor (prin click-dreapta de mouse).

Clasele pentru ferestrele grafice despre concepte:

Clasa **ConceptTypesPanel** – Fereastra grafică pentru ierarhia tipurilor de concepte. Oferă funcționalități de creare și structurare prin bara de instrumente (creare/distrugere/legare concepte prin relația is-a) sau prin click-dreapta al mouse-ului (determină deschiderea unei ferestre pentru proprietățile conceptului)

Clasa **ConceptTypesToolBar** – Fereastra grafică pentru construirea ierarhiei tipurilor de concepte.

Clasa **ConceptPropertyFrame** – Fereastra grafică pentru specificarea proprietăților unui concept.

Clasa **ConceptTypePropertyFrame** – Fereasta grafică pentru specificarea proprietăților unui tip de concept.

Clasa **ConceptTypesFrame** – Fereastra grafică pentru crearea/structurarea axiomelor, prin bara de instrumente sau click-dreapta de mouse.

Clasa **InstancePropertyFrame** – Fereastra grafică de specificare a proprietăților instanței unui tip de concept.

Clasele pentru ferestrele grafice despre relații:

Clasa**RelationTypesPanel** – Clasă pentru fereastra grafică de construire a ierarhiei tipurilor de relații. Oferă posibilitatea de creare/structurare prin bara de instrumente sau click-dreapta pe mouse (deschiderea ferestei de proprietăți).

Clasa **RelationTypesToolBar** – Fereastra grafică pentru construirea ierarhiei tipurilor de relații, prin bara de instrumente sau click-dreapta pe mouse (deschiderea ferestei de proprietăți ale relației).

Clasa **RelationTypesFrame** – Fereastra grafică pentru construirea ierarhiei tipurilor de relații prin bara de instrumente (crearea/distrugerea/definirea unei relații de tipul is-a) sau click-dreapta pe mouse (deschiderea ferestei de proprietăți ale relației).

Clasa**RelationTypePropertyFrame** – Fereastră grafică pentru specificarea proprietăților unui tip de relație.

Clasa **RelationPropertyFrame** – Fereastă grafică pentru specificarea proprietăților unei relații.

Clasele pentru ferestrele grafice despre axiome:

Clasa **AxiomsFrame** – Fereastră grafică pentru construirea axiomelor, compusă dintr-un obiect AxiomsPanel în care se afișează sau se modifică axioma (ca graf) și AxiomsToolBar pentru acțiunile asupra axiomei. Clasa AxiomsFrame furnizează funcțiile de comunicare între AxiomsToolBar și AxiomsPanel.

Clasa **AxiomsToolBar** – Grupează toate butoanele pentru acțiunile realizabile asupra axiomei curente de lucru.

Clasa **AxiomsPanel** – Permite desenarea unei axiome, mutarea nodurilor, adăugarea/ștergerea de arce și de noduri, crearea și structurarea tuturor acțiunilor care pot fi realizate asupra unei axiome.

Clasa **AxiomPropertyFrame** – Fereastra grafică pentru specificarea proprietăților unei axiome.

Clasa **GraphDisplayArea** – Reprezintă zona în care va fi afișat graful/axioma.

Clasa **GraphicalObjectDisplayArea** – Panoul grafic de afișare a unui graf/axiome.

Clasa **GraphDrawing** – Reprezintă algoritmul forței direcționate (engl. “force directed algorithm”) utilizat în desenarea grafului. Algoritmul folosește o reprezentare proprie a grafului, optimizată prin aplicarea algoritmului. Atributele nnodes, nodes, nedges și edges furnizează o reprezentare optimizată a grafului, deoarece vectoriii de pointeri permit un acces rapid la noduri și arce (acces linear). Atributul temperature reprezintă numărul maxim de mișcări pemise pentru fiecare punct (în pixeli) și poate fi micșorat. Un pointer către AxiomPanel permite redesenarea lui frecventă, clasa GraphDrawing executând algoritmul într-un thread separat.

Clasa **GraphEditionFrame** – Reprezintă fereastra grafică de editare a unui graf.

Clasa **GraphEditionPanel** – Panoul (inclus în JscrollPanel) în care utilizatorul va desena graful.

Clasele **InferenceEngine** și **InferenceEngineFrame** – Fereastrele grafice în care rulează motorul de inferență (figura 3-42).

Clasa **OperationalizationBox** – Folosită în operaționalizarea axiomelor.

Clasa **OperationalizationForm** – Fereastra grafică de alegere a contextului utilizării unei axiome.

Ferestre grafice pentru motorul de inferența, operaționalizarea axiomelor, contextul de utilizare

Clasa **OntL_CGAxiomListCellRenderer** – Furnizează lista axiomelor în frame-ul axiomelor.

Clasa **OntL_CGMutableTreeNode** – Reprezintă nodurile arborelui din interfață.

Clasa **OntL_CGTreeCellRenderer** – Furnizează arborele din rezumatul ontologiei.

Clasa **OntL_CGProgressBar** – Fereastra grafică de prezentare și accesare a obiectelor din ontologie. Furnizează acțiunile corespunzătoare click-ului-dreapta pe mouse.

Pentru dezvoltarea aplicației am folosit **mediul Netbeans**.

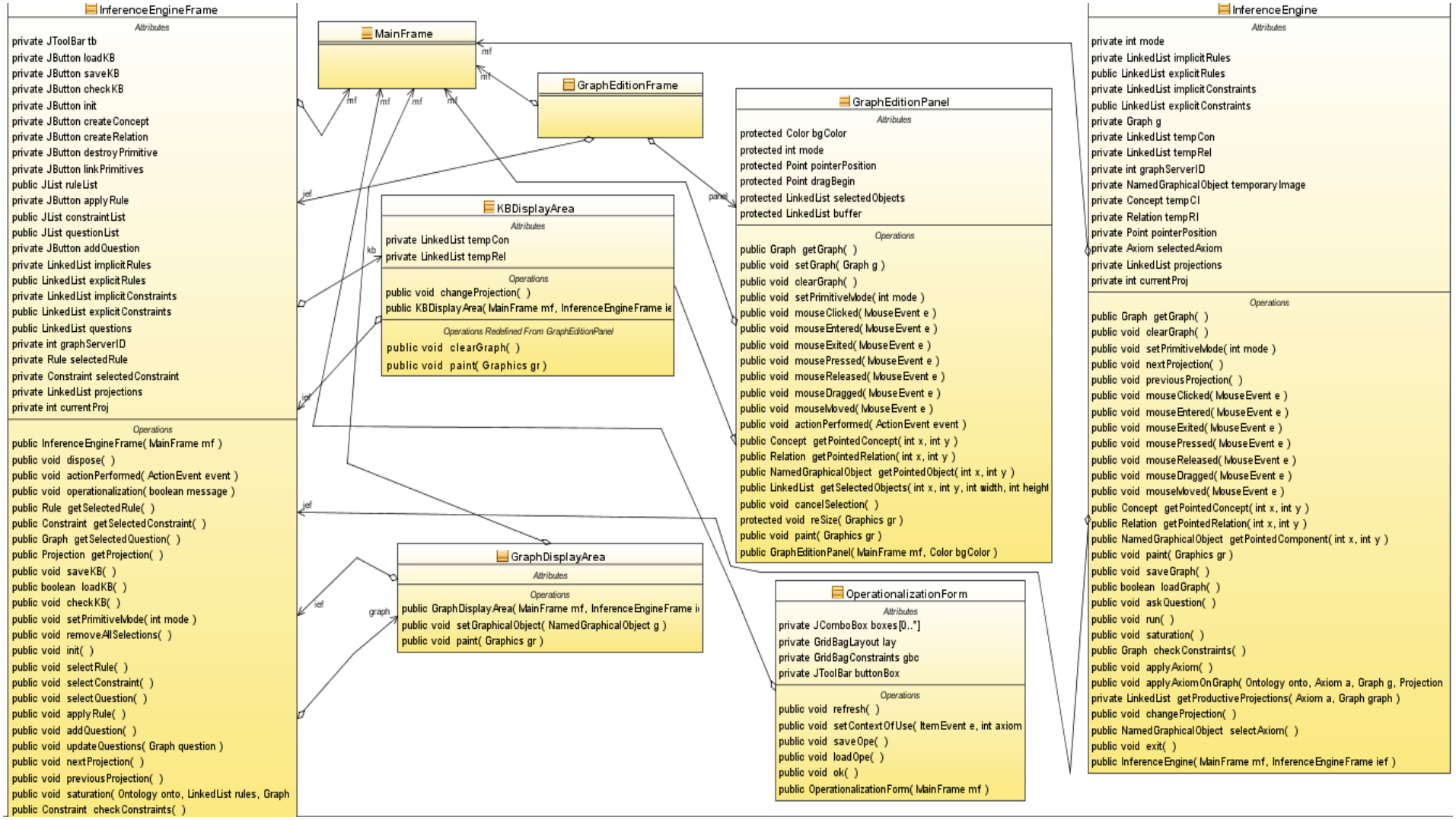


Figura 3-42 Ferestre grafice pentru motorul de inferența, operaționalizarea axiomelor, contextul de utilizare

Aplicația este **documentată** (un exemplu în figura 3-43, alte exemple în Anexa II).

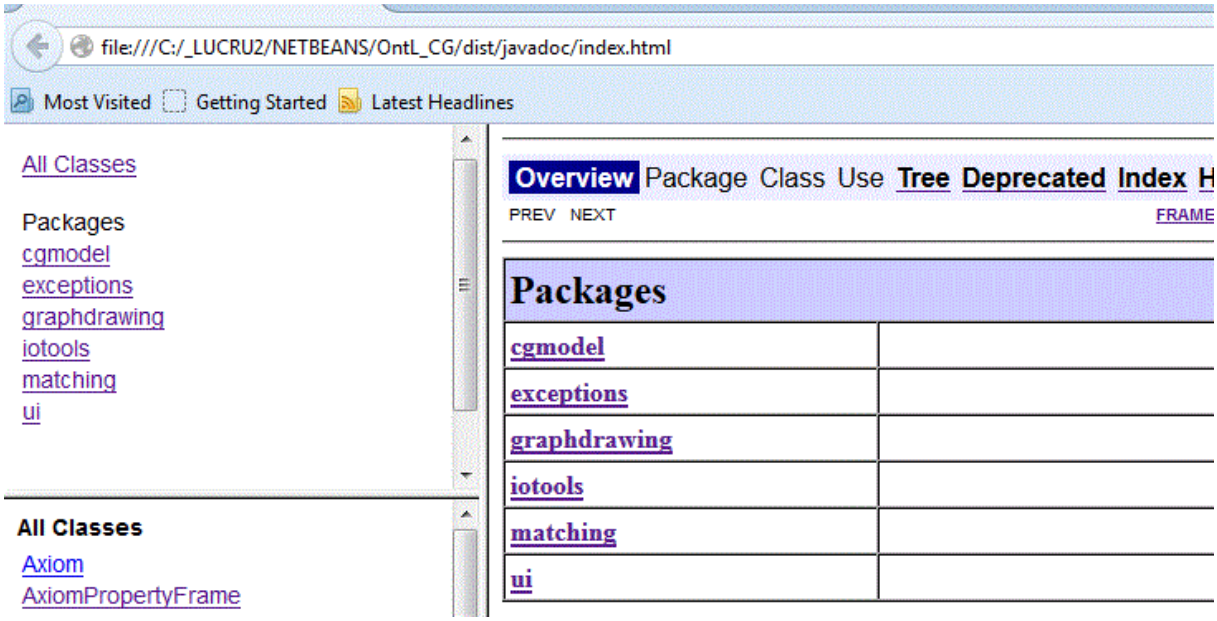


Figura 3-43 Documentarea aplicației OntL_CG

3.7 STUDIU DE CAZ

Instrumentul OntL_CG poate fi utilizat ca și componentă a unui sistem autor de proiectare a unui sistem inteligent de instruire, deoarece permite:

- Crearea sau utilizarea unei ontologii existente într-un sistem de proiectare pedagogică, bazat pe cunoștințe.
- Operaționalizarea ontologiei domeniului și specificarea grafică a axiomelor domeniului, fie că este vorba de ontologia domeniului în care se realizează instruirea, fie ontologia sudentului, fie ontologia despre strategiile pedagogice.
- Validarea unei baze de cunoștințe (în care cunoștințele ontologice sunt folosite pentru validarea bazei de cunoștințe în raport cu semantica domeniului).
- Scenariile inferențiale automate în care cunoștințele ontologice sunt folosite pentru obținerea de noi cunoștințe într-un caz dat, fără intervenția utilizatorului, este un aspect important. În această situație (ca în cazul unui sistem expert), inferențele automate conduc la obținerea de cunoștințe conforme cu semantica domeniului, nemaifiind necesară validarea. (Dacă ontologia este coerentă, cunoștințele produse nu pot fi contradictorii).

Studiu de caz:

Profesorul, specialist în domeniul de instruire și pedagogie, dar nespecialist în informatică sau în reprezentarea cunoștințelor dorește să creeze un document pedagogic pentru studenții săi, pe baza unor materiale (LO[39]) pedagogice existente.

Materialele de care dispune sunt de tipurile specificate în ontologia din figura 3-44.

[39] LO – obiecte instrucționale (engl., “learning object”)

Observație: Unui material pedagogic (resurse pedagogice) i se poate atribui un rol dinamic, nu static. Ne propunem doar să exemplificăm posibilitățile oferite de către aplicația OntL_CG, nu analizăm aici alte aspecte.

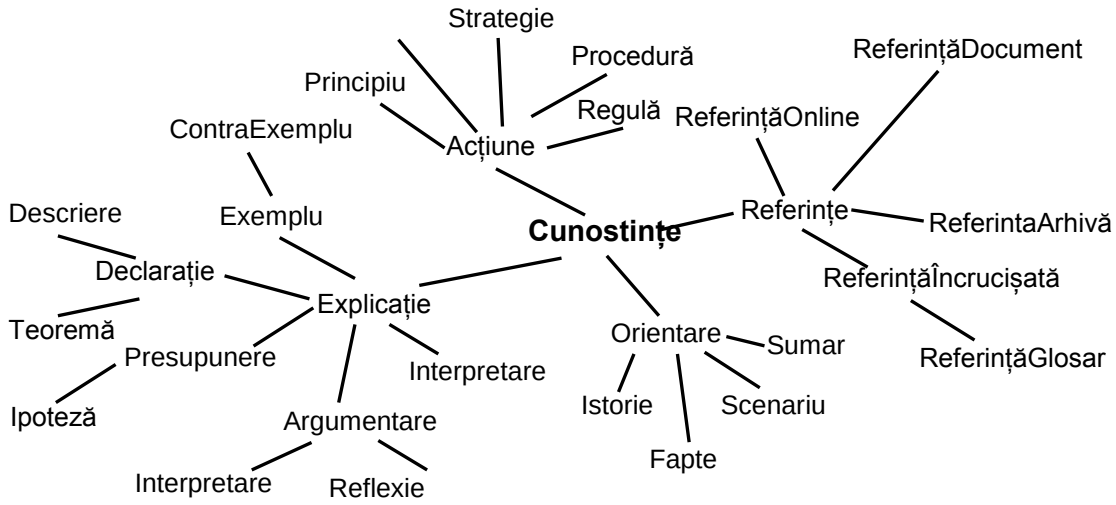


Figura 3-44 Ontologie pedagogică (Ulrich, Leidig)

Ierarhia tipurilor de concepte din ontologie (din ontologia prezentată în figura anterioară):

- Obiect instrucțional
  - Auxiliar
    - Evidență
      - Demonstrație
      - Dovadă
    - Explicație
      - Concluzie
      - Introducere
      - Remarcă
    - Ilustrare
      - ContraExemplu
      - Exemplu
    - Interactivitate
      - Exercițiu
      - Explicație
      - Invitație
      - ProblemăReală
  - Fundamental
    - Definiție
    - Fapte
    - Legi
      - LegiFundamentale
      - Teoreme
    - Proces
      - Politică
        - Procedură

De exemplu, **strategia** pe care dorește să o aplice este: ***Resursele fundamentale au prioritate asupra celor auxiliare.***

Folosind OntL_CG, va trebui să adauge această relație la cele existente (folosind editorul grafic) și să specifice axioma corespunzătoare (o regulă cu ipoteza vidă, deci o constrângere pozitivă).

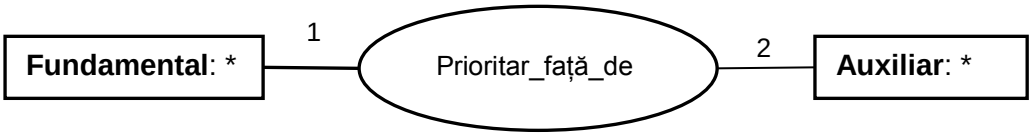


Figura 3-45 Axioma corespunzătoare strategiei specificate

Faptele care exprimă relația de prioritate între resursele (instanțele) B și C, D și E, G și F, G și A pot exista sau nu în baza de cunoștințe. Dacă nu există, pedagogul le specifică prin grafuri conceptuale și le adaugă la baza de fapte.

Axioma care exprimă strategia aleasă poate fi operaționalizată într-un context inferențial explicit. Pornind de la faptele existente (subsumarea conceptelor și relația de prioritate dintre B și C, etc.), se vor deduce noi fapte, care vor fi adăugate la baza de cunoștințe (figura 3-45). Dacă baza de cunoștințe era coerentă, faptele adăugate nu sunt contradictorii, deci nici nu este necesară validarea).

Scenariul de validare poate fi unul implicit sau explicit.

Exemplul din acest studiu de caz demonstrează cum poate fi susținută activitatea de modelare pedagogică: profesorul nu trebuie să apeleze la inginerul de cunoștințe.

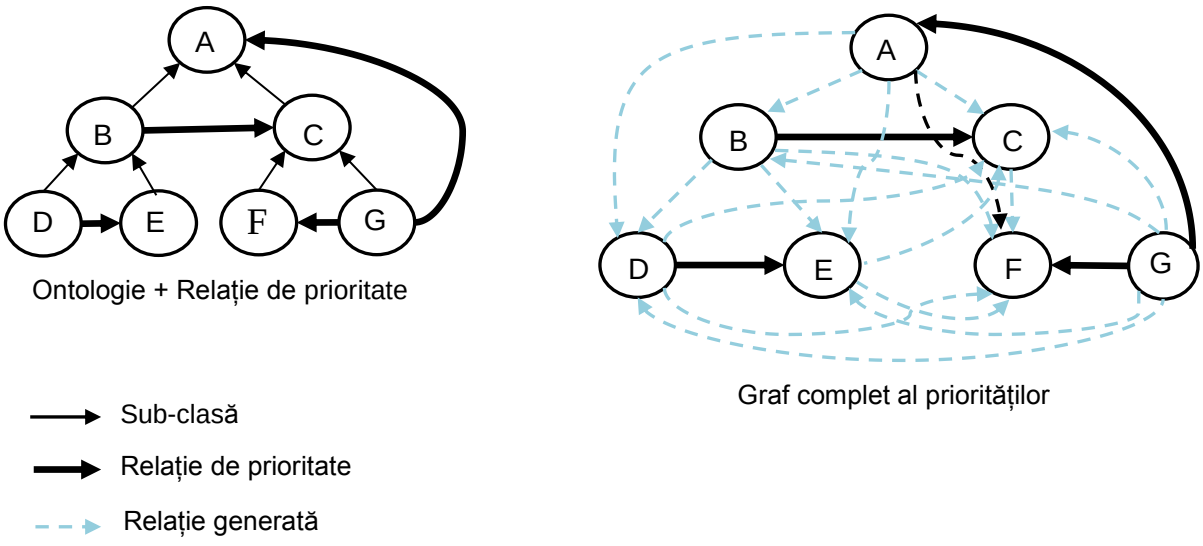


Figura 3-46 Deducerea unor noi relații pe baza tipului semantic specificat în ontologie și a unei strategii pedagogice de compunere a documentului virtual

3.8 CONCLUZII ŞI CONTRIBUŢII

Ingineria ontologică are ca scop diversificarea aplicațiilor bazate pe cunoștințe și găsirea unor modalități de reprezentare a cunoștințelor *independente de aplicațiile care le vor utiliza în raționamente*. Ontologiile vor trebui să integreze cunoștințele terminologice ale domeniului și să exprime semantica acestuia, păstrând independența față de utilizarea lor operațională în cadrul unui SBC. Practic, neutralitatea vizată a ontologiilor față de aplicație este în opoziție față de utilizarea lor practică într-o aplicație.

Soluțiile propuse în acest capitol al lucrării permit punerea în acord a celor două obiective aparent contradictorii. În acest sens, am propus un limbaj (numit OntL_CG) de reprezentare a ontologiei unui domeniu. Limbajul (grafic) se bazează pe grafurile conceptuale și pe paradigma entitate-relație și permite unui utilizator nespecialist (în informatică sau reprezentarea cunoștințelor) să construiască o ontologie puternic formalizată (care nu include doar o taxonomie a conceptelor și a relațiilor domeniului). Utilizatorul nespecialist poate specifica în manieră grafică orice axiomă a domeniului.

A fost propusă, deasemenea, o metodă generală de operaționalizare a ontologiei domeniului (reprezentată în limbajul propus), indiferent de scopul în care aceasta va fi utilizată. Metoda se bazează pe un scenariu de utilizare în care, pentru fiecare axiomă trebuie specificat contextul de utilizare al acesteia. Pe baza scenariului stabilit de către utilizator, se generează automat forma operațională a ontologiei.

Am evidențiat faptul că pentru operaționalizarea ontologiei poate fi utilizat orice limbaj (ca de exemplu, logica predicatelor de ordinul I), însă am folosit limbajul grafurilor conceptuale ca limbaj operațional, pentru operaționalizarea axiomelor.

Așa cum s-a evidențiat în Anexa I (în prezentarea informală a grafurilor conceptuale ca model de reprezentare a cunoștințelor) și în Anexa III (formalismul acestora), există două abordări în privința acestui limbaj privit ca limbaj de reprezentare operațional: (a) prima, în care grafurile conceptuale sunt folosite doar ca notație grafică, raționamentul fiind unul logic și (b) cea în care în sistemul formal al grafurilor conceptuale raţionamentul se bazează pe operaţii interne (practic, un morfism al grafurilor) care păstrează corectitudinea şi completitudinea în raport cu logica predicatelor de ordinul I. A doua abordare este și cea aleasă în această lucrare.

Pentru operaționalizarea axiomelor a fost folosit modelul grafurilor conceptuale simple (SCG) extins cu reguli și constrângeri (propus de Chein & Mugnier [175].

Aplicația numită OntL_CG (ca și limbajul propus). Aplicația translatează automat o reprezentare ontologică a cunoașterii domeniului într-o reprezentare operațională, conform scenariului de utilizare definit de către utilizatorul nespecialist, demonstrând modul de aplicare a contribuțiilor teoretice din acest capitol. Este o aplicație client, implementată în Java, care comunică cu serverul CoGITaNT prin fișiere .xml.

Studiul de caz ilustrează o modalitate prin care instrumentul OntL_CG susține activitatea de proiectare pedagogică.

"Pedagogy is perhaps the area of greatest divergence across intelligent tutoring systems."

"Pedagogia este, probabil, zona cu cele mai mari divergenţe în ceea ce priveşte tutorialele inteligente."

(Tom Murray)

4. CONTRIBUŢII LA MODELAREA STRATEGIILOR DE INSTRUIRE ADAPTIVE

4.1 PROPUNEREA UNUI CADRU STRUCTURAT DE MODELARE A CUNOAŞTERII PEDAGOGICE

4.1.1 TERMENI DE DESCRIERE A CUNOAŞTERII PEDAGOGICE

Unul dintre scopurile acestei teze este acela de a propune un cadru de modelare a cunoaşterii pedagogice, motivată fiind şi de afirmaţia lui Tom Murray [136] *"Pedagogy is perhaps the area of greatest divergence across intelligent tutoring systems"* ("Pedagogia este, probabil, zona cu cele mai mari divergenţe în ceea ce priveşte tutorialele inteligente").

Demersul de înţelegere şi de stabilire a termenilor universului de discurs - expertiza pedagogică - s-a dovedit a fi extrem de dificil, deoarece:

- Domeniul vizat este unul interdisciplinar, aşa cum am demonstrat în capitolul 1;
- Termenii folosiţi de către diverse comunităţi (comunitatea psihologilor, a pedagogilor, a proiectanţilor sistemelor de instruire, etc.) sau, mai mult, chiar termenii acceptaţi în cadrul aceleaşi comunităţi – sunt extrem de diferiţi;
- Deşi arhitectura de (cel puţin) "triplu expert[40]" a unui ITS este unanim acceptată, cunoştinţele pe care se bazează "funcţionarea" şi deciziile *modulului expert pedagogic* - numit şi *modul de predare* ("*teaching modul*"), *modul tutorial* ("*tutoring modul*") sau *modul instrucţional* ("*instructional modul*") - sunt greu de izolat de cunoaşterea celorlalte module; în plus, "inteligenţa" modulului pedagogic trebuie să exploateze tocmai natura cunoaşterii obţinute prin interconectarea celorlalte două arii.

Barnes [176] afirma: *"Talking about teaching[41] is analogous to describing a tapestry that has many threads of different colours woven into complicated textures and patterns. One can*

[40] "Expert în domeniu", "Expert în cunoaşterea subiecţilor instruirii", "Expert în pedagogie", eventual "Expert în comunicare"

[41] "Predare" (teaching), cu sens de instruire

remove individual threads and examine them separately, but one cannot appreciate the complexity of the tapestry without seeing how the threads are interwoven to create the whole cloth". ("A descrie activitatea de predare este analog cu a descrie o tapiserie formată din multe fire de diferite culori, ţesute într-o textură complicată, cu modele. O dată ce scoatem fire pentru a le examina separat, nemaivăzând cum sunt ţesute pentru a crea întregul tablou, pierdem din complexitatea tapiseriei").

O problemă importantă a fost generată de substantivele "pedagogie[42]" şi "didactică[43]", ale căror semantici le-am întâlnit în raporturi de opoziţie, de egalitate (de acoperire sau quasi-sinonimie) sau de incluziune; cu toate acestea, adjectivele "pedagogic" şi "didactic" sunt folosite cu aceeaşi conotaţie. Nici definiţiile oferite de dicţionarele de specialitate pentru termenii pedagogie[44] şi didactică[45], nici opiniile cercetătorilor în educaţie nu sunt convergente (în cea mai mare parte sunt, eventual, complementare, în sensul că tratează doar anumite aspecte ale unei probleme). Cu toate acestea, pot fi evidenţiate două abordări: abordarea *anglo-americană* (în care apare doar termenul de pedagogie) şi abordarea franco-germană (în care apar ambii termeni, atât cel de pedagogie, cât şi cel de didactică[46]).

Principalele caracteristici ale modelelor anglo-americane sunt:

- Scopul urmărit de aceste modele este identificarea şi determinarea factorilor care influenţează predarea şi învăţarea, pentru a dezvolta o înţelegere a proceselor implicate în predare şi învăţare.
- Sunt modele în care se acordă o mai mică importanţă fundamentelor teoretice (astfel, termenul *didactică* – bazat pe psihologia educaţională – *lipseşte*) şi o mai mare importanţă practicii. De aceea, unii cercetători le consideră – în mare măsură - "empirice".
- Cunoştinţele pedagogice nu sunt statice, sunt în continuă schimbare şi dezvoltare, fiind determinate de componente ca interacţiunea dintre domeniul în care se realizează instruirea, subiecţii instruirii şi experienţa profesională a educatorului.

Principalele caracteristici ale modelelor franco-germane sunt:

- Gândirea filozofică, teoretizarea şi construirea modelelor teoretice;
- În aceste modele apar atât termenul de *pedagogie*, cât şi cel de *didactică*.

---

[42] În limba greacă: "*pais"*, *"paidos"* – copil; "*agoge"* - conducere, educaţie; "*paidea"* - învăţământ, educaţie; *"paidagogos"* - îndrumător de copii, pedagog

[43] În limba greacă: "*didaktos*" – a spune: "*didaskein, didak-*" – a preda, a educa

[44] Pedagogia este (conf. dicţionarului):

- arta sau ştiinţa de a fi profesor şi vizează, în general, strategiile instruirii sau un stil de instruire;
- ştiinţă care se ocupă cu metodele de educaţie şi de instruire.

[45] Didactica este (conf. dicţionarului):

- disciplină pedagogică care studiază principiile şi metodele de învăţământ;
- ramură a pedagogiei care se ocupă cu problemele învăţământului şi ale instruirii;
- parte a pedagogiei care se ocupă cu principiile şi metodele predării materiilor de învăţământ şi cu organizarea învăţământului.
- pedagogia unei anumite discipline

[46] Termenul "didactică" a fost introdus de către francezi.

- Pedagogia şi didactica sunt privite în raporturi diferite, fiind încadrate în aşa-numitul triunghi pedagogic (figura 4-1).

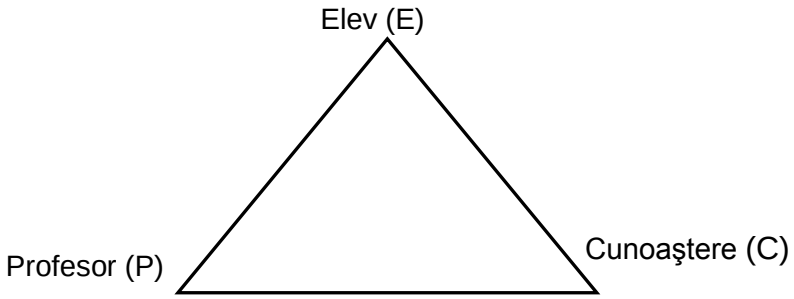


Figura 4-1 Triunghiul pedagogic propus de Houssaye

În modelele franco-germane, pentru raporturile dintre pedagogie şi didactică, studiate în cadrul triunghiului pedagogic propus de Houssaye[47], au fost propuse următoarele relaţii:

- *P ≠ D*
  Pedagogia încearcă să raţionalizeze şi să optimizeze procesele de învăţare.
  Didactica încearcă să asigure transmisia optimală a cunoştinţelor definite prin obiectivele şi conţinutul fiecărei discipline.

- *P = axa P-E; D = axele E-C şi P-C*
  Pedagogia reprezintă relaţia de instruire (ghidare sau însoţire) stabilită între profesor şi elev, concentrându-se asupra aspectelor relaţionale ale învăţării.
  Didactica priveşte accesul la cunoaştere al elevului (o activitate corelată de învăţare-predare), concentrându-se asupra dimensiunii cognitive a activităţii educative.

- *P ⊂ D*
  Pedagogia reprezintă doar componenta aplicată a didacticii.

- *P = axele P-E, E-C, P-C*
  După Filloux, pedagogia, iniţial artă, devine ştiinţa de a educa; finalitatea pedagogiei este aceea de a determina obiectivele şi metodele (strategii şi tehnici) care caracterizează transmiterea socială sau interindividuală a cunoaşterii. Pedagogia încearcă să dezvolte un corp de cunoştinţe asupra problematicii psihologice în raportul profesor-elev, aceleaşi ca în raportul elev-conţinut.

  Acelaşi punct de vedere a fost adoptat de Legendre, pentru care relaţia pedagogică subsumează didactica, predarea şi învăţarea.

- *D = axele E-C, P-C, P-E*
  Didactica studiază interacţiunile care se stabilesc într-o situaţie de predare-învăţare între profesor, elev, cunoaştere.

[47] Triunghiul plasează în vârfuri termenii *profesor*, *elev* şi *cunoaştere*. Aceşti termeni sunt generici; de exemplu, termenul de profesor vizează componente unui sistem educaţional şi rolul acestora (de meditare ghidată, de învăţare prin descoperire, etc.), iar termenul “cunoaştere” vizează atât cunoştinţele unei anumite discipline, cât şi modalităţile de transformare a acestora în vederea atingerii obiectivului urmărit de profesor.

Aşa cum se observă, chiar şi în cadrul aceleaşi comunităţi şi al aceleiaşi abordări, specialiştii folosesc cei doi termeni în mod diferit.

4.1.2 STRUCTURAREA CUNOAȘTERII PEDAGOGICE

Scopul studiului prezentat succint în secțiunea anterioară a fost acela de a identifica punctele comune, principiile de bază şi "echivalenţele" posibile între termenii folosiţi în comunitatea cercetătorilor din ştiinţele educaţiei. Consider că este necesară o structurare a cunoaşterii pedagogice pe mai multe niveluri de abstractizare, aşa cum se prezintă în figura 4-2.

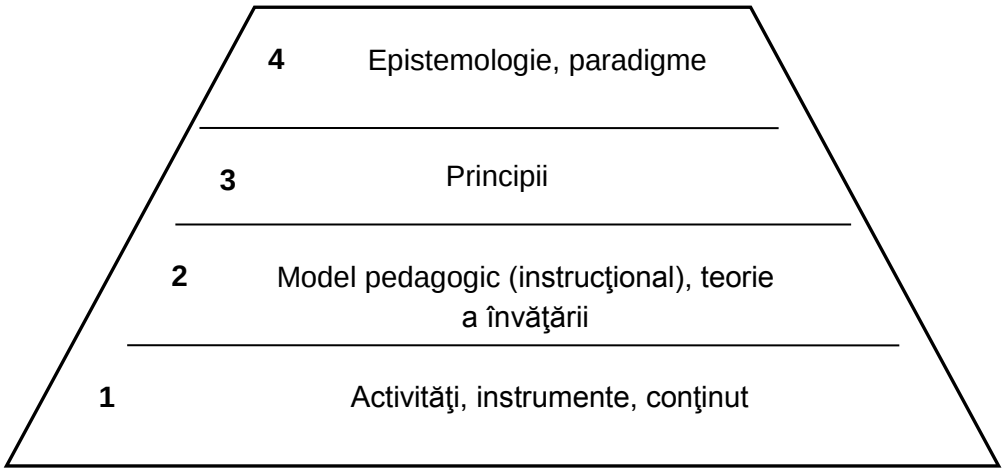


Figura 4-2 Cunoaşterea pedagogică, la diferite niveluri de abstractizare

Nivelul 4 (epistemologie, paradigme), cel mai înalt nivel de abstractizare, vizează paradigmele din epistemologie, teoria predării, a învăţării, a comunicării sau a proiectării instruirii. Aceste paradigme descriptive (considerate separat sau combinate) stabilesc (implicit sau explicit) o serie de principii pedagogice. Exemple de astfel de principii (situate la *nivelul 3*) au fost prezentate în capitolul 1, în sinteza din figura 1.5. (ancorarea noilor concepte în structurile cognitve ale elvului, etc.). Principiile determină conceperea modelelor instrucţionale (*nivelul 2*) şi strategiile de organizare a procesului de învăţare. Conform lui Merrill [14], "principiile sunt implementate printr-un program" şi "un program se bazează pe o serie de principii". Cunoştinţele care vor fi prezentate elevului, modul de structurare a acestora, activităţile pe care le va desfăşura, apar la *nivelul 1*. Metadatele oferite de către standardul LOM conţin informaţii pedagogice insuficiente, ca, de exemplu, un obiect instrucţional ("learning object") ar putea fi utilizat în învăţarea prin colaborare sau în învăţarea prin rezolvare de probleme?

Aceeaşi idee de structurare a cunoaşterii pedagogice pe diferite niveluri de abstratizare o regăsim şi în lucrările cercetătorilor Reigeluth şi Merrill. Reigeluth [177] identifică două tipuri de "metode instrucţionale": ***de bază*** (*fundamentale*) şi *variabile*.

Merrill le numeşte "*Primul principiu al instruirii*[48]", respectiv "*Programs*[49]*, practices*[50]". O teorie poate specifica atât principii, cât şi practici. În ambele abordări, *"Primul principiu al instruirii"*

[48] Conform *"Primului principiu al instruirii"*, un principiu de bază (metodă) este o relaţie adevărată în condiţiile adecvate unui *"program"* sau *"practice"*.

[49] Un *"Program"* este o abordare care constă într-o mulţime de "practici" (aplicări) prescrise (stabilite).

(conf. Merrill) (*"de bază",* conf. Reigeluth) este orientat mai mult către *proiectarea instrucţională (instructional design)*, al doilea principiu (*"program, practice"*, conf. Merrill sau *"variabil"*, conf. Reigeluth) este orientat mai mult către *învăţare*.

Studiile prezentate anterior m-au condus la propunerea de structurare a cunoaşterii pedagogice - utilizate în proiectarea sistemelor de instruire - pe *două niveluri* (figura 4-3):

- **Cunoaşterea pedagogică generală (macro-nivel)**;
- **Cunoaşterea pedagogică de conţinut (micro-nivel)**;

CUNOAȘTEREA PEDAGOGICĂ GENERALĂ

Cunoaşterea pedagogică generală, *la macro-nivel*, este independentă de domeniu şi furnizează principiile propuse prin teoriile de predare, învăţare şi proiectare a instruirii. Există o corespondenţă între paradigmele teoretice şi strategiile pedagogice adecvate.

Susţinerea acestei afirmaţii a fost evidenţiată deja în capitolul 1.2. şi în tabelul 1.5. (influenţa teoriilor asupra strategiilor de instruire), aşa că, aici, vom mai da un singur exemplu.

De exemplu, în **domeniul cognitiv**, există patru *tipuri*[51] *de învăţare* şi pentru fiecare tip de învăţare (rezultat vizat: obiectiv cognitiv) se recomandă metode de instruire diferite.

Tabelul 4-1 Tipul învăţării şi termenii folosiţi de cercetători

Tipul învăţării	Termeni folosiţi	Cercetător
Memorarea Informaţiei	Cunoaştere (*"knowledge"*)	Bloom
	Amintirea celor discutate (*"remember verbatim"*)	Merrill
	Învăţarea pe de rost, memorarea (*"rote learning"*)	Ausubel
	Un aspect al *"*remember verbatim"	Gagné
Înţelegerea relaţiilor	Cuprindere *("comprehension"*)	Bloom
	Amintirea parafrazaă *("remember paraphrased"*)	Merrill
	Învăţarea semnificaţiei verbale *("meaningfull verbal learning")*	Ausubel
	Alt aspect al *"remember verbatim"*	Gagné
Aplicarea aptitudinilor, deprinderilor (*"skills"*)	Aplicare *("application"*)	Bloom
	Folosirea generalităţii *("use-a-generality"*)	Merrill
	Deprinderi intelectuale *("intellectual skills"*)	Gagné
Aplicarea aptitudinilor, deprinderilor generale	Analiză, sinteză, evaluare *("analysis, synthesis, evaluation"*)	Bloom
	Găsirea generalităţii *("find-a-generality"*)	Merrill
	Strategii cognitive *("cognitive strategies"*)	Gagné

[50] Termenul *"Practice"* desemnează o activitate instrucţională specifică.

[51] Tipurile de învăţare şi termenii corespunzători acestora în teoriile lui Bloom, Ausubel, Gagné, sunt prezentate în tabelul 4.1.

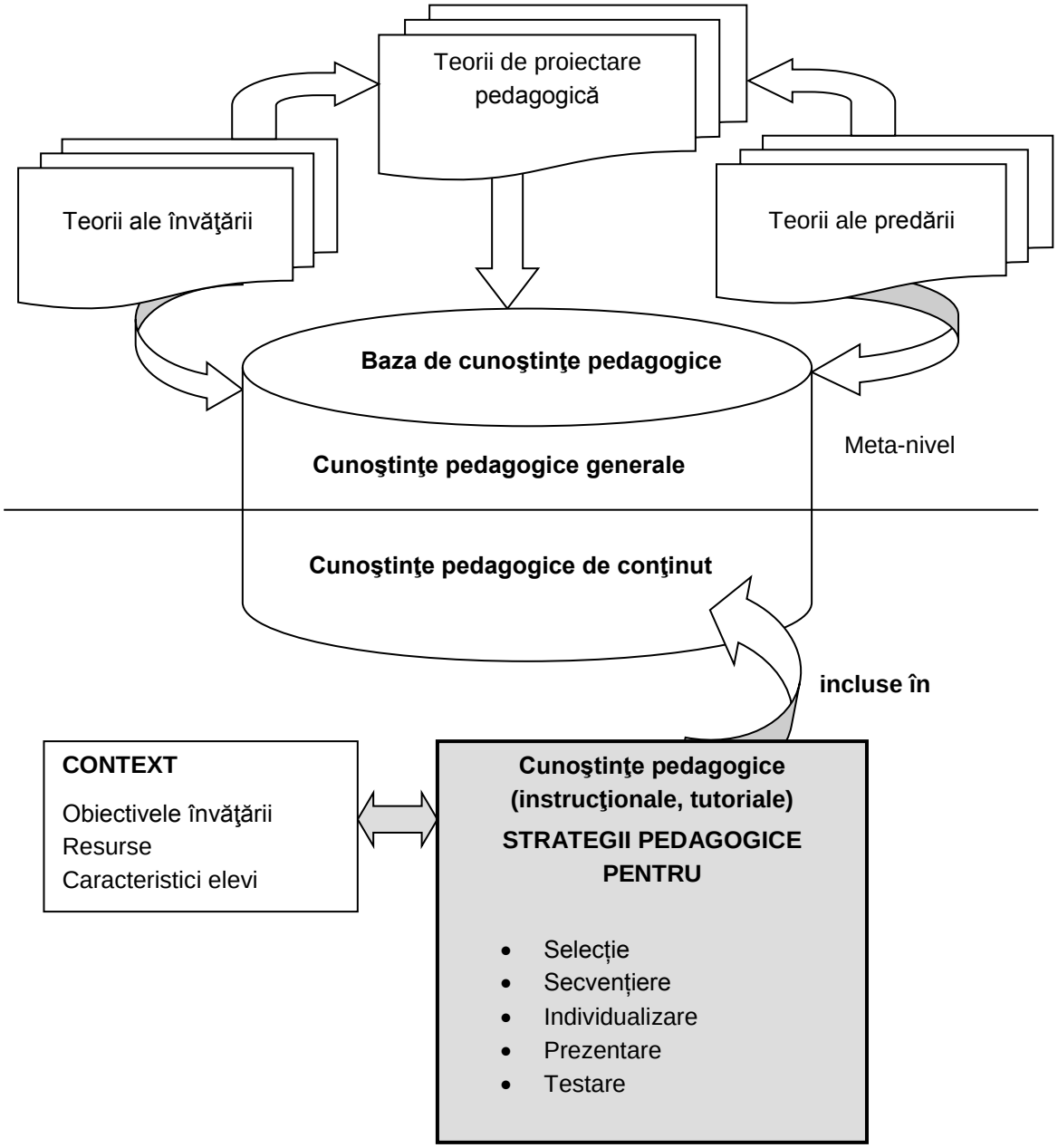


Figura 4-3 Cunoaştere *pedagogică generală* şi cunoaştere *pedagogică contextuală*

Tipurile variate de *conţinut specific domeniului*, cum ar fi concepte, proceduri, principii, pot fi dobândite prin unul din primele trei tipuri de învăţare: un *concept poate fi memorat* (dându-se o definiţie sau un exemplu al conceptului), *poate fi înţeles* (punându-l în relaţie cu alte cunoştinţe pe care le are elevul) sau *poate fi aplicat* (instanţele pot fi clasificate ca exemple sau contra-exemple ale conceptului). Al patrulea tip de învăţare este *independent de domeniu* şi necesită, în general, mai mult timp pentru a putea fi dobândit. Până în prezent, în instruirea asistată, eforturile au fost canalizate mai mult către memorarea informaţiei şi către aplicarea aptitudinilor (clasificarea conceptelor şi utilizarea procedurilor), neglijându-se înţelegerea relaţiilor şi aplicarea aptitudinilor (deprinderilor) generale.

CUNOAȘTEREA PEDAGOGICĂ DE CONȚINUT

Pentru cunoașterea de ***micro-nivel***, propunem termenul "***cunoaştere pedagogică de conţinut***", sinonim cu termenii "*didactică*" (din modelele franco-germane) şi cu termenii anglo-americani "*teaching*", "*curriculum*[52]" sau "*cunoştinţe pedagogice de conţinut*[53]". La acest nivel, cunoştinţele pedagogice sunt ***cunoştinţe contextuale***, fiind folosite de către profesor/sistem de instruire într-o situaţie cu anumite caracteristici (disciplina, capitolul sau lecţia; elevul sau grupul de elevi; obiectivele instrucționale vizate).

În această lucrare tratăm ***aspecte ale reprezentării cunoașterii pedagogice de conținut***, la micro-nivel.

Cunoştinţele pedagogice de conţinut[54] (figura 4-4), sunt cunoştinţe *dinamice*, care, pe baza principiilor de cunoaştere pedagogică generală, "organizează" (*CUM?*) cunoştinţele domeniului în cadrul unei discipline, capitol sau lecţii, în funcţie de *obiectivele* urmărite şi de viziunea profesorului asupra posibilelor structurări ale conceptelor ţinând seama de relaţiile existente (*CE?*) şi de publicul cărora se adresează (*CUI?*).

Includ atât cunoştinţe conceptuale, cât și procedurale, un repertoriu de tehnici variate sau activităţi (care ţin seama de diferitele stiluri de învăţare) pentru apreciere şi evaluare, cunoştinţe despre varietatea resurselor care vor fi folosite.

Strategiile instrucționale (pedagogice) de conținut vizează numeroase aspecte (figura 4-3, figura 4-5), cum sunt cele legate de selectarea conceptelor, ordinea în care vor fi prezentate (diverse trasee de instruire), ce fel de materiale (resurse) vor susține activitatea, cum se vor testa cunoștințele studentului sau cum va fi adaptată instruirea la stilul cognitiv al studentului?

La acest nivel, ***cunoştinţele pedagogice sunt cunoştinţe contextuale***.

Definiția 4-1 Cunoștințele pedagogice de conținut (CPC) includ **cunoașterea declarativă** și **cunoașterea procedurală** folosită pentru organizarea și secvențierea conținutului, pentru specificarea activităților de învățare și deciziile luate pentru prezentarea

---

[52] Curriculum-ul (din latinescul "curriculum" - cursă, alergare), conf. lui L. D'Hainaut, cuprinde:
- Obiectivele specifice unui domeniu (nivel de învăţământ, profil, disciplină) sau activitate educativă;
- Conţinutul necesar pentru realizarea obiectivelor stabilite;
- Condiţiile de realizare (metode, mijloace), programarea şi organizarea situaţiilor de instruire şi educare;
- Evaluarea rezultatelor.

[53] Cunoştinţele pedagogice de conţinut, conf. lui Shulman, includ:
- Modurile de reprezentare şi formulare a subiectului predat, astfel încât să poată fi înţeles de către elev (pentru fiecare noţiune - cele mai bune forme de reprezentare a ideilor, cele mai puternice analogii, exemple, demonstraţii şi explicaţii);
- Alternative variate ale reprezentării şi formulării, obţinute atât din cercetările teoretice, cât şi din practică;
- Cunoştinţele profesorului despre ceea ce face ca o noţiune să fie dificil sau uşor de învăţat;
- Modalităţile de organizare şi reorganizare a activităţilor de predare/învăţare.

[54] Nu trebuie confundate cu cunoştinţele domeniului în care se realizează instruirea.

conținutului și desfășurarea activităților într-un context dat (într-o anumită situație, pentru un anumit elev).

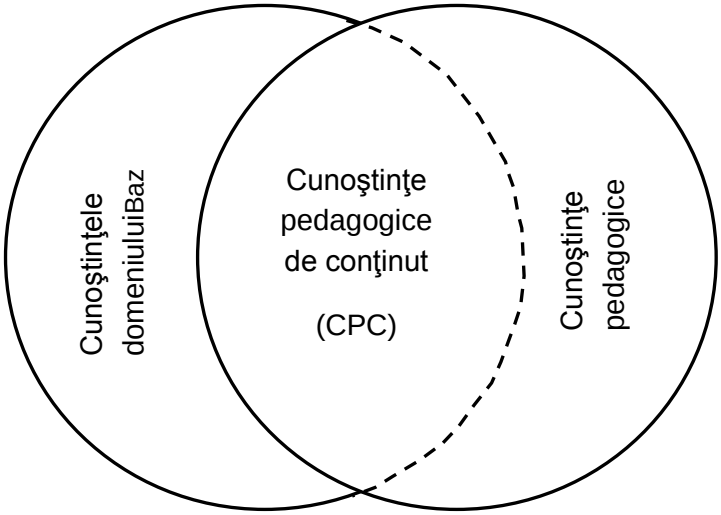


Figura 4-4 Cunoştinţe pedagogice de conţinut

Definiția 4-2 Strategia pedagogică de conținut reprezintă componenta procedurală a cunoașterii pedagogice de conținut.

Strategiile sunt reprezentate prin proceduri, planuri [178], [179] constrângeri, reguli, pattern-uri [180]. Așa cum am evidențiat în secțiunea 1.3.1, *majoritatea sistemelor autor se limitează la implementarea unei anumite strategii, predefinite*. Puține dintre sistemele autor oferă flexibilitate în alegerea *în mod explicit a unei strategii pedagogice* (exemple de astfel de sisteme – prezentate în 1.3.1 sunt: EON, COCA, REDEEM, IDE, GTE, IRIS).

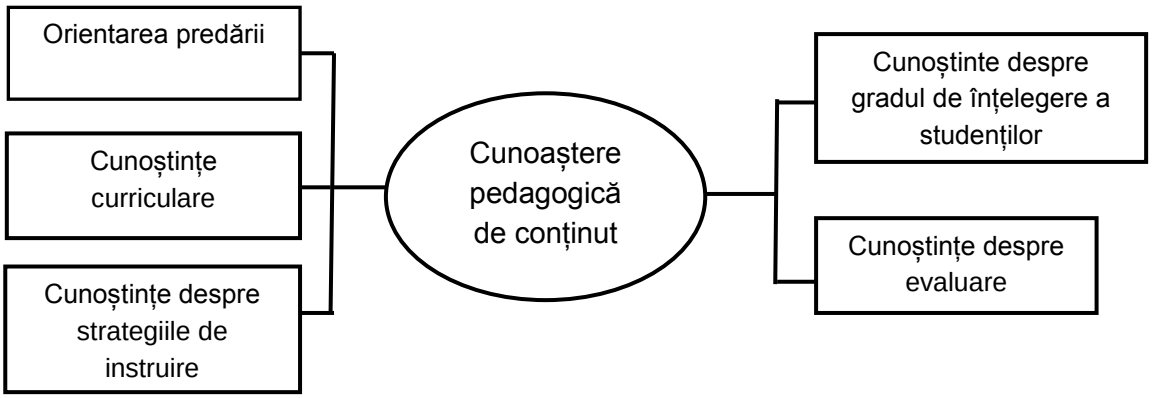


Figura 4-5 Componente ale cunoașterii pedagogice de conținut

Cunoaşterea despre strategiile şi tacticile pedagogice de instruire trebuie reprezentată **separat** de *cunoaşterea domeniului în care se realizează instruirea*. Evident, cunoştinţele domeniului, cunoștințele despre stilul de învățare al studentului şi cunoştinţele tutoriale nu sunt independente, dar această separare (între "formă" şi "conţinut"), însoţită de o *reprezentare explicită a cunoştinţelor despre strategiile pedagogice și contextul de aplicare a acestora*, poate remedia într-o măsură apreciabilă ceea ce li se "reproşează" tutorialelor inteligente, şi anume, "*pedagogia implicită*".

Definiția 4-3 Strategia pedagogică este o **problemă de planificare**, un proces prin care se decide ce acțiune va fi executată pentru a atinge un anumit obiectiv.

Definiția 4-4 Numim **plan** mulțimea regulilor de decizie pe care se bazează strategia pedagogică.

4.2 APLICAREA METODEI MIIO PENTRU ONTOLOGIA PROPUSĂ

Pentru dezvoltarea (în subcapitolele 4.3, 4.4) unei ontologii a strategiilor pedagogice vom aplica metoda MIIO, propusă în capitolul 2; metoda se bazează pe:

a) Considerarea *ciclului de viaţă al ontologiilor* (propuse prin On-To-Knowledge şi METHONTOLOGY);
b) Strategiile de *identificare a conceptelor (propusă de Uschold&King*);
c) Întrebările de *competenţă (din Grüninger&Fox)*;
d) Criteriile de evaluare din *METHONTOLOGY şi Goméz-Pérez*.

Metoda MIIO constă în cinci faze, fiecare dintre acestea cuprinzând, la rândul lor, mai multe etape. Vom evidenția doar fazele și răspunsurile pentru problemele de la fiecare nivel.

I. Studiul fezabilităţii şi al mediului: ontologia va putea fi utilizată pe orice platformă, externă sau internă une aplicații (propuse în METHONTOLOGY), este posibil și oportun să avem o astfel de ontologie (propuse On-To-Knowledge).
II. Modelarea ontologică (specificare, conceptualizare, formalizare şi implementare):
    - Specificarea:
        - Furnizarea unei baze de cunoștințe unui sistem autor de dezvoltare, asistarea pedagogului în procesul de modelare a strategiilor pedagogice sau de alegere a acestora;
        - Concepție bottom-up (câteva concepte și relații, doar ca exemplu);
        - Surse de informare: elementele prezentate în capitolul 1 și în 4.1;
        - Întrebările de competență (Gruninger&Fox): ce tipuri de activăți sunt asociate cu cea de predare/testare, ce materiale sunt mai ”potrivite” pentru un anumit tip de student, etc.?
        - Pe baza întrebărilor de competență și a scenariilor de utilizare se selectează termenii (material pedagogic – LO, activitate de predare, activitate de evaluare, student, etc.);
    - Conceptualizare și documentarea ontologiei;
    - Formalizare – alegerea unui instrument și a unui limbaj (în funcție de expresivitatea acestuia și de capacitățile inferențiale) – nu e cazul dacă folosim OntLCG;
    - Implementare (în sensul de transcriere într-un limbaj operațional) – nu e cazul dacă folosim OntLCG;
    - Documentarea ontologiei formale;
III. Operaţionalizarea (în sensul utilizării) ontologiei;
IV. Evaluarea, constând în verificarea şi validarea ontologiei şi testarea aplicaţiei care foloseşte ontologia.
    - Se va realiza conform criteriilor din *METHONTOLOGY* și Gomez-Perez. În cazul OntL_CG, testarea se realizează prin teste interne (parte din teste – propuse de Gomez-Perez).
V. Documentarea.

Dacă aplicăm propunerile din capitolul 3, implementate în OntL_CG, multe dintre elementele "clasice" vor putea fi "neglijate": scenariile de utilizare a ontologiei, contextul de utilizare, testele intrene de validare, etc.

4.3 CONTRIBUȚII LA MODELAREA CONCEPTUALĂ A STRATEGIILOR DE INSTRUIRE ADAPTIVE

Premiza de la care pornește propunerea noastră este:

Cunoaşterea pedagogică despre strategiile şi euristicile de conducere a procesului de instruire poate fi descrisă în *termeni generici, de task/metodă pedagogică* (TP/TM) și poate fi adaptată în funcție de contextul de aplicare pentru un anumit domeniu de instruire și pentru un anumit student. De aceea, o ontologie a strategiilor de instruire descrise prin paradigma task/metodă poate completa arhitectura unui sistem inteligent de instruire.

În acest caz, un sistem autor de dezvoltare se va baza pe:

1. ***Ontologia task-urilor pedagogice şi a metodelor pedagogice;***
2. **Ontologia conținutului instrucțional, care poate fi, la rândul ei, constituită din:**
    a. **Ontologia domeniului de instruire**
    b. **Ontologia resurselor de instruire**
3. **Ontologia studentului din punct de vedere al profilului de învățare.**

Modelul propus are ca punct de plecare conceptul de *task instrucțional* introdus de către VanMarcke [181]. Sistemele GTE ("Generic Tutoring Environment") și DCG ("Dynamic Course Generation"), propus de Vassileva [182], în urmă cu 2 decenii, sunt sisteme de referință, introducând conceptele de generare a cursului și a prezentării. GTE a fost primul sistem cu o reprezentare explicită a expertizei instrucționale, separată de cea a domeniului; sistemul nu folosește tehnici AI, implementează un algoritm propriu, de procesare a semnalelor.

Abordarea propusă se apropie de proiectarea pedagogică prin scenarizare, cu următoarele observații:

- Scenariul descrie desfășurarea unui task precis într-un context și un sistem dat.
- Task-ul descrie activităţi posibile din punct de vedere teoretic şi relaţiile dintre ele.
- Scenariile au o acoperire medie şi sunt mult mai precise.
- Scenariile pot ajuta la modelarea task-urilor.
- Modelele task-urilor pot ajuta la identificarea scenariilor interesante.

Definiția 4-5 Task-ul pedagogic (TP) este activitatea realizată de către profesorul-pedagog pe parcursul procesului instrucţional, la diferite niveluri de granularitate. (CE?)

Definiția 4-6 Metoda pedagogică (MP) exprimă modalitatea de realizare a task-ului pedagogic (CUM?)

Exemple de task-uri pedagogice: *Selectează-Noțiune, Propune-Exercițiu, Rezumă, Furnizează element ajutător, etc.* Ele oferă caracter de generalitate sistemului, fiind întâlnite în diferite contexte instrucţionale (diferite domenii şi diferiţi studenţi).

Metodele, care descriu modul de realizare a task-ului, au același caracter de generalitate.

De exemplu*, task-ul de clarificare a unui concept* poate fi realizat prin metodele: *clarificare_cu_exemplu*, *clarificare_cu_analogie*, *clarificare_cu_simulare*, *clarificare_cu_descriere*, *clarificare_cu_definiţie* (figura 4-6).

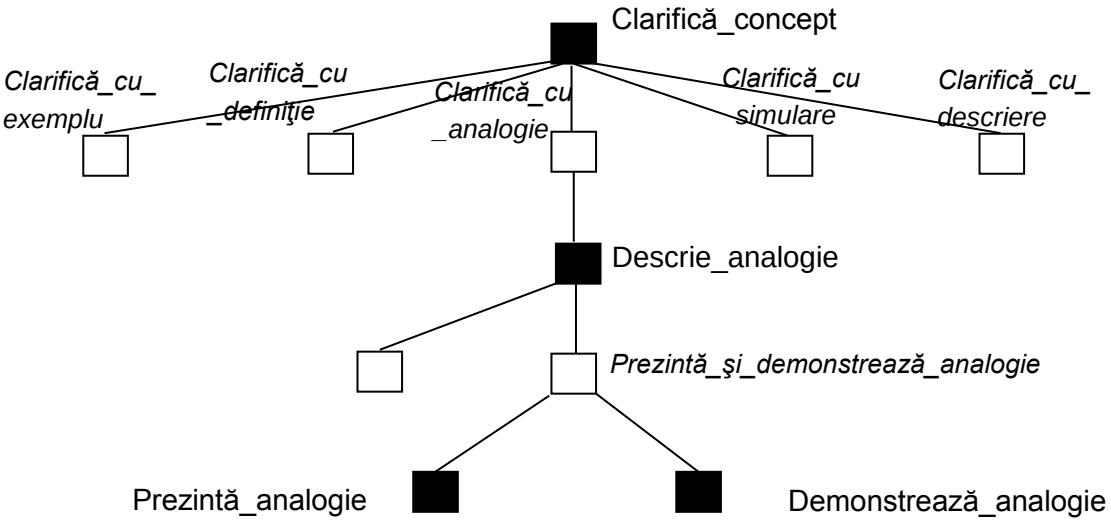


Figura 4-6 Exemplu de descompunere a unui task repetitiv

În orice moment al procesului de instruire, se alege acea metodă, considerată ca cea mai potrivită **contextului instrucţional**.

De exemplu: Metoda Clarifică_cu analogie poate fi aplicată dacă:
1. Există o resursă pedagogică disponibilă pentru analogie și
2. Studentul este familiar cu analogia

Pentru fiecare metodă se poate calcula o valoare de aplicabilitate, în funcţie de condiţii:
- absolute (disponibilitatea materialului, principii pedagogice, modelul elevului, starea procesului) și
- relative (de exemplu, pentru un elev de tip reflector, se ia în considerare o informație ajutătoare).

Strategia pedagogică poate fi modelată ca o problemă de planificare ierarhică. Drept urmare, ontologia strategiilor pedagogice este o ontologie a TP/MP (task-urilor/metodelor pedagogice) – conform abordării funcționale a sistemelor bazate pe cunoștințe - (prezentate în subcapitolul 1.4.4).

4.4 CONTRIBUȚII LA MODELAREA FORMALĂ A STRATEGIILOR DE INSTRUIRE ADAPTIVE

Cunoaștere a *declarativă* despre o strategie pedagogică este constituită din elementele TP/MP (task pedagogic/metodă pedagogică).

Pentru reprezentarea cunoașterii despre strategiile pedagogice se va folosi același formalism, al grafurilor conceptuale (avantajele au fost enumerate pe parcursul lucrării) și vom utiliza aplicația OntL_CG pentru validarea bazei de cunoștințe (testele interne), pentru raționamente deductive sau pentru exprimarea axiomelor domeniului.

Reprezentarea cunoașterii pedagogice cu ajutorul grafurilor conceptuale a fost prezentată și în alte lucrări ([183], [184], [185], [186]).

Baza de cunoștințe despre strategiile pedagogice descrise în termeni de task-uri și metode este structurată pe nivelurile ilustrate în figura 4-7.

Definiția 4-7 Ontologia task-urilor și metodelor este definită ca:
$O = (C, \mathcal{R}, \mathcal{A}_S, \mathcal{A}_D)$, în care:

- C - conceptele de înalt nivel *(task aplicabil, metodă aplicabilă)*
- $\mathcal{R}$ - relațiile dintre ele
- $\mathcal{A}s, \mathcal{A}_D$ – axiomele schemă și axiomele domeniului.

Ne vom referi la nivelul 4, la task-urile și metodele pedagogice:

Definiția 4-8 Baza de cunoștințe a task-urilor și metodelor pedagogice bazată pe ontologia O este $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{E}, C)$, unde $\mathcal{F}$ este mulțimea faptelor (grafuri conceptuale), $\mathcal{R}$ este mulțimea regulilor de inferență, $\mathcal{E}$ este mulțimea regulilor de evoluție, iar C este mulțimea constrângerilor (pozitive, C^+ și negative, C^-). Dacă C este vidă, atunci $\mathcal{R}$ și $\mathcal{E}$ sunt identice.

Elementele domeniului de rezolvare a problemei de planificare ierarhică (realizată prin descompunerea task-metodă) este o **reuniune** de elemente din domeniul de instruire, din domeniul studentului și din domeniul TPMP.

De exemplu, un task pedagogic de planificare a unei unități pedagogice (curs) va folosi în planificarea traseelor de instruire sau a modului de prezentare a noțiunilor aferente concepte și relații din domeniul de instruire ([187], [188], [189], [190], [191]).

Un task de prezentare a unei noțiuni, poate alege o demonstrație, o expunere, etc., în funcție de caracteristicile contextului ([192], [185], [184]).

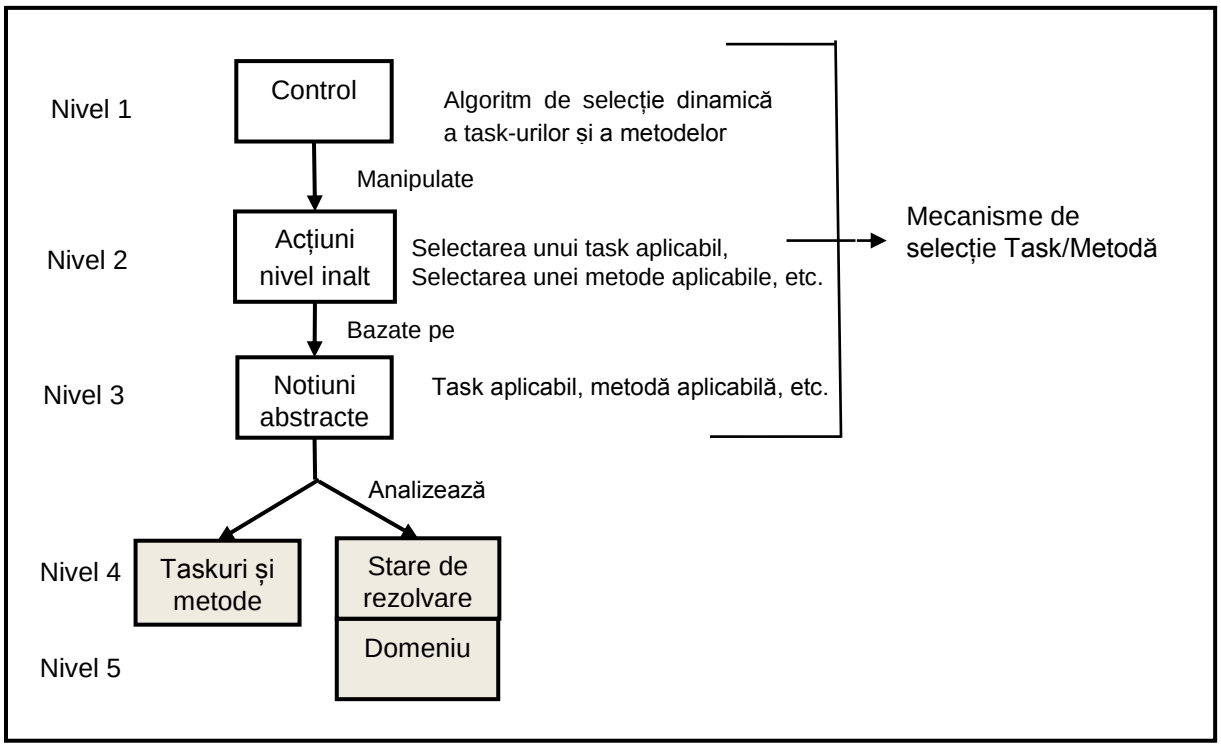


Figura 4-7 Arhitectuta multi-nivel a bazei de cunoștințe despre strategiile pedagogice

La nivelul 1 se realizează un control general asupra acțiunilor de la nivelul 2, cum ar fi selectarea unui *task aplicabil* sau selectarea unei *metode aplicabile*. Acțiunile de nivel-înalt (1 și 2) se bazează pe noțiunile abstracte de la nivelul 3 (*Task aplicabil*, *Metodă aplicabilă*) care exprimă stările posibile în care se pot găsi un task sau o metodă în cursul procesului de rezolvare a problemei. Noțiunile abstracte de la nivelul 3 sunt definite pe baza descrierii primitivelor de modelare task și metodă, descriere realizată la nivelul 4, prin *Obiectivele unui task* și *Contextul de selecție* al unei metode. Nivelul 4 cuprinde, deasemenea, task-urile și metodele efective, instanțe ale primitivelor de modelare și starea de rezolvare a problemei. La nivelul 5 sunt definite conceptele, relațiile, constrângerile și faptele domeniului de rezolvare a problemei.

Modul de rezolvare a problemei constă într-un proces de planificare ierarhică. Starea de rezolvare – datele inițiale ale problemei și evoluția rezolvării – este descrisă prin două grafuri conceptuale simple: graful OG (obiective găsite) și RP (rezultate/cunoștințe produse de către metode). Starea este actualizată de către sistem sau, eventual, în urma dialogului cu utilizatorul. Starea va fi actualizată într-un mod monoton, în sensul că: grafurile OG și CP calculate la pasul i sunt mai specifice decât grafurile obținute la pasul i-1 și mai generale decât cele produse la pasul i+1.

REPREZENTAREA PRIMITIVELOR DE MODELARE A TASK-URILOR ȘI A METODELOR

Primitivele de modelare sunt task-urile și metodele. Fiecare primitivă (task sau metodă) este reprezentată printr-o regulă activată în funcție de context. Parametrii formali ai regulii sunt elemente din baza de cunoștințe de conținut și din baza de cunoștințe despre student.

Mecanismul de selecție a task-urilor și a metodelor se realizează pe baza legăturilor dintre obiectele din ontologia task-urilor și metodelor și cele din ontologia domeniului de instruire și a studentului, prin punerea în corespondență a proprietăților primitivelor de modelare cu cele din starea de rezolvare.

Regulile de producţie pot fi utilizate în modelele bazate pe teoria grafurilor conceptuale, fiind considerate tot o extindere a modelului de bază. Regulile de inferenţă se bazează pe operaţiile asupra grafurilor.

Reprezentarea task-urilor

Proprietățile pentru primitiva de modelare task sunt:

- *Condiţia de activare* descrie contextul în care task-ul poate fi realizat şi este dată de antecedentul regulii. Parametri formali sunt obiectele cu care lucrează task-ul.
- *Obiectivele* descriu la nivel abstract scopul task-ului și apar ca postcondiţie (consecventul regulii).
- *Metodele asociate*: listă de metode prin care poate fi realizat task-ul.
- *Regulile de interpretare* sunt reguli care permit abstractizarea rezultatelor obţinute printr-o metodă la nivelul corespunzător obiectivului unui task (o listă de CGR).

Reprezentarea metodelor

Proprietățile pentru primitiva de modelare metodă sunt:

- *Contextul de selecție* - descrie precondițiile în care are metoda este selectată și este reprezentat prin antecedentul regulii care reprezintă metoda. Parametri formali indică resursele metodei.
- *Rezultatele produse* corespund consecventului regulii care reprezintă metoda.

Primitiva **metodă operaţională** are o *proprietate adiţională* numită Tratament, care specifică implementarea (o componentă soft-ware). Când *contextul de selecție* al metodei operaţionale este verificat, metoda este executată iar rezultatele acesteia sunt mai specifice (în sensul operației de proiecție) decât rezultatele/cunoștințele produse de către M.

Primitivele pentru **metodele de descompunere** au proprietățile adiționale:

- *Tip*: specifică controlul prin care metoda realizează descompunerea (ordonată, neordonată, iterativă)
- *Descompunerea*: lista de task-uri.

Metodele de descompunere permit descrierea unui graf aciclic al task-urilor și al metodelor (figura 4-8), cu un o singură rădăcină (un arbore), reprezentând obiectivele dorite, frunzele sunt metode operaționale, iar nodurile intermediare sunt task-uri și metode de descompunere a task-urilor.

Definiția 4-10 Baza de cunoștințe despre task-uri și metode este **consistentă** dacă şi numai dacă toate *task-urile ei sunt bine definite*.

Definiția 4-11 Un task T este **bine-definit** dacă toate *metodele asociate acestuia sunt relevante.*

Definiția 4-12 *O metodă M este* ***relevantă*** *pentru task-ul T dacă şi numai dacă:*

a) Există o proiecţie a *contextului de activare a lui T* în *contextul de selecţie a metodei M*.
b) Există o proiecţie a *obiectivelor lui T* în graful obţinut prin (prin raționament cu înlănțuire înainte) aplicarea regulilor de interpretare ale lui T în *rezultatele produse* ale metodei M.

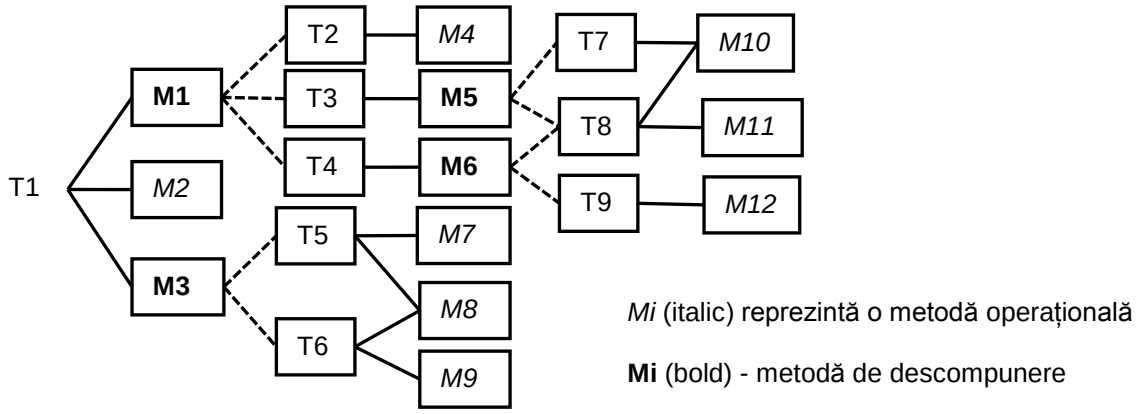


Figura 4-8 Graful task-urilor și metodelor

Primitivele de modelare a noțiunilor abstracte de la nivelul 4 pot fi: task realizat, task productiv, task aplicabil, metodă aplicabilă și constituie criterii de selecție.

Definiția 4-13 Un task T este **realizat** dacă și numai dacă există o proiecţie de la obiectivele lui T în OG (partea din starea de rezolvare care reprezintă obiectivele găsite).

Definiția 4-14 Un task T este **productiv** dacă și numai dacă nu există nici o proiecţie de la obiectivele lui T în RP (cunoştinţe produse pe durata întregului proces de rezolvare).

Definiția 4-15 Un task T este **aplicabil** dacă şi numai dacă există o proiecţie a contextului de activare în OG.

Definiția 4-16 O metodă M este **aplicabilă** dacă și numai dacă există o proiecţie a contextului de activare a sa în RP.

Proiecţia contextului de selecţie al metodei M în RP este calculat prin cuple definite prin:

- Imaginea contextului de activare a task-ului corespunzător T în contextul de selecție al metodei M.
- Imaginea în RP a imaginii contextului de activare a lui T în OG (datorită faptului că, la fiecare pas, o proiecţie de la OG în RP este menţinută şi există o proiecţie de la contextul de activare a lui T în OG).

În cazul existenței mai multor proiecții a contextului de activare a task-ului în OG, pentru eliminarea nedeterminismului, soluțiile propuse sunt:

- În urma unui dialog sistem-pedagog, pedagogul decide ce task va realiza;
- Putem restrânge calculul proiecției la o parte a grafului OG (putem lua în considerare contextul de activare al super-task-ului lui T);
- Prin constrângeri euristice asupra grafului.

SOLUȚII DE OPERAȚIONALIZARE

Motorul de inferență pentru baza de cunoștințe asupra strategiei pedagogice realizează o selecție dinamică a task-urilor și metodelor intr-un context dat.

În cazul modelului propus, modelul de rezolvare a problemei este inițializat prin două SCG vide. Pe parcursul rezolulției, cele două grafuri sunt actualizate (obiectivele task-ului realizat sunt adăugate la OG, iar rezultatele metodelor operaționale sunt adăugate la RP) și extinse prin inferențele regulilor de interpretare (figura 4-9).

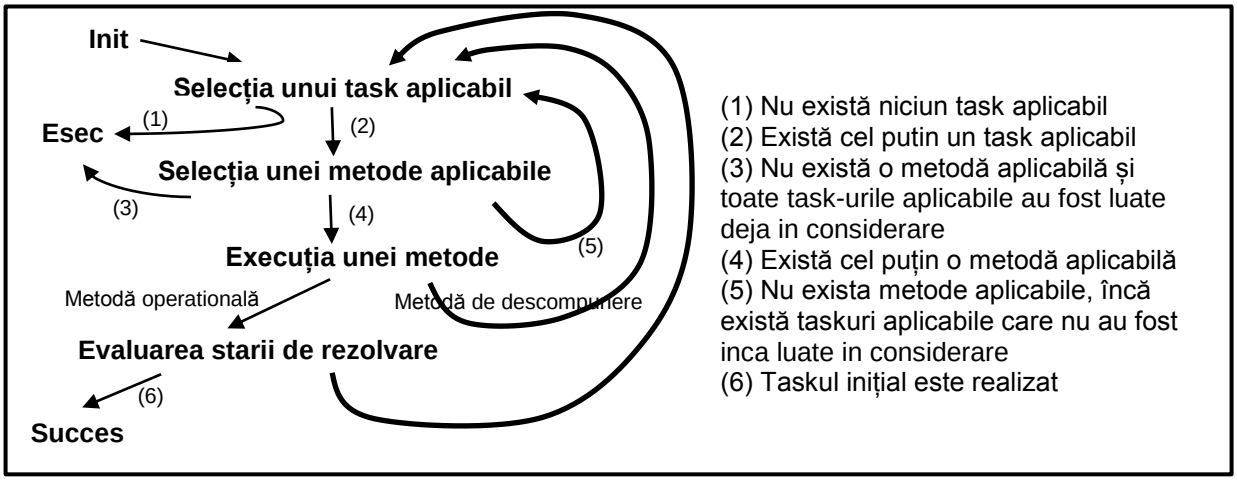


Figura 4-9 Raționamentul realizat de către motorul de inferență

4.5 CONCLUZII ȘI CONTRIBUȚII

Au fost prezentate argumentele care susţin faptul că sistemele de instruire inteligente sunt sisteme bazate pe cunoştinţe, în care cunoaşterea despre strategiile şi tacticile pedagogice de instruire trebuie reprezentată separat de cunoaşterea domeniului în care se realizează instruirea. Evident, cunoştinţele domeniului şi cunoştinţele tutoriale nu sunt independente, dar această separare (între “formă” şi “conţinut”), însoţită de o reprezentare explicită a cunoştinţelor tutoriale, poate remedia într-o măsură apreciabilă ceea ce li se “reproşează” tutorialelor inteligente, şi anume, “pedagogia implicită”.

În acest capitol a fost propus un cadru structurat de modelare a cunoașterii pedagogice, pe două niveluri: *cunoașterea pedagogică generală (la nivel macro)* și *cunoașterea pedagogică de conținut (la nivel micro)*. La micro-nivel, cunoașterea pedagogică este contextuală. Am definit cunoștințele pedagogice de conținut și strategiile pedagogice de conținut.

Am exemplificat modalitatea de aplicare a metodei MIIO (propusă în capitolul 2.3) unei ontologii a strategiilor pedagogice și am evidențiat avantajele utilizăriiaplicației OntL_CG (propuse în capitolul 3.6).

Am propus modelarea unei strategii pedagogice în termeni de task-uri și metode pedagogice, ca pe o problemă de planificare ierarhică. Cunoaşterea pedagogică despre strategiile şi euristicile de conducere a procesului de instruire poate fi descrisă în *termeni generici, de task/metodă pedagogică* (TP/TM) și poate fi adaptată în funcție de contextul de aplicare pentru un anumit domeniu de instruire și pentru un anumit student. O ontologie a strategiilor de instruire descrise prin paradigma task/metodă poate completa arhitectura unui sistem inteligent de instruire. Am definit termenii la nivel conceptual și formal și am propus o metodă de operaționalizare.

" This is not the end.
It is not even the beginning of the end.
But it is, perhaps, the end of the beginning. "

"Acesta nu este sfârşitul.
Nu este nici începutul sfârşitului.
Este, poate, sfârşitul începutului."

(Winston Churchill , 1874 – 1965)

5. CONCLUZII, CONTRIBUȚII ȘI DIRECȚII VIITOARE DE CERCETARE

In ultimele decenii s-a deschis o poartă largă sistemelor software educaționale, ca rezultat al necesității de modernizare și eficientizare a activităților pedagogice și favorizată de dezvoltarea explozivă a tehnologiilor informatice. Cu o astfel de motivație, lumea științifică a căutat să răspundă cât mai bine cerințelor, concentrându-se pe diverse aspecte ale pedagogiei, cu diferite obiective, pentru diferite categorii de public. A rezultat astfel o mare diversitate de propuneri, soluții, implementări.

În cea mai mare masură, sistemele software de instruire asistată de calculator urmează scheme uzuale de prezentare a tematicii, simulând obișnuitele lecții predate de profesor în fața elevilor. Dar marele câștig pe care îl pot aduce astfel de sisteme informatice ar rezulta tocmai din flexibilizarea procesului de predare, adaptarea firului prezentării cunoștințelor pentru fiecare student în parte, generarea dinamică a traseului de activități, etc.

Obiectivul principal al lucrării de faţă a fost acela de a aduce contribuţii la realizarea unui sistem de instruire asistată de calculator, în sensul de a-l pregăti pentru aplicarea contextuală a strategiilor pedagogice în procesul de instruire.

În acest sens, am găsit că:

- Ontologiile interne sau externe unui sistem bazat pe cunoștințe pot servi la o reprezentare a componentelor bazei de cunoștințe a sistemului (atât a cunoașterii declarative, cât și a celei de raționament: modelul domeniului – conținut, modelul studentului, modelul strategiei). În plus, ontologiile permit acumularea şi interoperabilitatea cunoştinţelor (pe web); astfel, utilizatorii (profesori, pedagogi, proiectanţi ai sistemelor educaţionale, elevi) pot accesa materialele educaţionale relevante aflate în Internet, profesorii pot folosi strategii pedagogice care și-au demonstrat deja eficacitatea, etc. În cadrul acestei lucrări ontologiile au fost utilizate pentru rezolvarea unor probleme de *achiziţie, reprezentare şi modelare a cunoaşterii (pedagogice)*.

- Ingineria ontologică oferă cadrul adecvat pentru cercetările orientate spre conținut: metodologii și principii pentru ontologii, instrumente de lucru cu ontologiile. Interesul nostru pentru ingineria ontologică s-a concentrat pe *aspectele de concepție, formalizare șI operaționalizare* a unei ontologii, dar și pe căutarea *unor modalități de a reprezenta cunoștințele ontologice* cât mai puțin dependent de aplicația care va utiliza ontologia.
- Formalismul grafurilor conceptuale oferă un model matematic bazat atât pe teoria grafurilor, cât şi a logicii (raţionamentul bazându-se pe algoritmi asupra grafurilor (proiecție) sau pe logica predicatelor de ordinul I); în același timp, el prezintă prezintă avantajul unei reprezentări grafice în procesul de modelare, foarte accesibilă unui nespecialist în reprezentarea cunoștințelor. În lucrare, formalismul a fost folosit nu numai pentru reprezentarea cunoștințelor declarative, ci și pentru cele de raționament.
- O strategie pedagogică poate fi modelată ca un proces de planificare ierarhică; pe baza acesteia va rezulta activitatea propusă la fiecare pas al instruirii. În alegerea unei strategii pedagogice (la nivelul unei lecții, a unui concept, etc.) trebuie să se țină seama de o serie de factori ca: obiectivul pedagogic vizat, stilul cognitiv al studentului și o serie de alte elemente. Un sistem de instruire ar trebui să aplice nu doar o singură strategie, ci să poată alege din mai multe posibile, în funcție de situație.
- Proiectarea sistemelor inteligente de instruire este o problemă extrem de complexă (un astfel de sistem este, practic, un triplu sistem-expert), necesită expertiza unor specialiști din mai multe domenii și o reprezentare explicită a cunoaşterii (referitoare la domeniul în care se realizează instruirea, persoana instruită și procesul de instruire). Ca urmare, proiectarea trebuie susținută de sistemele de dezvoltare autor, care sunt, la rândul lor, sisteme bazate pe cunoștințe și pot utiliza cunoașterea ontologică.

În contextul de mai sus, contribuțiile metodologice, teoretice și aplicative aduse prin această lucrare sunt:

- Sistematizarea într-o *manieră personală a elementelor din domeniul de cercetare* (instruirea asistată, ingineria ontologică, ontologii); realizarea unor *studii comparative* și formularea unor concluzii (unele dintre acestea determinând stabilirea unor obiective ale lucrării).
- Propunerea unei *metode integrate de inginerie ontologică (MIIO)*, bazate pe principiile și metodele ingineriei ontologice, *pe metodele ingineriei ontologice* (intergrează elemente precum *considerearea ciclului de viață al ontologiil*or din On-To-Knowledge și METHONTOLOGY, *strategiile de identificare a conceptelor* propuse de Uschold&King, *întrebările de competență* din Grüninger&Fox și *criteriile de evaluare* din *METHONTOLOGY şi Goméz-Pérez)* și *pe standardele din ingineria software* (capitolul 2). Menționăm că metoda a fost validată parțial (motive și concluzii în subcapitolul 2.4).
- *Propunerea şi validarea teoretică şi experimentală a unei o metode şi a unui instrument de creare şi operaţionalizare a unei ontologii puternic formalizate, care să poată fi utilizată în sistemele bazate pe cunoștințe într-un mod cât mai puțin dependent de scopul operațional al acestor sisteme.*
- *Punerea în acord a două obiective aparent contradictorii* (ontologiile trebuie să integreze cunoștințele terminologice ale domeniului și să exprime semantica acestuia, păstrând independența față de utilizarea lor operațională în cadrul unui sistem bazat pe cunoștințe). Pentru aceasta, s-au realizat:
    - Un *limbaj de reprezentare a unei ontologii, OntLCG, caracterizat prin elementele:*

        - Este bazat pe paradigma Entitate-Relație (spre deosebire de majoritatea celorlalte, bazate pe cadre).
        - Permite construirea unei *ontologii puternic formalizate* (nu doar a unei taxonomii a termenilor domeniului).
        - Permite reprezentarea cunoștințelor și a proprietăților acestora.
        - Limbaj *grafic, bazat pe modelul grafurilor conceptuale* (SCG extins cu reguli și constrângeri [175]). Operația fundamentală a modelului CG, proiecția, poate fi ușor înțeleasă de către un expert în domeniu, nespecialist în informatică și/sau în limbajul de reprezentare și îi permite acestuia să urmărească, pas cu pas, raționamentele realizate, cu scopul de a valida reprezentarea.
        - Sprijină *raționamentul, indiferent de forma acestuia* (interogarea unei baze de cunoștințe, inferență asupra unor fapte sau testarea coerenței bazei de cunoștințe) și de axiomele care exprimă semantica domeniului.
        - Definirea elementelor (nivelul axiomatic, interpretare în teoria mulțimilor)
    - Propunerea unei metode care să permită utilizatorului un mod de specificare intuitiv și simplu a scopurilor și modului în care cunoștințele din ontologie vor fi folosite într-o aplicație: nivelul terminologic este același, doar reprezentările operaționale ale axiomelor sunt determinate de scopul aplicației.
        - Definirea noțiunii de scenariu de utilizare.
        - Definirea noțiunii de context de utilizare.
        - Identificarea contextelor de utilizare (inferențial/validare, explicit/implicit).
    - Limbajul de operaționalizare al metodei propuse poate fi *oricare* (toate axiomele sunt bazate pe λ-abstracțiuni)ty8, dar s-au ales grafurile conceptuale.
    - Utilizatorul poate specifica în mod grafic orice axiomă a ontologiei domeniului
    - Utilizatorul poate exprima (cu ajutotul axiomelor schemă) și operaționaliza proprietățile relațiilor și a conceptelor (reflexivitatea, simetria, tranzitivitatea, cardinalitatea, etc.) unei relații oarecare, incompatibilitatea a două relații, genericitatea unui concept – în sensul din CG, concept fără instanțe, etc.).
    - Aplicația OntL_CG (ca și limbajul) - aplicație client, implementată în Java, care comunică cu serverul CoGITaNT prin fișiere .xml - compusă dintr-un editor grafic, mecanism de operaționalizare care folosește modelul SCG(cu reguli și constrângeri), motor de inferență care folosește biblioteca CoGITaNT).
    - Permite validarea ontologiei și permite raționament deductiv
    - Ilustrarea prin studiul de caz al modelării și operaționalizării unei strategii pedagogice într-o bază de cunoștințe cu ajutorul aplica'iei OntL_CG.
- Modelarea într-un formalism unic, cel al grafurilor conceptuale, atât a cunoștințelor declarative, cât și a celor de raționament (capitolele 3 și 4).
- Contribuții la modelarea strategiilor pedagogice adaptive
    - Definirea termenilor de descriere a universului de discurs;
    - Modelarea conceptuală bazată pe paradigma task pedagogic/metodă pedagogică (TP/MP);
    - Reprezentarea explicită a cunoașterii declarative și procedurale despre strategiile pedagogice într-un sistem autor, bazat pe cunoștințe;
    - Definirea formală a strategiei pedagogice;
    - Identificarea surselor de adaptivitate ale unei strategii pedagogice;
    - OntL_CG – componentă de asistență în crearea OTPMP;

Propunerile din teză reușesc să modeleze doar anumite aspecte ale activității de proiectare pedagogică.

Direcțiile de cercetare sunt numeroase, fiecare având un grad de complexitate ridicat, așa că vom enumera numai câteva dintre cele care se bazează *în mod direct pe rezultate prezentei lucrări*:

- Implementarea modelului formal propus în capitolul 4 într-un sistem autor de proiectare pedagogică.
- Utilizarea nu numai a grafurilor conceptuale simple cu reguli și constrângeri, ci și a altor extinderi ale modelului: grafurile imbricate – pentru gestiunea contextelor.
- Modelarea și implementarea unor module de asistență a pedagogului în conceperea unei strategii pedagogice (reactiv la deciziile incorecte de proiectare, explicativ).
- Utilizarea aplicației OntL_CG pentru alinierea/fuzionarea termenilor folosiți în proiectarea pedagogică.
- O abordare multi-agent a sistemului autor.

6. LISTA LUCRĂRILOR PUBLICATE ȘI PREZENTATE

Articole publicate în reviste cotate ISI

1. Pecheanu, E., Segal, C., **Ştefănescu, D.**, *"Content Modeling in Intelligent Instructional Environments"*, In Proceedings of the 7th Int. Conf. "Knowledge-Based Intelligent Information and Engineering Systems, Oxford, UK, Sept 2003, Proceedings, Part II, pp 1229-1235, Ed. Springer-Verlag, Berlin, ISBN 3-540-40804-5, ISSN 03020-9743, Springer Verlag Berlin, In Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence, LNCS-LNAI Vol 2774/2003, ISBN 3-540-40804-5 Springer Verlag Berlin, pp. 1229-1235, www.springerlink.com/content/qpkfqn5emhpb0kge (**Journal ISI quoted),** http://apps.webofknowledge.com/summary.do?SID=S1%40DM8dOlB28cCBcOL8&product=WOS&qid=33&search_mode=GeneralSearch, 2003.

2. Pecheanu, E., Dumitriu, L., **Ştefănescu, D.**, Segal, C., *"A Framework for Conceptually Modelling the Domain Knowledge of an Instructional System"*, International Conference on Computational Science (2), *Book Chapter*, Lecture Notes in Computer Science, In Lecture Notes in Computer Science-LNCS Vol 3992/2006, ISBN 978-3-540-34381-3 Publisher Springer Berlin / Heidelberg, Volume 3992/2006, ISSN 0302-9743, ISBN 3-540-34381-4, pp. 199-206, www.springerlink.com/content/f5417380320um935, *(Journal ISI quoted)* http://apps.webofknowledge.com/summary.do?SID=S1%40DM8dOlB28cCBcOL8&product=WOS&qid=33&search_mode=GeneralSearch, 2006.

Articole publicate în ISI Proceedings

1. Pecheanu, E., Dumitriu, L., **Ștefănescu, D.**, Stîngă O., - "*Developing Environments for Distance Engineering Education"* – Proceedings of the 7th World Conference on Continuing Engineering Education, WCCEE'98, Torino, Italia, pp. 528-534, 1998.
   ***ISI Proceedings*:**
   http://apps.webofknowledge.com/summary.do?SID=S1%40DM8dOlB28cCBcOL8&product=WOS&qid=35&search_mode=GeneralSearch

2. Dumitriu, L., **Ştefănescu, D.**, Pecheanu, E., *"Finding and using global implications for the association rule problem"*, Vol "Knowledge-based intelligent information engineering systems and allied technologies", KES 2002, Part 1, Edited by E Damiani, RJ Howlett, LC Jain, N Ichalkarange, ISBN I 58603 280 I (IOS Press), ISBN 4 274 90535 7 C3055 (Ohmsha), ISSN 0922-6389, pg 537-541, LNCS-LNAI ISBN I 58603 280, 2002.

ISI Proceedings:
IOS Press Nederland, ISBN: 978-1-58603-280-7 - http://www.iospress.nl/, http://www.iospress.nl/book/knowledge-based-intelligent-information-engineering-systems-and-allied-technologies/

3. **Ştefănescu, D.**, Pecheanu, E., *"Experiences in Courseware Development Using the ARIADNE Approach"*, Proceedings of the 2nd balkan region conference on engineering education, Bridges for co-operation in engineering education, Conference proceedings, 16-19 sept, 2003, Sibiu, Editors: Constantin Oprean, Claudiu Vasile Kifor, Nicolaie Georgescu, pg 160-163, ISBN 973-651-673-3, 2003.
ISI Proceedings:
http://apps.webofknowledge.com/summary.do?SID=S1%40DM8dOlB28cCBcOL8&product=WOS&qid=35&search_mode=GeneralSearch

4. **Ştefănescu, D.**, Pecheanu, E., Dumitriu, L., *"Knowledge representation models based on conceptual graphs in intelligent educational systems"*, Proceedings of The 3nd balkan region conference on engineering education, Advancing Engineering Education, Conference proceedings, 12-15 sept, 2005, Sibiu, Editors: Constantin Oprean, Claudiu Vasile Kifor, Nicolaie Georgescu, pg 67-70, ISBN 973-739-147-0, 2005.
ISI Proceedings:
http://apps.webofknowledge.com/summary.do?SID=S1%40DM8dOlB28cCBcOL8&product=WOS&qid=35&search_mode=GeneralSearch

5. Pecheanu, E., **Ştefănescu, D.,** Istrate, A, Jâşcanu, V., *"On Modeling Adaptive Web-based Instructional Systems"*, Proc. of the 8th RoEduNet International Conference on Networking in Education and Research, Galati, Romania, ISBN 978-606-8085-15-9, ***Journal ISI quoted (Web of Science), EI Compendex. Engineering Index, ISTP index, ISI index,*** pp. 84-90, 2009.

6. **Ştefănescu, D.**, Tudorie, C., Pecheanu, E., *"Diversity in Software Support for Computer-Assisted Education"*, Proc. of the 8th RoEduNet International Conference on Networking in Education & Research, Galati, Romania, ISBN 978-606-8085-15-9, ***Journal ISI quoted (Web of Science), EI Compendex. Engineering Index, ISTP index, ISI index,*** pp 29-33, 2009.

Articole publicate în reviste indexate în baze de date internaționale

1. Neagu, D., Bumbaru, S., **Ştefănescu, D.**, "*Structural and Procedural Knowledge Modelling using 3D Neural Networks"*, The Annals of "Dunarea de Jos" of Galati, Fascicle III, ISSN 1221-454X 454X (**Revista tip B, cod CNCSIS 482**), pg. 94-98, www.ann.ugal.ro/eeai/index.html BDI : www.doaj.org/doaj?func=subject&cpid=104, 1996.

2. **Ştefănescu, D.**, Pecheanu, E., Dumitriu, L., Neagu, C.D., *"Student Modelling in Educational Environments",* The Annals of "Dunarea de Jos" University of Galati, Fscicle III, ISSN 1221-454X 454X (**Revista tip B, cod CNCSIS 482**), www.ann.ugal.ro/eeai/index.html BDI : www.doaj.org/doaj?func=subject&cpid=104, pp. 74-79, 1999.

3. **Ştefănescu, D.**, Pecheanu, E., *"A Hybrid Approach to Dynamic Course Generation",* The Annals of "Dunarea de Jos" University of Galati, Fascicle III, ISSN 1221-454X 454X, (**Revista tip B, cod CNCSIS 482**), pp. 71-74, www.ann.ugal.ro/eeai/index.html BDI : www.doaj.org/doaj?func=subject&cpid=104, 2000.

4. Pecheanu, E., **Ştefănescu, D.**, Dumitriu, L., Istrate, A., *"Achitecture of a Computer Based Instructional System",* The Annals of "Dunarea de Jos" University of Galati, Fascicle III , ISSN 1221-454X 454X (**Revista tip B, cod CNCSIS 482**), pp. 60-64, www.ann.ugal.ro/eeai/index.html BDI : www.doaj.org/doaj?func=subject&cpid=104, 2000.

5. **Ştefănescu, D.**, Pecheanu, E., Buraga, S.C., *"Pedagogical agents in intelligent tutoring systems"*, Proceedings of the 11-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS 2001, Galati, octombrie 2001, pp. 178-183, ISBN 973-8139-98-8, 2001.

6. Pecheanu, E., **Ştefănescu, D.**, Buraga, S.C., Istrate, A., *"Integrating hypermedia objects in an intelligent tutoring system",* The Annals of "Dunarea de Jos" University of Galati, Fascicle III , ISSN 1221-454X 454X (**Revista tip B, cod CNCSIS 482**), pp. 60-64, 2000, www.ann.ugal.ro/eeai/index.html BDI : www.doaj.org/doaj?func=subject&cpid=104Proceedings of the 11-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS 2001, Galati, oct. 2001, pp.170-178, ISBN 973-8139-98-8, 2001.

7. Buraga, S.C., **Ştefănescu, D.**, Pecheanu, E., *"Modeling relations between resources of an internet teleconferencing system"*, Proceedings of the 11-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS 2001, Galati, octombrie, 2001, pp. 68-74, ISBN 973-8139-98-8, 2001.

8. Pecheanu, E., **Ştefănescu D.**, Istrate, A., "*An Extended Structural Model for Intelligent Instructional Systems",* The Scientific Annals of the A.I. Cuza University of Iasi, Computer Science Section Tome XV , ISSN 1224-2268, (**Revista Tip B, cod CNCSIS 115**), pp. 167-177, www.infoiasi.ro/Annals/content.jsp?c=15 BDI : www.informatik.uni-trier.de/~ley/db/journals/cuza/cuza15.html#PecheanuSI04a, 2004.

9. Pecheanu, E., **Ştefănescu D.**, Istrate, A., *"Building and Using Ontologies in Intelligent Instructional Systems",* The Scientific Annals of the A.I. Cuza University of Iasi, Computer Science Section Tome XV , ISSN 1224-2268, (**Revista Tip B, cod CNCSIS 115**), pp. 178-190, www.infoiasi.ro/Annals/content.jsp?c=15 BDI : www.informatik.uni-trier.de/~ley/db/journals/cuza/cuza15.html#PecheanuSI04a, 2004.

10. **Ştefănescu, D.**, Pecheanu, E., Cocu, A., *"Pedagogical Knowledge Model Based on Conceptual Graphs and Ontology",* The Annals of "Dunarea de Jos" University of Galati, Fascicle III , ISSN 1221-454X (**Revista tip B, cod CNCSIS 482**), pp 12-17, www.ann.ugal.ro/eeai/index.html BDI: www.doaj.org/doaj?func=subject&cpid=104, 2005.

11. Cocu, A., **Ştefănescu, D.**, *"Uncertainty management using bayesian networks in student knowledge diagnosis",* The Annals of "Dunarea de Jos" University of Galati, Fascicle III, pp. 29-32, (**Revista tip B, cod CNCSIS 482**), ISSN 1221-454X, www.ann.ugal.ro/eeai/index.html BDI : www.doaj.org/doaj?func=subject&cpid=104, 2005.

12. Tudorie, C., Frangu, L., **Ştefănescu, D.**, *"Automatic modeling of the linguistic values for database fuzzy querying",* The Annals of "Dunarea de Jos" University of Galati, Fascicle III, (**Revista tip B, cod CNCSIS 482**), ISSN 1221-454X, www.ann.ugal.ro/eeai/index.html BDI : www.doaj.org/doaj?func=subject&cpid=104, 2007.

13. Buraga, S.C., **Ştefănescu, D.**, *"Ontology Editing Support – A Study of Existing Tools"*, Proceedings of the 13-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS Galati, Editors Luminiţa DUMITRIU, Mihai VLASE, Galati University Press, ISSN 1843-5130, 2007.

14. Tudorie, C., **Ştefănescu, D.**, Bumbaru, S., *"Discovery linguistic values definitions from the database content in a fuzzy querying context"*, Proceedings of the 13-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS Galati, Editors Luminiţa DUMITRIU, Mihai VLASE, Galati University Press, ISSN 1843-5130, 2007.

15. Pecheanu, E., **Ştefănescu, D.**, Istrate, A, *"On Modeling Instructional Content in Computer Assisted Learning"*, The Annals of "Dunărea de Jos" University of Galaţi, Fascicle III, Electrotechnics, Electronics, Automatic Control, Informatics, Vol. 32, Nr. 2, ISSN 1221-454X, Paginile: 21-24, http://www.ann.ugal.ro/eeai/archives/2009/Lucrare-4-EPecheanu-p21-24.pdf, (**Indexare BDI: DOAJ - Directory of Open Access Journals, 2010, Lund University Libraries**), http://www.doaj.org/doaj?func=openurl&issn=1221454X&genre=journal, 2009.

16. Pecheanu, E., Dumitriu, L., Segal, C., **Ştefănescu, D.,** *"Methods to Evaluate Open Source Learning Platforms"*, in Global Engineering Education Conference, EDUCON 2011: New Pedagogic Challenges in Engineering Education, Amman, Iordania, **IEEE**, ISBN: 978-1-61284-642-2, **INSPEC Accession Number:** 12029769, Digital Object Identifier: 10.1109/EDUCON.2011.5773292, pp. 1152 – 1161, 2011, http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5773291, 2011.

Articole publicate în buletine ale simpozioanelor internaţionale

1. Răzmeriţă, L., **Ştefănescu, D.**, Bumbaru, S., Istrate, A., *"Student testing and assessment in Computer Based Training Systems"*, Proceedings of 4-th International Conference Computer Aided Engineering Education, Krakov, Poland, pg. 223-230, ISBN 83-7108-029-8, 1997.

2. Pecheanu**,** E.**,** Dumitriu, L., **Ştefănescu, D.,** Bumbaru, S., "*Mentor – a Computer-Based Instructional and Assessing System"*, Proceedings of the 5th International Conference on Computer Aided Engineering Education, Sofia, Bulgaria, pp. 274- 279, 1999.

3. Pecheanu, E., **Ştefănescu, D.**, Segal, C., Popescu, F., *"Assisted-Learning and Individual Cognitive Style"***,** Proceedings of the Third International Conference "New Horizons in Industry and Education" – NHIE''03, august 2003, Santorini, Grecia, pp 258-264, ISBN 9609-85316-7-5, 2003.

4. Popescu, F., Pecheanu, E., **Ştefănescu, D.**, Istrate, A., *"New Solutions in Building Integrated Educational Frameworks"***,** Proceedings of the Third International Conference "New Horizons in Industry and Education" – NHIE''03, august 2003, Santorini, Grecia, pp 270-276, ISBN 9609-85316-7-5, 2003.

5. Pecheanu, E., **Ştefănescu, D.**, Istrate, A., Dascalescu, D., *"Modeling the Domain Knowledge of an Instructional Environment",* publicatie on-line, The 3rd Annual Ariadne Conference,Leuven, Belgia, http://rubens.cs.kleuven.ac.be/ariadne/CONF2003/intro.html, 2003.

6. Pecheanu, E., ***Ştefănescu***, D., Istrate, A., Dascalescu, D., *"Conceptually Modelling the Domain Knowledge of an Instructional Environment"***,** Proceedings of the International Conference on Computer aided learning in Engineering Education – CALIE''04, febr. 2004, Grenoble, Franta, pp. 215 –221, 2004.

Articole publicate în buletine ale simpozioanelor cu participare internaţională

1. **Ştefănescu, D.**, Bumbaru, S., *"The Instructional Theory"*, The 9-th International Symposium on Modeling, Simulation and Systems Identification, SIMSIS '96, Galati, oct. 1996, pg. 382-387, 1996.

2. **Ştefănescu, D.**, Bumbaru, S., Ariton, V., *"Hybrid Systems for Assisted Learning"*, The 9-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS '96, Galati, oct. 1996, pg. 372-381, 1996.

3. Răzmeriţă, L., **Ştefănescu, D.**, Bumbaru, S., Istrate, A., *"Computer testing and assesing package in Computer Based Training Systems"*, Proceedings of 3-rd International Workshop Session: Internet as a Vehicle for Teaching, Kida-Ilieni, Romania, RILW 1998, ISBN 973-97403-40, POLYGON si COMPREX, Cluj, 1998.

4. Bumbaru, S., Pecheanu, E., **Ştefănescu, D.**, *"MENTOR – A Project for Computer-Aided Educational Systems",* Proceedings of the International Symposium "The Role of Academic Education and Research in the Development of Information Society", Bucuresti, apr. 1999, pp. 89-94, 1999.

5. Segal, C., Dumitriu, L, **Ştefănescu, D.**, *"Test properties in a possibilistic relational diagnosis model"*, The 11-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS 2001, pp. 184-189, ISBN 973-8139-98-8, 2001.

6. Pecheanu, E., **Ştefănescu, D.**, Dumitriu, L., Istrate, A., *"Architecture of an intelligent tutoring system",* The 7-th International Symposium on Automatic Control and Computer Science and Parallel Workshop on Control Theory, Modeling, Simulation and Systems' Identification - Programme and abstracts, SACCS2001, Iasi, octombrie 2001, pp. 100, CD+ISBN 973-8292-10-7, 2001.

7. **Ştefănescu, D.**, Pecheanu, E., Bumbaru, S., *"Using pedagogical agents in intelligent tutoring systems"*, The 7-th International Symposium on Automatic Control and Computer Science and Parallel Workshop on Control Theory, Modeling, Simulation and Systems' Identification - Programme and abstracts, SACCS2001, Iasi, octombrie 2001, pp.100, CD+ISBN 973-8292-10-7, 2001.

8. **Ştefănescu, D.**, Pecheanu, E., Istrate, A., Segal, C., *"An object oriented approach to produce distributed educational hypermedia software"*, Proceedings of the International Conference, RILW 2001, august 2001, Miercurea-Ciuc, pp 182-187, 2001.

9. **Ştefănescu, D.**, Pecheanu, E., Dumitriu, L., Popescu, F., *"Student Modeling in WWW-based tutoring Systems - An overview"*, lucrare prezentata ca tutorial la International Conference RILW2001, Miercurea-Ciuc, http://rilw.emp.paed.uni-muenchen.de/2001/info.html-, 2001.

10. Pecheanu, E., **Ştefănescu, D.**, Istrate, A., Popescu, F., *"Student modeling methods in WWW-based tutoring systems"*, 10 pag., International Conference on Qualification and Training in Information and Communication Technologies, October, 2002, Galaţi, CD+ISBN 973-8409-48-9, 2002.

11. **Ştefănescu, D.**, Pecheanu, E., Buraga, S.C., *"Integrating a Virtual Library in an Educational System"*, International Conference on Qualification and Training in Information and Communication Technologies, October, 2002, Galaţi, CD+ISBN 973-8409-48-9, 2002.

12. Buraga, S.C., **Ştefănescu, D.**, Pecheanu, E., *"VMRL Representation of Lindenmayer Systems"*, The 6-th International Conference on Development and Aplication Systems – DAS 2002, May 2002, Suceava, ISBN 973-88670-9, 2002.

13. Pecheanu, E., **Ştefănescu, D.**, Dumitriu, L., Segal, C., *"Knowledge Modelling in Computer-Assisted Instruction"*, OL-KWM - International Symposium on Organizational Learning and Knowledge Work Management, Bucuresti, 2005.

14. Novac-Ududec, C., **Ştefănescu, D.**, *"Patterns Instruction in Software Engineering"*, Proceedings of the International Conference of "Interactive computer aided learning" ICL2007: Eportofolio and Quality in e-learning, 8 pages, http://telearn.noe-kaleidoscope.org/open-archive/search?resource=1658_v1&back=%2Fopen-archive%2Fsearch%3Fsearch%3Dnovac%26type%255B%255D%3Dpublication%26type%255B%255D%3Dvideo%26type%255B%255D%3Dtool%26orderBy%3Dtitle%26search_button%3DSearch, Austria, 2007.

15. Pecheanu, E., **Ştefănescu, D.**, *"On Modeling Unstructured Data Collections in Computer Assisted Instruction"*, 5th EuropeanConference on Intelligent Systems and Technologies (ECIT'2008) 2008.

16. Tudorie, C., **Ştefănescu, D.**, *"Special Cases of Relative Object Qualification. Using the AMONG Operator"*, In Proceedings of ECIT2008 – 5th European Conference on Intelligent Systems and Technologies, Iasi, Romania, July 10-12, 2008

17. Pecheanu, E., **Ştefănescu, D.**, Popescu, F., *"A Solution for Modeling the Instructional Content in Computer Assisted Learning"*, Proc. of the 6th International Conference on New Horizons in Industry, Business and Education – NHIBE 2009, Santorini, Greece, I, ISBN 978-960-88785-8-7, pp 171-175, 2009.

Articole publicate în buletine ale simpozioanelor naționale

1. Dumitriu, L., Segal, C., Pecheanu, E., **Ştefănescu, D.**, *"O nouă abordare în extragerea de reguli de asociere"*, Conferința Națională CITTI'2000, Constanța, pp.13-18., ISBN 973-8082-10-2, 2000.

2. **Ştefănescu, D.**, Pecheanu, E., "*Strategii Pedagogice de Grup folosite în Sistemele Inteligente de Instruire"*, Simpozionul Medii virtuale-Ontologie, Cognitivism şi Paradigma Lingvistică-2001, Universitatea Politehnică Bucureşti, www.racai.ro/ANUNT/programv.htm, 2001.

3. Pecheanu, E., **Ştefănescu, D.**, "*Stilul Cognitiv Individual şi Implicaţiile sale în Elaborarea Sistemelor de Instruire Asistată Bazate pe Tehnologia W*WW", Simpozionul Medii virtuale-Ontologie, Cognitivism şi Paradigma Lingvistică-2001, Universitatea Politehnica Bucureşti, 2001, www.racai.ro/ANUNT/programv.htm, 2001.

4. Popescu, F., **Ştefănescu, D.**, Pecheanu, E., Istrate, A., *"Noi soluţii pentru dezvoltarea învăţământului asistat de calculator in România"*, Simpozionul Tehnologii educaţionale pe platforme electronice în învăţământul ingineresc - TEPE 2003, Bucureşti, Universitatea Tehnică de Construcţii Bucureşti, CD + ISBN 973-8165-44-X, 2003.

5. Cocu, A., Pecheanu, E., **Ştefănescu, D.**, *"Modalități de personalizare în e-learning"*, pp 103-108, in volumul *Conferintei Nationale de invatamint Virtual, editia I*, Bucuresti, ISBN 973-575-822-9, 2003.

6. Cocu, A., **Ştefănescu, D.**, *"Implementarea modulului de evaluare a studentului prin standardul IMS"*, Conferința Națională de Învățământ Virtual, CNIV - 2004, Virtual Learning – Noi tehnologii de e-learning, Bucuresti, pp. 43-48, ISBN 973-575-947-0, 2004.

7. **Ştefănescu, D.**, Cocu, A., *"Rolul strategiilor pedagogice de grup folosite în sistemele inteligente de instruire"*, Conferința Națională de Învățământ Virtual, CNIV – 2004, Virtual Learning – Noi tehnologii de e-learning, Bucuresti, 2004.

8. **Ştefănescu, D.**, Cocu, A., Segal, C., Dumitriu, L., *"Strategii pedagogice de grup folosite în sistemele educaționale"*, Conferința Națională de Învățământ Virtual, CNIV - 2005, Virtual Learning – Software & management educational, Bucuresti, pp. 123-130, ISBN 973-737-097-X, 2005.

9. **Ştefănescu, D.**, Cocu, A., *"Grafuri conceptuale aplicate în dezvoltarea unui sistem autor"*, Conferința Națională de Învățământ Virtual, CNIV - 2005, Virtual Learning – Software & management educational, Bucuresti, pp. 115-122, ISBN 973-737-097-X, 2005.

10. **Ştefănescu, D.**, Pecheanu, E., Bumbaru, S., Novac-Ududec, C., *"Computer-assisted education in universities: a comparative study of open-source e-learning platforms"*, Inter-Ing 2007, Scientific Conference with international participation: Interdisciplinarity in engineering, Editura Universitatii „Petru Maior" Targu-Mures, pg. VI-2-1 – VI-2-6, ISSN 1843 - 780X, 2007.

11. **Ştefănescu, D.**, Pecheanu, E., Bumbaru, S., Buraga, S.C., *"Authoring instructional strategies in intelligent educational systems: a model based on conceptual graphs"*, Inter-Ing 2007, Scientific Conference with international participation: Interdisciplinarity in engineering, Editura Universitatii „Petru Maior" Targu-Mures, pg. VI-3-1 – VI-3-6, ISSN 1843-780X, 2007.

Cărți de specialitate publicate

1. Segal, C, **Ştefănescu, D.**, *Sisteme inteligente de diagnoză,* Editura Tehnică, ISBN 973-31-3068-5, 210 pag., 2002.

2. **Ştefănescu, D.**, Pecheanu, E., Istrate, A., 2002, *Sistemul Ariadne - Cadru pentru dezvoltarea, gestionarea şi utilizarea materialelor instrucţionale*, http://www.bsufonline.org/lite/uv/, format electronic, Buletinul Laboratorului pentru

Tehnologia Informatiei in Educatie, Volumul 1, Numarul 1, Buletin informativ realizat in cadrul Proiectului uniSMART, finantat de Ministerul Educatiei si Cercetarii prin Programul INFOSOC, http://www.bsufonline.org/lite/tehnologie&educatie, 2002.

Proiecte de cercetare-dezvoltare pe bază de contract

Proiecte internaţionale

1. 1999-2000 - Contract nr 250/1999-2000, tema: *Configurarea repetitivă și automată de la distanță a echipamentelor SC3002*, beneficiar: CSTELECOM FRANTA, responsabil Segal, C. membri: Dumitriu, L., Pecheanu, E., **Ştefănescu, D.**, ș.a., valoare: 44.000$
2. 1999-2000 - Act Adițional 1 la Contract 250/1999-2000, tema: *Modul de administrare MIB proprietar echipamente SC300*, beneficiar CSTELECOM FRANTA, responsabil Segal, C., membri: Dumitriu, L., Pecheanu, E., **Ştefănescu, D.**, Istrate, A., ș.a., valoare: 14.000$
3. 1998-2000 - Contract nr 228/1998-2000, tema: *Ameliorarea funcțiilor de configurare oferite de echipamentele de comunicație SC300 prin ecrane de configurare de sinteză și servicii de configurare repetitivă*, beneficiar: CSTELECOM FRANTA, responsabil Segal, C., membri: Dumitriu, L., Pecheanu, E., **Ştefănescu, D.**, Istrate, A., ș.a. , valoare: 25.000$
4. 1998-2000 - Act Adițional 1 la Contract 228/1998-2000, tema: *Modul de administrare MIB proprietar echipamente SC300*, beneficiar CSTELECOM FRANTA, responsabil Segal, C., membri: Dumitriu, L., Pecheanu, E., **Ştefănescu, D.**, Istrate, A., ș.a., valoare: 14.000$
5. 2001-2004 - Contract cu Ministerul Educației din Elveția, DANU CH/DC – SCOPES/2001-2004, tema: *Program pentru dezvoltarea unui sistem de educaţie asistată* în colaborare cu Fondul Naţional Elveţian, responsabil Popescu, F., membri: Istrate, A., Pecheanu, E., **Ştefănescu, D.**, ș.a.
6. 2010-2012 - Contract Leonardo da Vinci, 2010–1–PL1–LEO05–11472, tema : *SkillsUp - Supporting system for nonformal learning for low-skilled workers,* responsabil Pecheanu, E., membri: **Ștefănescu, D.**, Cocu, A., Istrate, A., ș.a., Coordonator proiect: Institutul Naţional de Cercetare, Radom, Polonia, Valoare partener român : 76.000 Euro, Valoare Proiect : 400.000 Euro

Proiecte naționale

1. 2001-2002 - Contract MCT şi Academia Română INF-20/2001, proiect C1-142/2001-2002, tema: Societatea informaţională INFOSOC - *Sistem informatic educaţional prin Internet - biblioteci virtuale - e-book,* Mînzu, V., membri: Ceangă, E., Bumbaru, S., Novac, C., Pecheanu, E., **Ştefănescu, D.**, Istrate, A., ş.a., valoare : 2,2 miliare lei

CITĂRI ALE AUTORULUI

Citări în cărţi publicate în edituri internaţionale de renume

Articol citat: *Pecheanu E., Segal C. and **Ştefănescu D.**, "Content modeling in Intelligent Instructional Environment"*. Lecture Notes in Artificial Intelligence, Vol. 3190. Springer-Verlag, Berlin Heidelberg, New Jork (2003) 1229–1234

1. **Citat în**: *"User-System Interaction for Redundancy-Free Knowledge Discovery in Data"*, *Lehn, R., Briand, H., Guillet, F.*
   Book Series: Studies in Computational Intelligence, Vol 127, ISSN 0302-9743, Pp 463-479, 2008

2. **Citat în**: *"Diagnostic Tests Based on Knowledge States"*, *Encheva, S., Tumin, S.* Source: Computational Collective Intelligence. Technologies and Applications, Lecture Notes in Computer Science, 2010, Volume 6423/2010, 133-141, DOI:10.1007/978-3-642-16696-9_15

3. **Citat în**: *"A Tutoring System Discovering Gaps in the Current Body of Students' Knowledge"*, *Encheva, S., Tumin, S.*
   Source: Computational Collective Intelligence. Technologies and Applications, Lecture Notes in Computer Science, 2010, Volume 6423/2010, 133-141, DOI:10.1007/978-3-642-16696-9_15

4. **Citat în**: "Evaluation *of Learning Outcomes", Encheva, S.*
   Source: Advances in Web-Based Learning – ICWL 2010 , Lecture Notes in Computer Science, 2010, Volume 6483/2010, 72-80, DOI:10.1007/978-3-642-17407-0_8

5. **Citat în: "***Evaluation of Concepts Understanding", Encheva, S., Tumin, S.*
   Source: Advances in Web-Based Learning - ICWL 2010, 9th International Conference, Shanghai, China, 2010, Proceedings, Series: Lecture Notes in Computer Science, Vol. 6483, Luo, X.; Spaniol, M.; Wang, L.; Li, Q.; Nejdl, W.; Zhang, W. (Eds.), 1st Edition., 2010, XIV, 412 p., Softcover, ISBN: 978-3-642-17406-3

6. **Citat în**: "*Weak Orderings of Knowledge States", Encheva, S., Tumin, S.*
   Source: Technological Developments in Networking, Education and Automation, Khaled Elleithy, Tarek Sobh, Magued Iskander, Vikram Kapila, Mohammad A. Karim and Ausif Mahmood, Eds. ISBN: 978-90-481-9150-5, pp 7-13 Springer Verlag London,NewYork, Springer Science+Business, 2010

7. **Citat în**: "*Dominance Relations in Rough Sets Approximations for Assessing Students Knowledge", Entcheva, S*., Recent Researches in Artificial Intelligence, Knowledge Engineering and Data Bases, ISBN 978-960-474-273-8, WSEAS Press 2011, pp.162-168
   Source: Proc. of the 10th WSEAS Int. Conf. on Artificial intelligence, knowledge engineering and data bases, AIKED'11, Stevens Point, USA, 2011, ISBN: 978-960-474-273-8, ACM Digital Library,
   dl.acm.org/citation.cfm?id=1959485.1959516&coll=DL&dl=GUIDE&CFID=89281639&CFTOKEN=15089107

8. **Citat în**: *"Complete Graphs and Orderings*", *Encheva, S.*
   Source: Book Series: Communications in Computer and Information Science, Vol 188, pp 309-319, 2011

Articol citat: *Tudorie C.***,** *Frangu, L.,* ***Ştefănescu, D.****, "Automatic Modelling of the Linguistic Values for Databases Fuzzy Querying"* (2007), The Annals of "Dunărea de Jos" University of Galaţi

9. **Citat în**:*"Automatic and Incremental Generation of Membership Functions", Derbel, I., Ounelli H.*

Source: Artificial Intelligence and Soft Computing, Book Series: Lecture Notes in Computer Science, Volume: 6113 ISSN 0302-9743 Pages: 97-104 Published: 2010

Citări în articole publicate în reviste cotate ISI

Articol citat: *Pecheanu E., Segal C., **Ştefănescu D**., "Content modeling in intelligent instructional environments"*, Lecture Notes in Artificial Intelligence, Knowledge-Based Intelligent Information And Engineering Systems, Pt 2, Proceedings Vol. 2774 pag 1229-1234, ISSN 0302-9743, LNCS-LNAI ISBN 3-540-40804-5 Springer Verlag Berlin, 2003

1. **Citat în:** *"Cooperative shared learning objects in an intelligent Web-based tutoring system environment", Encheva S., Tumin S.*, 2nd International Conference on Cooperative Design, Visualization, and Engineering, SEP 18-21, 2005 Palma de Mallorca, Spain
   Source: Cooperative Design, Visualization, And Engineering, Proceedings Book Series: Lecture Notes In Computer Science, Volume: 3675 Pages: 227-234 Published: 2005 *(Journal ISI quoted)*
2. **Citat în:** *"Specific aspects of training IT students for modeling pulses in physics", Podoleanu A, Toma C, Morarescu C, et al.*, International Conference on Computational Science and Its Applications (ICCSA 2005), May 09-12, 2005 Singapore
   Source: Computational Science And Its Applications - Iccsa 2005, Pt 3 Book Series: Lecture Notes In Computer Science Volume: 3482 Pages: 556-562 Published: 2005 *(Journal ISI quoted)*
3. **Citat în:** "*Automated discovering of what is hindering the learning performance of a student*", *Encheva S., Tumin S.,* 8th Asia-Pacific Web Conference and Workshops (APWeb 2006), jan. 16-18, 2006 Harbin, Peoples R China
   Source: Frontiers Of WWW Research And Development - APWEB 2006, Proceedings Book Series: Lecture Notes In Computer Science, Volume: 3841 Pages: 521-531 Published: 2006 *(Journal ISI quoted)*
4. **Citat în**: *"Enabling Self-Organization of the Educational Content in Ad Hoc Learning Networks", Șușnea, I., Vasiliu, G., Mitu, D.E.*, In *Studies in Informatics and Control*, ISSN 1220-1766 vol. 22(2), 2013, pp. 143-152, published: jun 2013

Citare în articole publicate în ISI Proceedings sau în reviste BDI

Articol citat: **Pecheanu E., Segal C., Ștefănescu, D.**, "Content modeling in Intelligent Instructional Environment - Lecture Notes in Artificial Intelligence, Vol. 3190. Springer-Verlag (2003), pp 1229–1234

1. **Citat în**: *"A monitoring system discovering gaps in the current body of students' knowledge", Encheva S., Tumin, S.*, 3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI 2006), jun 07-09, Athens, Greece *(ISI Proceedings)*
   Source: Artificial Intelligence Applications and Innovations, Book Series: International Federation for Information Processing, Vol. 204, Pages 442-449, Published: 2006
2. **Citat în**: *"Decision Support Systems in Logistics", Encheva S., Kondratenko Y., Solesvik M.Z., et al.*, International Electronic Conference on Computer Science, nov. 30-dec. 10, 2007, ELECTRNETWORK
   Source: International Electronic Conference On Computer Science

Book Series: AIP Conference Proceedings Volume: 1060 Pages: 254-256 Published: 2008 *(ISI Proceedings)*

3. **Citat în:** *"On Facilitating the Process of Providing Expert Advises Applying Association Rules", Encheva S., Tumin S.*, International Conference on Telecommunicatios and Networking/ International Conference on Industrial Electronics, Technology and Automation, May 31, 2007 Univ Bridgeport, Bridgeport, CA
   Source: Novel Algorithms And Techniques In Telecommunications, Automation And Industrial Electronics Pages: 242-247 Published: 2008 *(ISI Proceedings)*

4. **Citat în:** *"Non-classical logic in an intelligent assessment sub-system", Encheva S., Kondratenko Y., Tumin S., et al.*, International Conference on Computational Science and Its Applications (ICCSA 2007), AUG, 26-29, 2007 Kuala Lumpur, MALAYSIA
   Source: Computational Science and Its Applications - ICCSA 2007, Pt 1, Proceedings Book Series: Lecture Notes In Computer Science,Volume: 4705 Pages: 305-314 Published: 2008 *(ISI Proceedings)*

5. **Citat în:** ***"Preference orderings for regulatory concepts in environmental management",*** *Encheva, S.*
   Source: Proc. of the 9th WSEAS Int. Conf. on Telecommunications and informatics, Tele Info 2010, ISBN: 978-954-92600-2-1, Stevens Point, USA, 2009, Indexata ACM Digital Library, http://portal.acm.org/citation.cfm?id=1844648.1844679

6. **Citat în:** *"Hints, learning styles, and learning orientations", Encheva, S., Tumin, S.,*
   Source: Proceedings of the 9th WSEAS Int. Conference on Telecommunications and informatics, Tele Info 2010, ISBN: 978-954-92600-2-1, Stevens Point, USA, 2009 Indexata ACM Digital Library, http://portal.acm.org/citation.cfm?id=1844648.1844680

7. **Citat în:** *"Evaluation of Help Functions Based on Ordered Sets", Encheva, S., Tumin, S.*
   Source: Proc. of the 8th WSEAS Int. Conf. on education and educational technology, ISBN: 978-960-474-128-1 ISSN: 1790-5109 , pp 132-136 , Genova 2009

8. **Citat în**: *"Rough Sets Approximations for Learning Outcomes", Encheva, S., Tumin, S.*
   Conference**:** 2nd Int. Conference on Adv Sci and Technol/4th Int Conf on Informat Security and Assurance/2nd Int Conf on Adv Commun and Networking/Int Conf on Ubiquitous Computing and Multimedia Miyazaki, JAPAN, 2010**,** Book Series: Communications in Computer and Information Science Vol**.** 75 Pp 63-72 **,** 2010 -
   Citare detectata in 2011

9. **Citat în**: *"Diagnostic Tests Based on Knowledge States", Encheva, S., Tumin, S.*
   Conference**:** 2nd International Conference on Computational Collective Intelligence: Technologies and Applications, Kaohsiung, TAIWAN, 2010 Source: COMPUTATIONAL COLLECTIVE INTELLIGENCE: TECHNOLOGIES AND APPLICATIONS,
   Book Series**:** Lecture Notes in Artificial Intelligence Vol. 6423 Pp133-141 **,** 2010
   Citare detectata in 2011

10. **Citat în***: "On Indiscernibility in Assessments",* Encheva, S., International Conference on U- and E-Service, Science and Technology**,** SOUTH KOREA, Source: U- AND E-SERVICE, SCIENCE AND TECHNOLOGY Book Series: Communications in Computer and Information Science Volume: 124 Pages: 39-47 Published**:** 2010
    Citare detectata in 2011

11. **Citat în**: *"Evaluation of Concepts Understanding", Encheva, S.,* The 6th WSEAS/IASME Int. Conference on Educational Technologies (EDUTE 10) **,** TUNISIA, 2010
    Book Series: Recent Advances in Computer Engineering, Pp 178-182, 2010
    Citare detectata in 2011

12. **Citat în**: *"Automated decision making based on weak orderings", Enheva, S.*
    Source: Int. Journal of Intelligent Information and Database Systems, Vol 6 Issue 1, Geneva, 2012, Doi: 10.1504/IJIIDS.2012.045114, ACM Digital Library, dl.acm.org/citation.cfm?id=2125250.2125252&coll=DL&dl=GUIDE&CFID=89281639&CFTOKEN=15089107

Articol citat: Pecheanu, E., Dumitriu, L., **Ștefănescu, D.**, & Segal, C., *"A framework for conceptually modelling the domain knowledge of an instructional system",* In *International conference on computational science* (Vol. 1, pp. 199–206), 2006.

13. **Citat în**: *"Formal Concept Analysis in knowledge processing: A survey on models and techniques"*, *Poelmans, J., Kuznetsov, S.O., Ignatov, D.I., Dedene, G.,* Elsevier ScienceDirect – Expert Systems with Applications - Volume 40, Issue 16, Pages 6225-6692, nov 2013, http://www.sciencedirect.com/science/article/pii/S0957417413002935 http://dx.doi.org/10.1016/j.eswa.2013.05.007

Articol citat: Pecheanu, E., **Ștefănescu, D.**, Dumitriu, L., Segal, C., "*Methods to evaluate open source learning platforms",* (EDUCON), 2011 IEEE, 2011 - ieeexplore.ieee.org

14. **Citat în** *"A Comparative Study of Moodle, Sakai, Drupal and Blackboard's"*, *Goumin, D.* Chinese Scientific Journals Database - Humanities and Social Sciences - Education - Educational Technology, "China Educational Technology" on page 6 2013 - http://www.cqvip.com/main/none.aspx?lngid=45986404

15. **Citat în** "*Using FCA for Modelling Conceptual Difficulties in Learning Processes", Priss, U., Riegler, P., Jensen, N.*
    In: Domenach; Ignatov; Poelmans (eds.), Contributions to the 10th International Conference on Formal Concept Analysis (ICFCA 2012), 2012, p. 161-173
    http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.416.3070, *ceur-ws.org/Vol-876*

Citări în teze de doctorat:

Articol citat: *Cocu Adina, Pecheanu Emilia,* ***Ştefănescu Diana****, "Modalităţi de personalizare în e-learning"*, în vol. Conferinţa Naţională de Învăţământ Virtual - CNIV'2003, Bucureşti, pp. 109-112, ISBN 973-575-822-9, pp. 109-112, 2003.

1. **Citat în teza:** *"Contribuţii la tehnicile de reprezentare a cunoştinţelor în sistemele educaţionale inteligente"*, susţinută de Cocu Adina în 2009

Articol citat: *Cocu Adina,* ***Ştefănescu Diana****, "Student Evaluation Implementation using IMS Standard"*, National Conference in Virtual Learning, Bucharest – CNIV'2004, ISBN 973-575-947-0, pp. 43-49, 2004

2. **Citat în teza:** *"Contribuţii la tehnicile de reprezentare a cunoştinţelor în sistemele educaţionale inteligente"*, susţinută de Cocu Adina în 2009

Articol citat: *Cocu Adina, **Ştefănescu Diana**, Uncertainty management using bayesian networks in student knowledge diagnosis"*, The Annals of "Dunarea de Jos" University of Galati, Fascicle, pp. 29-32, ISSN 1221-454X, 2005

3. **Citat în teza**: *"Contribuţii la tehnicile de reprezentare a cunoştinţelor în sistemele educaţionale inteligente"*, susţinută de Cocu Adina în 2009

Articol citat: ***Ştefănescu Diana**, Pecheanu Emilia, Cocu, Adina., "Pedagogical knowledge model based on conceptual graphs and ontology"*, The Annals of "Dunarea de Jos" University of Galati, Fascicle III, pp. 11-17, ISSN 1221-454X, pp. 12-16, 2005.

4. **Citat în teza:** *"Contribuţii la tehnicile de reprezentare a cunoştinţelor în sistemele educaţionale inteligente"*, susţinută de Cocu Adina în 2009

Citări în lucrări/studii publicate în volumele unor conferinţe internaţionale (din ţară sau din străinătate)

Articol citat: *Pecheanu E., **Ştefănescu D**.,Buraga, S.C., Istrate, A., "Integrating Hypermedia Objects in an Intelligent Tutoring System", Proc. of the 11-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS 2001, Galati, pp. 170-178., ISBN 973-8139-98-8, 2001.*

1. **Citat în:** *"Developing Agent-Oriented E-Learning Systems", Buraga, S.C.,* Proceedings of The 14th International Conference on Control Systems And Computer Science – vol.II, I. Dumitrache and C.Buiu (eds.), Politehnica Press, Bucharest, 2003

Articol citat: ***Ştefănescu D**., Pecheanu E., Buraga, S.C., "Pedagogical Agents in Intelligent Tutoring Systems", Proc. of the 11-th International Symposium on Modeling, Simulation and Systems Identification, SIMSIS 2001, Galati, pp. 178-183, ISBN 973-8139-98-8, 2001.*

2. **Citat în:** *"Developing Agent-Oriented E-Learning Systems", Buraga, S.C.,* Proceedings of The 14th International Conference on Control Systems And Computer Science – vol.II, I.Dumitrache and C.Buiu (eds.), Politehnica Press, Bucharest, 2003

Articol citat: *Pecheanu E., Segal, C., **Ştefănescu D**., "Content Modeling in Intelligent Instructional Systems"*, KES2003, Oxford, UK, in press, 2003.

3. **Citat în:** *"e-Prelegere: organizarea şi distribuirea resurselor educaţionale",Pecheanu, E., Dumitriu, L., Segal, C.,* Source:Conferinţa Naţiomală de Învăţământ Virtual, Secţiunea Software Educaţional, Bucuresti, 2003

Articol citat: *Pecheanu, E., **Ştefănescu D**., Istrate, A., "An Extended Structural Model for Intelligent Instructional Systems"*, The Romanian Symp. on Computer Science – ROSYCS 2004, July 2004, Iasi, Scientific Annals of the "A.I. Cuza" University of Iasi, Tome XV, pp. 167-177, 2004

4. **Citat în:** *"Heuristic Evaluation Of Web-Based Intelligent Tutoring Systems", Andone, I, Sireteanu, A.,* Source: Proceedings of the 4th International Conference e-learning and software for education, Bucuresti, 2008

Articol citat: Pecheanu, E., **Ștefănescu, D.**, Istrate, A., & Dascalescu, D.**,** *"Conceptually Modeling the Domain Knowledge for Assisted Learning in an IT Discipline"*. International Conference on Computer Aided Learning in Engineering Education, pp. 215-220, 2004.

5. **Citat în**: Vivanet, G., *"Modelli descrittivi di oggetti per l'apprendimento: stato dell'arte e implicazioni per la progettazione didattica"*, Proceedings of Didamatica 2011 , ISBN 9788890540622, Torino, Italy, 2011.

7. REFERINȚE BIBLIOGRAFICE[55]

1. Cucoş, C., *Educația. Experiențe, reflecții, soluții.* Editura POLIROM, Iaşi, ISBN 978-973-46-3232-9, 2013.
2. Popescu, F., Pecheanu, E., **Ştefănescu, D.**, Istrate, A., *New Solutions in Building Integrated Educational Frameworks*, Proc. of the 3th International Conference "New Horizons in Industry and Education" – NHIE''03, Santorini, Grecia, pp 270-276, ISBN 9609-85316-7-5, 2003.
3. Morrison, G. R., Steven, M., R., Howard, K.K., *Designing effective instruction*, 7th, Wiley Global Education Ed., ISBN 1118473590, 9781118473597, 2012.
4. Paquette, G., *Instructional Engineering in Networked Environments*, Preiffer/ Wiley Publisher, ISBN 0-7879-6466-2, 2004.
5. Carliner, S., Shank, P., *The e-learning handbook: past promises, present challenges*, San Francisco: Pfeiffer, ISBN: 978-0-7879-7831-0, 2008.
6. Stahl, G., Koschmann, T., Suthers, D., *Computer-supported collaborative learning: An historical perspective*, In R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences (pp. 409-426). Cambridge, UK: Cambridge University Press, ISBN 9780521607773, 2006.
7. Paquette, G., *Chapter 56: Technology-Based Instructional Design - Evolution and major Trends*, In book: Handbook of Research on Educational Communications and Technology, Fourth Edition, The 4th Edition, Springer Academic Publisher, J. Michael Specto, M. David Merrill, J. Ellen, M.J.Bishop Eds., ISBN 978-1-4614-3184-8, 2013.
8. Orsucci, F., *Changing Mind: Transitions in Natural and Artificial Environments*, Vol.9, Studies on nonlinear phenomena in life sciences, World Scientific, ISBN 9812380272, 9789812380272, 2002.
9. [9]***, http://psychology.about.com/od/psychology101/u/psychology-theories.htm#s1, (u.v. mai 2014).
10. Schunk, D., *Learning Theories: An educational Perspective (6th Edition),* Pearson Education. Inc., publishing as Allyn&Bacon, Boston, ISBN-13: 978-0137071951, 2011.
11. Ormrod, J.E., *Human Learning (6th Edition)*, Pearson Publisher, ISBN-10: 0132595184 | ISBN-13: 978-0132595186, 2011.
12. Anderson, J.R., *Cognitive Psychology & its Implications*, The 7th Edition, Worth Publishers, NY., p.241, ISBN-10: 1429219483, 2009.
13. Lefrancois, G.R., *Theories of Human Learning: What the Professor Said*, The 6th Edition, Cengage Learning Publisher, ISBN-10: 1111829748, ISBN-13: 978-1111829742, 2011.
14. Merrill, M.D., *Instructional Strategies and Learning Styles: Which takes Precedence?*, In R. A. Reiser & J. V. Dempsey (Eds.), Trends and Issues in Instructional Technology, pp. 99-106, Columbus, OH: Prentice Hall, Print ISSN: 0143-0343; Online ISSN: 1461-7374, 2002.
15. [15]***, Instructional Design, URL: http://www.instructionaldesign.org/theories/index.html
16. [16]***, *Constructivism as a Paradigm for Teaching and Learning*, Thirteen Ed Online: URL:http://www.thirteen.org/edonline/concept2class/ constructivism/index.html.

[55]Pentru resursele online – u.v. (ultima vizualizare) mai 2014

17. Conole, G., Oliver, M., *Contemporary perspectives in e-learning research: themes, methods and impact on practice*, London: Routledge, ISBNI0 0-415-39393-0, 2007.

18. Self, J.A., *Computational mathetics: towards a science of computer based learning system design*, Computer Based Learning Unit, University of Leeds, URL:http://telearn.archives-ouvertes.fr/docs/00/19/05/03/PDF/Self1995.pdf

19. **Ştefănescu, D.**, Pecheanu, E., Istrate, A., Segal, C., *An Object Oriented Approach to Produce Hypermedia Educational Software*, Internet as a vehicle for teaching, Proceedings of the International Romanian Internet Learning Workshop RILW 2001, Miercurea-Ciuc, pp 182-187, 2001.(Impact Factor: 1.16)

20. Allen's, M., *e-Learning Annual*, Pfeiffer Publ., ISBN 978-0-7879-8743-5, ISSN 1046-333-X, USA, 2008

21. Nicholson, P., *A history of e-learning*, In Computers and Education: E-learning, from Theory to Practice, Springer, The Netherlands, ISBN 978-1-4020-4913-2 (HB), 2007.

22. Pecheanu, E., Dumitriu, L., **Ştefănescu, D.**, Bumbaru, S., *Mentor – a Computer-Based Instructional and Assessing System*, Proc. of the 5th Internat. Conference on Computer Aided Engineering Education, CAEE'99, Sofia, Bulgaria, pp. 274- 279, 1999.

23. **Ştefănescu, D.**, Bumbaru, S., Ariton, V., *Hybrid Systems for Assisted Learning*, The 9thInternational Symposium on Modeling, Simulation and Systems' Identification, SIMSIS '96, pg. 372-381, ISSN 1221-454X, 1996.

24. Pecheanu, E., Segal, C., **Ştefănescu, D.**, *Content Modeling in Intelligent Instructional Environments,* Proc. of the 7th International Conf. on Knowledge-Based Intelligent Information Engineering, KES'2003, Oxford UK, pp. 1229-1235, ISSN 0302-9743, LNCS-LNAI ISBN 3-540-40804-5 Springer Verlag Berlin, **Journal ISI Quoted (Web of Science)**, 2003.

25. Pecheanu, E., **Ştefănescu, D.**, Buraga, S.C., Istrate, A., *Integrating Hypermedia Objects in an Intelligent Tutoring System*, Proc. of the 11thInternational Symposium on Modeling, Simulation and Systems' Identification, SIMSIS 2001, Galati, pp. 170-178., ISBN 973-8139-98-8, 2001.

26. Hsiao, I-H., Guerra, J., Parra, D., Bakalov, F., König-Ries, B., Brusilovsky, P., Comparative Social Visualization for Personalized E-learning, In International Working Conference on Advanced Visual Interfaces, 2012

27. **Ştefănescu, D.**, Pecheanu, E., Buraga, S.C., *Integrating a Virtual Library in an Educational System*, Internat. Conf. on Qualification and Training in Information and Communication Technologies - QTICT'02, Galaţi, CD+ISBN 973-8409-48-9, 2002

28. Popescu, F., **Ştefănescu, D.**, Pecheanu, E., Istrate, A., *Noi soluții pentru dezvoltarea învățământului asistat de calculator în Romania*, Simpozionul Tehnologii Educaționale pe Platforme Electronice în Învățământul Ingineresc- TEPE 2003, București, CD + ISBN 973-8165-44-X, 2003.

29. Paiva, A., Dias, J., Sobral, D., Aylett, R., Woods, S., Hall, L.,Zoll, C., *Learning By Feeling: Evoking Empathy With Synthetic Characters*, *Applied Artificial Intelligence Journal, Vol. 19 (3-4), pp. 235-266, ISSN 0883-9514 (*Print), 1087-6545 (Online), 2005.

30. Pecheanu, E., **Ştefănescu, D.**, *Stilul Cognitiv Individual si Implicatiile sale in Elaborarea Sistemelor de Instruire Asistata Bazate pe Tehnologia WWW*, Simpozionul - Dezbatere interdisciplinara: Medii virtuale-Ontologie, Cognitivism si Paradigma Lingvistica, Bucuresti, 2001.

31. Pecheanu, E., **Ştefănescu, D.**, Segal, C., Popescu, F., *Assisted-Learning and Individual Cognitive Style*, Proc. of the 3th Internat. Conference "New Horizons in Industry and Education"– NHIE''03, Grecia, pp. 258-264, ISBN 9609-85316-7-5, 2003.

32. Buraga, S.C., **Ştefănescu, D.**, Pecheanu, E., *Modeling Relations between Resources of an Internet Teleconferencing System*, Proc. of the 11-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS Galati, pp. 68-74., ISBN 973-8139-98-8, 2001.

33. **Ştefănescu, D.,** Pecheanu, E., Dumitriu, L., *A Hybrid Approach to Dynamic Course Generation on the WWW*, The Annals of "Dunarea de Jos" of Galati, Fascicle III (**Revista tip B, cod CNCSIS 482**), pp. 60-64, ISSN 1221-454 X, 2000.

34. Bumbaru, S., Pecheanu, E., **Ştefănescu, D.**, *MENTOR – A Project for Computer-Aided Educational Systems*, Proc. of the International Symposium "The Role of Academic Education and Research in the Development of Information Society", Bucuresti, pp. 89-94, 1999.

35. **Ştefănescu, D.**, Bumbaru, S., *The Instructional Theory*, The 9th International Symposium on Modeling, Simulation and Systems Identification, SIMSIS '96, pg. 382-387, ISSN 1221-454X, 1996.

36. Pecheanu, E., Dumitriu, L., **Ştefănescu, D.**, Istrate, A., *Developing Environments for Distance Engineering Education*, prezentare poster la conferinta 7th World Conference on Continuing Engineering Education WCCEE'98, Torino, Italia, 1998.

37. Pecheanu, E., **Ştefănescu, D.**, Dumitriu, L., Istrate, A., *Architecture of an Intelligent Tutoring System*, The 7thInternational Symposium on Automatic Control and Computer Science, SACCS 2001, Iasi, CD+ISBN 973-8292-10-7, 2001.

38. Safdar, A.S., Memon, A., G., Soomro, S., *Pre-Generation of Student Module in Intelligent Tutoring System*, Journal of Information & Communication Technology, Vol. 5, No. 1, Spring 2011, pp.12-21, 2011.

39. **Ştefănescu, D.**, Pecheanu, E., Dumitriu, L., *Student Modeling in WWW-based Tutoring Systems – An Overview*, lucrare prezentata ca tutorial la International Confernce RILW 2001, Miercurea-Ciuc, URL:http://rilw.emp.paed.uni-muenchen.de/2001/info.html, 2001.

40. Pecheanu, E., **Ştefănescu, D.**, Istrate, A., Popescu, F., *Student modeling methods in WWW-based tutoring systems*, International Conference on Qualification and Training in Information & Communication Technologies-QTICT'02, Galaţi, CD+ISBN 973-8409-48-9, 2002.

41. Răzmeriţă, L., **Ştefănescu, D.**, Bumbaru, S., Istrate, A., *Student testing and assessment in Computer Based Training Systems*, Proc. of 4thInternational Conference Computer Aided Engineering Education, Krakov, Poland, pg. 223-231, 1997.

42. Răzmeriţă, L., **Ştefănescu, D.**, Bumbaru, S., Istrate, A., *Computer testing and assesing package in Computer Based Training Systems*, Proc. of 3rdInternational Workshop Session: Internet as a Vehicle for Teaching, Kida-Ilieni, Romania, RILW 1998, POLYGON si COMPREX, Cluj, ISBN 973-97403-40, 1998.

43. **Ştefănescu, D.**, Pecheanu, E., Dumitriu, L., Neagu, D., *Student Modelling in Educational Multimedia Environments using Agents*, The Annals of "Dunarea de Jos" of Galati, Fascicle III (**Revista tip B, cod CNCSIS 482**), pp.74-79, ISSN 1221-454X, 1999.

44. Gagné, M.R., Golas, W.W., Wager, K., Keller, J.M., *Principles of Instructional Design (Edition 5),* ISBN-10: 0534582842, ISBN-13 978-0534582845, 2004.

45. Nkambou R., *Intelligent Tutoring Systems*, Educational Technology & Society Vol. 13 No. 1, ISSN 1176-3647, 2010.

46. Clark, R. C., Mayer, R. E., *E-learning and the science of instruction: proven guidelines for consumers and designers of multimedia learning*, San Francisco, CA: Pfeiffer, 2008.

47. **Ştefănescu, D.,** Pecheanu, E., Bumbaru, S., Novac-Ududec, C., *Computer-assisted education in universities: a comparative study of open-source e-learning platforms*, Inter-Ing 2007, Scientific

Conference with international participation: Interdisciplinarity in engineering, Ed. Univ. „Petru Maior" Targu-Mures, pg. VI-2-1 – VI-2-6, ISSN 1843-780X, 2007.

48. **Ştefănescu, D.**, Tudorie, C., Pecheanu, E., *Diversity in Software Support for Computer-Assisted Education*, Proc. of the 8th RoEduNet International Conference on Networking in Education & Research, Galati, Romania, ISBN 978-606-8085-15-9, pp 29-33, **Journal ISI quoted (Web of Science)**, 2009.

49. Pecheanu, E., Dumitriu, L., Segal, C., **Ştefănescu, D.,** *Methods to Evaluate Open Source Learning Platforms*, in Global Engineering Education Conference, EDUCON2011: New Pedagogic Challenges in Engineering Education, Amman, Iordania, IEEE, ISBN: 978-1-61284-642-2, **INSPEC Accession Number:** 12029769, Digital Object Identifier: 10.1109/EDUCON.2011.5773292, pp. 1152 – 1161, 2011.

50. Buraga, S.C., Cioca, M., *Modeling Aspects of Semantic Web-based E-learning Systems*, in C.Oprean et al. (eds.), Proceedings of the 3rd Balkan Region Conference on Engineering Education, "L.Blaga" University Press, Sibiu, ISBN 973–739–101-3, 2005.

51. Devedic, V., *Education and the Semantic Web*, International Journal of Artificial Intelligence in Education 14, pp. 39-65, IOS Press, 2004.

52. Brusilovsky, P., *Adaptive Hypermedia for Education and Training*, In Adaptive Technologies for Training and Education. Cambridge University Press, Cambridge, UK, pp. 46-68. ISBN 9780521769037; 0521769035, 2012.

53. Koper, R., *Current Research in Learning Design*, Educational Technology and Society, 9(1), 13-22, 2006.

54. Pecheanu, E., **Ştefănescu, D.,** Istrate, A, Jâşcanu, V., *On Modeling Adaptive Web-based Instructional Systems*, Proc. of the 8th RoEduNet International Conference on Networking in Education and Research, Galati, Romania, ISBN 978-606-8085-15-9, pp 84-90, **Journal ISI quoted (Web of Science)**,2009.

55. Cocu, A., Pecheanu, E., **Ştefănescu, D.**, *Modalitati de personalizare in e-learning*, in vol. Conferinta Nationala de Invatamint Virtual - CNIV'2003, Bucuresti, pp. 109-112, ISBN 973-575-822-9, 2003.

56. Cocu, A.,**Ştefănescu, D.**, *Uncertainty management using bayesian networks in student knowledge diagnosis*, The Annals of "Dunarea de Jos" University of Galati, Fascicle III (**Revista tip B, cod CNCSIS 482**), pp. 29-32, ISSN 1221-454X, 2005.

57. Dolog, P., Nejdl, W., Chapter 23: *The Adaptive Web: Methods and Strategies of Web Personalization*, , In book Semantic Web Technologies for the Adaptive, Eds. Peter Brusilovsky; Alfred Kobsa; Wolfgang Nejdl, Berlin, Springer, pp. 697-719 (Lecture Notes in Computer Science (LNCS), Vol. 4321), 2007.

58. **Ştefănescu, D.,** Pecheanu, E., *Strategii Pedagogice de Grup folosite în Sistemele Inteligente de Instruire*, Simpozionul - Dezbatere interdisciplinara: Medii virtuale-Ontologie, Cognitivism si Paradigma Lingvistica, Bucuresti, 2001.

59. Chou, C.Y., Chan, T.W., Lin, C.J., *Redefining the learning companion: the past, present, and future of educational agents*, *Computers & Education*, 40, 255-269, 2003

60. Aïmeur E., Frasson, C., Dufort, H., *Co-operative Learning Strategies for Intelligent Tutoring Systems*, International Journal of Applied Artificial Intelligence, Vol 14 (5), pp 465-490, 2000.

61. Tchounikine, P., Rummel, N., McLaren, B.M., Pierre 1, Nikol 2, and Bruce M. 3,4, *Chapter 22 - Computer Supported Collaborative Learning and Intelligent Tutoring Systems*, R. Nkambou et al. (Eds.): Advances in Intelligent Tutoring Systems, SCI 308, pp. 447–463, Springer-Verlag Berlin Heidelberg, 2010.

62. Lai, R.K., Chou, C.Y., Lan, C.H., *Supporting Adaptive Learning Sequences with Agent Negociation*, 12thIEEE International Conference on Advanced Learning Technologies, ICALT 2012, Rome, Italy, pp. 506-508, ISBN 978-1-4673-1642-2, 2012.

63. **Ştefănescu, D.,** Cocu, A., Segal, C., Dumitriu, L., *Strategii pedagogice de grup folosite in sistemele educationale*, Conferința Națională de Învățământ Virtual, CNIV - 2005, Virtual Learning – Software & management educational, București, pp. 123-130, ISBN 973-737-097-X, 2005.

64. Khoualdi, K., Benghezal, R., *Design of an Intelligent Tutor using a Multiagent Approach*, World Academy of Science, Engineering and Technology, International Journal of Social, Human Science and Engineering Vol:1 No:3, 2007.

65. Lavendelis, E., Bicans, J.,*Multi-Agent and Service Oriented Architectures for Intelligent Tutoring System Development,* Scientific Journal of Riga Technical University 2011, Series 5, Volume 43, 2011, pp. 27-36, 2011.

66. Zoll, C., Enz, S., Schaub, H., Aylett, R., Paiva, A., *Fighting Bullying with the Help of Autonomous Agents in a Virtual School Environment, Proc. 7th International Conference on Cognitive Modelling (ICCM-06), Italy, 2006.*

67. Lavendelis, E., Grundspenkis, J., *MIPITS - An Agent based Intelligent Tutoring System*, Proceedings of2ndInternational Conference on Agents and Artificial Intelligence (ICAART 2010) Vol. 2.,Valensia, Spain, pp. 5-13, 2010.

68. **Ştefănescu, D.,** Pecheanu, E., Bumbaru, S., *Using Pedagogical Agents in Intelligent Tutoring Systems*, The 7thInternational Symposium on Automatic Control and Computer Science, SACCS'2001, Iasi, CD+ISBN 973-8292-10-7, 2001.

69. Baylor, A., Kim, Y., *Simulating instructional roles through pedagogical agents*, International Journal of Artificial Intelligence in Education, 15 (5), 95–115, 2005.

70. *Buraga, S.C., Developing Agent-Oriented E-Learning Systems,* Proceedings of The 14th International Conference on Control Systems And Computer Science – vol.II, I.Dumitrache and C.Buiu (eds.), Politehnica Press, Bucharest, 2003

71. Haake, M., *Embodied Pedagogical Agents: From Visual Impact to Pedagogical Implications*, Doctoral Thesis, Dept. of Design Sciences, Lund University, Sweden, 2009

72. Chaffar, S., Frasson , C., *The Emotional Conditions of Learning,* Proc. of the Eighteenth International Florida Artificial Intelligence Research Society Conference, AAAI Press 2005, pp. 201-206, 2005.

73. Gulz, A., Haake, M., *Pedagogical agents – design guide lines regarding visual appearance and pedagogical roles*, Proceedings of the IV International Conference on Multimedia and ICT in Education (M-ICTE2006), Seville, Spain, 2006.

74. **Ştefănescu, D.**, Pecheanu, E., Buraga, S.C., *Pedagogical Agents in Intelligent Tutoring Systems*, Proc. of the 11-th Internat. Symposium on Modeling, Simul.&Syst. Identification, SIMSIS 2001, Galati, pp.178-183, ISBN973-8139-98-8, 2001.

75. Johnson, W., *Interaction tactics for socially intelligent pedagogical agents*, In Proc. of the 8th International Conference on Intelligent User Interfaces, New York, ACM Press, pp. 251-253, 2003.

76. Gratch, J., Marsella, S., *Lessons From Emotion Psychology For The Design Of Lifelike Characters*, Applied Artificial Intelligence, Vol. 19(3-4), pp. 215-233, 2005.

77. Frasson, C., Martin, L., Gouardères, G. & Aïmeur, E., *LANCA : a distance Learning Architecture based on Networked Cognitive Agents*, ITS-98 Conference , Fourth International Conference on Intelligent Tutoring Systems , San Antonio, Texas, pp. 596-604, 1998.

78. [78]*** Peddy Agent, *URL: http://ldt.stanford.edu/~slater/pages/agents/pages/iapa_1.htm*

79. Lester, J., Callaway, C., Grégoire, J., et al., *Animated Pedagogical Agents in Knowledge-Based Learning Environments*, In *Smart Machines in Education: The Coming Revolution in Educational Technology*, pp. 269-298, AAAI/MIT Press, 2001.

80. Rickel, J., and al., *Steve goes to Bosnia: Toward a new generation of virtual humans for interactive experiences*, AAAI Spring Symposionum on Artificial Intelligence and Interactive Entertainment, 2001.

81. Johnson, W.L., Shaw, E., Marshall, A., Labore, C., *Evolution of user interaction: The case of agent Adele*, In Proc. of the 8th International Conference on Intelligent User Interfaces, pp. 93-100, 2003.

82. Murray T., *Eon: Authoring Tools for Content, Instructional Strategy, Student Model, and Interface Design*, Chapter 11 in Murray, Ainswoth&Blessing (eds.), Authoring Tools for Advanced Technology Learning Environments, Kluwer Publisher, 2003.

83. Murray T., *An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art.*, Chapter 17 in Murray, Ainswoth&Blessing (eds.), Authoring Tools for Advanced Technology Learning Environments, Kluwer Publisher, pp. 491-544, 2003.

84. Wagner, S., Weibel, S., *The Dublin Core Metadata Registry: Requirements, Implementation, and Experience*, J. of Digital Information, Vol.6(2), Art 330, 03-17, 2005.

85. [85]*** Dublin Core Metadata Initiative, URL: http://dublincore.org

86. Shayo, C., Olfman, L., *Chapter 12 - The Learning Objects Economy - What remains to Be Done?*, In Human-Computer Interaction and Management Information Systems: Applications, Galletta&Zhang Eds., ISSN 1554-6152, 2006.

87. Friesen, N., *Open Educational Resources: New Possibilities for Change and Sustainability*, International Review of Research in Open and Distance Learning, Vol. 10, N.5, ISSN 1492-3831, 2009.

88. [88]*** Ariadne Foundation, URL: http://www.ariadne-eu.org/

89. **Ştefănescu, D.,** Pecheanu, E., Istrate, A., *Sistemul ARIADNE – Cadru pentru dezvoltarea, gestionarea și utilizarea materialelor intructionale*, on-line, Tehnologie și Educație – Buletin editat în cadrul progr.-lui INFOSOC, Anul 1, Nr. 2, București,URL: http://www.bsufonline.org/lite/tehnologie&educatie, 2003.

90. **Ştefănescu, D.,** Pecheanu, E., *Experience in Courseware Development Using the ARIADNE Approach*, Proc. of the 2nd Balkan Region Conference on Engineering Education, Sibiu, pp. 160-163, ISBN 973-651-673-3, **Proceeding ISI Quoted (Web of Science)**, 2003.

91. Pecheanu, E., **Ştefănescu, D.**, Istrate, A., Dăscălescu, D., *Modeling the Domain Knowledge of an Instructional Environment*, on-line, The 3rd Annual Ariadne Conf., Belgia, URL:http://rubens.cs.kleuven.ac.be/ariadne/CONF2003/intro.html, 2003.

92. [92]*** IMS Global Learning Consortium, URL: http://www.imsglobal.org/

93. Cocu, A., **Ştefănescu, D.**, *Student Evaluation Implementation using IMS Standard*, National Conference in Virtual Learning, Bucharest–CNIV'2004, ISBN 973-575-947-0, pp 43-49, 2004.

94. [94]*** ADL, URL: http://www.adlnet.org/

95. [95]*** SCORM, URL: http://scorm.otg.

96. [96]*** AICC, URL: http://www.aicc.org/joomla/dev/

97. [97]*** IEEE Learning Technology Standards Committee (LTSC), URL: http://ltsc.ieee.org

98. Koper, R., Olivier, B., *Learning Design of Units of Learning*, Educational Technology & Society, vol. 7, pp. 97-111, 2004.

99. Trăuşan-Matu, Şt., *Inteligenţa artificială*, URL:http://www.racai.ro/~trausan/ia.pdf, 2004 (u.v. aug. 2009).

100. Trăuşan-Matu, Şt., *Achiziţia, gestiunea, partajarea şi prelucrarea cunoştinţelor pe web, elemente esenţiale în societatea cunoaşterii*, URL:http://www.racai.ro/INFOSOC-Proiect-Trausan_st_new.pdf, 2004 (u.v. aug. 2009).

101. Guarino, N., Giaretta, P., *Ontologies and knowledge bases, towards a terminological clarification*, In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, IOS Press, 1995.

102. Gruber, T.R., *What is an Ontology?*, URL:http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, (u.v. iulie. 2014).

103. Gruber, T., *Towards principles for the design of ontologies used for knowledge sharing*, International Journal of Human-Computer Studies, 1995.

104. Fox, M.S., Gruninger, M., *Enterprise modeling*, *Artificial Intelligence Magazine*, 19(3):109-121, 1998.

105. Takeda, *Agent organisation and communication with multiple ontologies*, International Journal of Cooperative Information Systems, Vol 4(4), pp.321-337, 1995.

106. Wielinga, B., Schreiber, A., Wielemaker, J., Sandberg, J., *From thesaurus to ontology*, Technical Report, URL:http://www.cs.vu.nl/~guus/.

107. Guarino, N., *Formal Ontologies and Information Systems*, Proc. of Formal Ontologies in Information Systems'98, Trento, Italy, Amsterdam, IOS Press, pp 3-15, 1998.

108. van Heijst, G., Schreiber, A.Th., Wielinga, B.J., *Using Explicit Ontologies in KBS Development*, International Journal Human-Computer Studies, pp. 183-291, 1997.

109. Guarino, N., Vieu, L., Borgo, S., *Formal Ontology for Semanticists - a course*, The 17th European Summer School in Logic, Language and Information, Edinburgh, 2005.

110. Guarino, N., *Helping People (and Machines) Understanding Each Other: The Role of Formal Ontology*, On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, OTM Confederated International Conf., Cyprus, 2004, Proc. 3290, pp. 599, 2004.

111. Schreiber, G., Wielinga, B., Jansweijer, W., *The KAKTUS View on the 'O' Word*, In Proc. of IJCAI95, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 1995.

112. Cristani, M., Cuel, R., *A Survey on Ontology Creation Methodologies*, International Journal: Semantic Web Information Systems 1(2), pp. 49-69, 2005.

113. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., *Understanding top-level ontological distinctions*, Proc. of IJCAI 2001 Workshop on Ontologies and Information Sharing, Seattle, Washington, pp. 26-33, 2001.

114. Mizoguchi, R., *Ontology Engineering Environments*, Handbook on Ontologies, S. Staab and R. Studer (eds), pp. 275-295, 2003.

115. Gómez-Pérez, A., *Ontology Evaluation*, in Handbook on Ontologies, International Handbooks on Information Systems Springer, pp. 251-274, 2004.

116. Chandrasekaran, B., Johnson, T.R., *Generic Tasks and Tasks Structures: History, Critique and New Directions*, pp 234-271, 1999.

117. Catenacci, C., Ciaramita, M., Gil, R., Gangemi, A., Guarino, N., Lehmann, J., *Ontology evaluation: A review of methods and an integrated model for the quality diagnostic task*, TechReport, URL:http://www.loa-cnr.it/Publications.html, (u.v.aug. 2009).

118. Oltramari, A., Gangemi, A., Huang, C., Calzolari N., Lenci A., P. L., *Synergezing ontologies and the lexicon: a roadmap*, Ontologies and the Lexicon, . Cambridge: Cambridge University Press, 2010.

119. Mizoguchi, R., *Tutorial on Ontological Engineering. Part 1: Introduction to Ontological Engineering*, New Generation Computing, 21, pp. 365-384, Springer, 2003.

120. Mizoguchi, R., *Tutorial on Ontological Engineering. Part 2: Ontology Development, Tools&Languages*, New Generation Computing, 22, pp 61-96, Springer-Verlag,, 2004.

121. Mizoguchi, R., *Tutorial on Ontological Engineering. Part 3-Advanced Course on Ontological Engineering*, New Generation Computing 22, pp. 196-220, Springer Verlag, 2004.

122. Bachimont, B., *Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en inginiérie des connaissances*, In Inginiérie des connaissances. Évolution Récentes et nouveaux défis, Charlet J., Zacklad M., Kassel G. and Bourgault D., Paris, pp. 305-323, 2000.

123. Pinto, H.S., Martins, J.P., *Ontologies: How Can They Be Built?*, Knowledge and Information Systems, 6 (4), 441-464, 2004.

124. Gangemi, A., Borgo, S., Catenacci, C., Lehman, J., *Task taxonomies for knowledge content*, proiect "Metokis" – Methodology and Tools Infrastructure for the Creation of Knowledge Units, on-line http://metokis.salzburgresearch.at/results/index.html.

125. Segal, C., Dumitriu, L., **Ştefănescu, D.**, *Test properties in a possibilistic relational diagnosis model*, The 11-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS2001, pp. 184-189, 2001.

126. Clancey, W.J., *Model construction operators*, Artificial Intelligence Magazine 53, pp. 1-115, 1992.

127. Steels, L., *Components of expertise*, AI Magazine, 11(2): pp. 28-49, 1990.

128. McDermott, J., *Preliminary steps towards a taxonomy of problem-solving methods*, In S. Marcus, ed., *Automating Knowledge Acquisition for Knowledge-Based Systems*, Boston: Kluwer Academic Publishers, 1988.

129. Zarei S., Malayeri A.D., *Data Management System and Knowledge Analyzing via KADS Approach*, In Middle-East Journal of Scientific Research, vol. 11(5), pp.595-601, ISSN 1990-9233, 2012.

130. Domingue, J., Fensel, D., *PSMs in a Global Networked Age*, AIEDAM Special Issue Problem Solving Methods: Past, Present and Future, 23(3), Cambridge University Press, 2009.

131. Trăuşan-Matu, Şt., *Studiu asupra instrumentelor hermenofone şi a ontologiilor existente pe Web*, Academia Română, Raport de cercetare RR-51, 2000.

132. Buraga, S.C., *Semantic Web-based Knowledge Management in Distributed Systems*, Post-proceedings of CANS – Complexity and Intelligence of the Artificial and Natural Complex Systems, In B.Iantovics et al. (eds.), IEEE Computer Society Press, 2009.

133. Trăuşan-Matu, Șt., *Ontology-Based Interoperability in Knowledge-Based Communication Systems*, In Ontologies in Urban Development Projects, Part.2, Advanced Information and Knowledge Processing (Falquet, G. et al.). Springer-Verlag London Limited, vol. 1, pp. 139-152, 2011.

134. Henze, N., *Personal Readers: Personalized Learning Object Readers for the Semantic Web,* In: C.-K. Looi, G. McCalla, B. Bredeweg and J. Breuker (eds.) Proceedings of 12th International Conference on Artificial Intelligence in Education, AIED'2005, Amsterdam, July 18-22, 2005, IOS Press, pp. 274-281, http://www.kbs.unihannoverde/Arbeiten/Publikationen/2005/aied05.pdf, 2005.

135. Trăuşan-Matu, Şt., *Intelligent personalizing web pages and understanding facilities*, Proceedings of WITREC-2000, Montpelier, France, URL:http://www.lirmm.fr/WITREC/wrk-abstracts/trausan.pdf, 2000.

136. Murray T., *Principles for Pedagogy-oriented Knowledge Based Tutor Authoring Systems: Lessons Learned and a Design Meta-Model*, Chapter 15 in Murray, Ainswoth&Blessing (eds.), Authoring Tools for Advanced Technology Learning Environments, Kluwer Academic/Springer Pub.: Netherlands, 2003.

137. Murray T., *Authoring for Advanced E-Learning Systems Moves Forward: Issues and Progress*, Technology Instruction Cognition and Learning (TICL) Vol. 2 #3, pp. 171-183, 2005.

138. Mizoguchi, R., Bourdeau, J., *Using Ontological Engineering to Overcome AI-ED Problems*, Internat. Journal of AI in Education, Vol.11, No2, pp107-121, 2000.

139. Bourdeau, J., Mizoguchi, R., Hayashi, Y., Psyche, V., and Nkambou, R.: *When the Domain of the Ontology is Education*, Proc. of the 4th Conf. on Intelligent, Interactive Learning Objects Repository Networks (I2LOR'07), CD-ROM, Nov. 4-7, 2007

140. Mizoguchi, R. (translated by Bourdeau, J.), *Le role de l'ingenierie ontologique dans le domaine des EIAH*, Sciences et Technologies de l´Information et de la Communication pour l´Education et la Formation, Vol.11, 2004.

141. Kasai, T., Yamaguchi, H., Mizoguchi, R., *An Ontological Approach for Supporting the Instructional Design Process of Information Education*, Proc. of the Artificial Intelligence in Education, pp.437-439, Sydney, Australia, 2003.

142. **K**asai, T., Yamaguchi, H., Nagano, K., Mizoguchi, R., *Building an ontology of IT education goals*, International Journal Cont. Engineering Education and Lifelong Learning, Vol. 16, Nos. 1/2, pp.1-17, 2006.

143. Aroyo, L., Mizoguchi, R., *Process-aware Authoring of Web-based Educational Systems*, The 15th Conference on Advanced Information Systems Engineering (CAiSE '03), Austria, Workshops Proc., 2003.

144. Aroyo, L., Dicheva, D., Cristea, A.I., *Ontological Support for Web Courseware Authoring*, The 6th International Conference, Intelligent Tutoring Systems, Biarritz, France and San Sebastian, Spain, Proc., pp. 270-280, 2003.

145. Kozachi, K., Kitamura, Y., Mizoguchi, R., *Developing Ontology-based Applications using Hozo*, International Conference on Computational Intelligence, Calgary, Canada, IASTED/ACTA Press, pp. 273-277, 2005.

146. Bourdeau, J., Mizoguchi, R., Psyché, V., Nkambou, R., *Selecting Theories in an Ontology-Based ITS Authoring Environment*, 7th International Conference Intelligent Tutoring Systems, Proc.: Lect. Notes in Comp. Science 3220, pp. 150-161, 2004.

147. Psyché, V., Bourdeau, J., Mizoguchi, R., *Ontology Development at the Conceptual Level for Theory-Aware ITS Authoring Systems*, in Hoppe, U., Verdejo, F. & Kay, J., AI in Education, Shaping the future of Learning through Intelligent Technologies, pp. 491-493, 2003.

148. Psyché, V., Bourdeau, J., Nkambou, R., Mizoguchi, R., *Making Learning Design Standards Work with an Ontology of Educational Theories*, Proc. of the 12th Artificial Intelligence in Education, pp. 539-546, Amsterdam, The Netherlands, 2005.

149. Jin, L., Hayashi, Y., Ikeda, M., Mizoguchi, R., Ohta, M., Takaoka, Y., *Design and realization of intelligent training system SmartTrainer*, Electronics and Communications in Japan (Part II: Electronics), Vol. 86, Issue 1, pp. 73–83, 2003.

150. Inaba, A., Mizoguchi, R., *Learners' Roles and Predictable Educational Benefits in Collaborative Learning - An Ontological Approach to Support Design and Analysis of CSCL*, Proc. of 7th Internat. Conf. on Intelligent Tutoring Systems (ITS2004), Springer-Verlag, pp.285-294, 2004.

151. Hayashi, Y., Bourdeau, J., Mizoguchi, R., *Toward Establishing an Ontological Structure for the Accumulation of Learning/Instructional Design Knowledge*, Proc. of 6th Internat. Worksh.on Ontologies&Semantic Web for E-Learning, pp.1-10, 2008.

152. Isotani, S., Mizoguchi, R., *CHOCOLATO: A Concrete and Hepful Ontology-Aware Collaborative Learning Authoring Tool*, Demonstrations Program of 9th International Conference of Intelligent Tutoring Systems (ITS'08), pp.21-24, Canada, 2008.

153. Nkambou R., Frasson C., Gauthier G., *CREAM-Tools: An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems*, In Authoring Tools forAdvanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software,Murray T., Blessing S., Ainsworth, S., pp. 93-138, Kluwer Publisher, 2003.

154. Apted, T., Kaz, J., Lum, A., Uther, J., *Visualisation of learning ontologies*, In Artificial Intelligence in Education: shaping the future of learning through intelligent technologies (papers selected for presentation at the 11th Internat. Conf. on Artificial Intelligence in Education), H.U. Hoppe&al. (Eds.), IOS Press, pp. 359-261, Australia, 2003.

155. Paquette, G., Tchounikine, P., *Contribution à l'ingénierie des systèmes conseillers: une approche méthodologique fondée sur l'analyse du modèle de la tâche*, In: Revue Sciences et Techniques Educatives 9(3-4), p. 409-435, 2002.

156. Paquette, G., Bourdeau, J., Psyché, V., *Vers une ontologie et une base de connaissances en téléapprentissage*, 3e colloque CIRTA, ACFAS, 2003, Québec, 2003.

157. Ranwez S., Crampes M., *Instanciation d'ontologies pondérées et calcul de rôles pédagogiques-Principe, mise en oeuvre*, *STE/STICEF*, vol. 9/2002, pp.341-370,2002.

158. Bittencourt, I.I., Isotani, S., Costa, E., Mizoguchi, R., *Research Directions on Semantic Web and Education*, SCIENTIA, Interdisciplinary Studies in Computer Science 19(1), pp. 59-66, URL: www.ei.sanken.osaka-u.ac.jp/pub/isotani/semanticweb.pdf.

159. Genesereth, M.R., Nillson, R., *Foundation of Artificial Intelligence*, 1987.

160. Guarino, N., Welty, C., *Identity and subsumption*, In R. Green, C.A. Bean, S. Hyon Myaend (eds.), The Semantics of Relationships: An Interdisciplinary Perspective, Kluwer, pp. 111-126, 2002.

161. Gómez-Pérez, A., Fernandez-Lopez, M., Corcho, O., *Ontological Engineering*, Springer-Verlag, Advanced Information and Knowledge Processing, 2003.

162. Pecheanu, E., **Ştefănescu, D.**, Istrate, A., *Building and Using Ontologies in Intelligent Instructional Systems*, The Romanian Symposium on Computer Science, ROSYCS 2004, Iasi, in Scientific Annals of the "A.I. Cuza" University of Iasi, Tome XV (**Revista Tip B, cod CNCSIS 115**), pp. 178–190, 2004.

163. van Aart, C., Wielinga, B., Schreiber, G., *Organizational Building Blocks for Design of Distributed Intelligent Systems*, International Journal Human-Computer Studies, Vol. 61(5), pp. 567-599, 2004.

164. Kasai, T., Yamaguchi, H., Nagano, K., Mizoguchi, R., *Development of a System that Provides Teachers with Useful Resources from Various Viewpoints Based on Ontology*, Proc. of the

sixteenth World Conference on Educational Mutimedia, Hypermedia & Telecommunications, pp. 3349-3356, Lugano, Switzerland, 2004.

165. Kasai, T., Yamaguchi, H., Nagano, K., Mizoguchi, R., *Goal Transition Model and Its Application for Supporting Teachers based on Ontologies*, Proc. of the 12th Artificial Intelligence in Education, pp. 330-337, Amsterdam, The Netherlands, 2005.

166. Minsky, M., *A framework for representing knowledge*, The Psychology of Computer Vision, P. H. Winston, Ed. New York, McGraw-Hill, 1975.

167. Aubert, J.P., Baget, J.F., Chein, M., *Simple Conceptual Graphs and Simple Concept Graphs,* In H. Scharfe et al, editor, Proc. ICCS'06, volume 4068 of LNAI, pp.87–101, Springer, 2006.

168. Berners-Lee, *Berners-Lee and the Semantic Web Vision*, URL:http://www.xml.com/pub/a/2000/12/xml2000/timbl.html, 2000, (u.v. dec. 2010).

169. Buraga, S. C.,**Ştefănescu, D.**, *Ontology Editing Support – A Study of Existing Tools*, Proc. of the 13-th International Symposium on Modeling, Simulation and Systems' Identification, SIMSIS Galati, Editors Luminiţa DUMITRIU, Mihai VLASE, Galati University Press, ISSN 1843-5130, 2007.

170. Gómez-Pérez, A., Fernandez-Lopez, M., Corcho, O., *OntoWeb – Technical Roadmap D.1.1.2*, IST-2001-29243, URL: http://www.aifb.uni-karlsruhe.de/Publikationen/ showPublikation_english?publ_id=713, 2002, (u.v. aug. 2010).

171. IEEE Guide for Software Quality Assurance Planning, Std. 730.1-1995, IEEE Computer Society, New York (USA), 1995

172. Noy, N.F., McGuinness, D.L., *Ontology Development 101: A Guide to CreatingYour First Ontology*, Stanford Knowledge Syst. Lab., Techn. Rep. KSL-01-05, 2001, http://wwwksl.stanford.edu/people.dlm/papers/ontology-tutorial-noy-mcguinness.pdf.

173. Garshol, L. M., *Tolog – a topic maps query language*, Paper presented at the Charting the Topic Map Research and Applications Landscape, TMRA'05 – International Workshop on Topic Map Research and Applications, Germany, 2005.

174. Chen, P.P.S, *The Entity-Relationship Model – Toward a Unified View od Data*, ACM Trannsaction on Database Systems, Vol. 1, No. 1, pp 9-36, 1976

175. Chein, M., Mugnier, M.L., *Graph-based Knowledge Representation: Computational Foundations of Conceptual* Graphs Book, ISBN: 978-1-84800-285-2, DOI 10.1007/978-1-84800-286-9, Springer-Verlag, 2009.

176. Barnes, H., *Structuring knowledge for beginning teachers*, In M.Reynolds (ed.), Knowledge Base for the Beginning Teacher, Oxfors, Pergamon Press, 1989.

177. Reigeluth (ed.) *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory*, Vol. 2 (Instructional Design Theories & Models), Lawrence Erlbaum Associates, ISBN 0805828591, 1999.

178. Pecheanu, E., **Ştefănescu, D.**, Dumitriu, L., Istrate, A., *Achitecture of a Computer Based Instructional System*, The Annals of "Dunarea de Jos" of Galati, Fascicle III (**Revista tip B, cod CNCSIS 482**), pp. 60-64, ISSN 1221-454 X, 2000.

179. Pecheanu, E., **Ştefănescu, D.**, Istrate, A., *An Extended Structural Model for Intelligent Instructional Systems*, The Romanian Symposium on Computer Science, ROSYCS 2004, Iasi, in Scientific Annals of the "A.I. Cuza" University of Iasi, Tome XV (**Revista Tip B, cod CNCSIS 115**), pp. 167-177, 2004.

180. Novac-Ududec, C.,**Ştefănescu, D.**, *Patterns Instruction in Software Engineering*, Proc. of, International Conference of"Interactive Computer Aided Learning ICL2007: Eportofolio and

Quality in e-learning, http://telearn.noe-kaleidoscope.org/open-archive/search?resource=1658_v1&back=%2Fopen-archive%2Fsearch%3Fsearch%3Dnovac%26type%255B%255D%3Dpublication%26type%255B%255D%3Dvideo%26type%255B%255D%3Dtool%26orderBy%3Dtitle%26search_button%3DSearch, Vilach, Austria,2007.

181. VanMarcke, K., *GTE: An epistemological approach to instructional modelling*, Instructional Science 26: 147-191, Kluwer Academic Publisher, Netherlands, 1998.

182. Vassileva, J., *DCG+GTE: Dynamic Courseware Generation with teaching expertise*, Instructional Science 26: 317-331, Kluwer Academic Publisher, Netherlands, 1998.

183. **Ştefănescu, D.,** Pecheanu, E., Dumitriu, L., *Knowledge representation models based on conceptual graphs in intelligent educational systems*, 3nd balkan region conference on engineering education, Advancing Engineering Education, Conference Proc., Sibiu, Editors: Constantin Oprean, Claudiu Vasile Kifor, Nicolaie Georgescu, pg 67-70, ISBN 973-739-147-0, 12-15 sept, **Proceeding ISI Quoted (Web of Science),** 2005.

184. **Ştefănescu, D,** Cocu, A., G*rafuri conceptuale aplicate in dezvoltarea unui sistem autor*, Conferința Națională de Învățământ Virtual, CNIV-2005, Virtual Learning**–**Software & management educational, București, pp.115-122, ISBN 973-737-097-X, 2005.

185. **Ştefănescu, D,** Pecheanu, E., Cocu, A., *Pedagogical knowledge model based on conceptual graphs and ontology*, **D. Ștefănescu**, The Annals of "Dunarea de Jos" University of Galati, Fasc.III (**Revista tip B+, cod CNCSIS 482**), pp.11-17, ISSN 1221-454X, 2005.

186. **Ştefănescu, D.**, Pecheanu, E., Bumbaru, S., Buraga, S.C., *Authoring instructional strategies in intelligent educational systems: a model based on conceptual graphs*, Inter-Ing 2007, Scientific Conference with international participation: Interdisciplinarity in engineering, Ed. Univ. „Petru Maior" Targu-Mures, pg. VI-3-1 – VI-3-6, ISSN 1843-780X, 2007.

187. Pecheanu, E., **Ştefănescu, D.**, Istrate, A., Dăscălescu, D., *Conceptually Modelling the Domain Knowledge of an Instructional Environment*, Proc. of the International Conference on Computer aided learning in Engineering Education – CALIE'04, febr. 2004, Grenoble, Franta, pp. 215 - 221, 2004.

188. Pecheanu, E., **Ştefănescu, D.**, Dumitriu, L., Segal, C., *Knowledge Modelling in Computer-Assisted Instruction*, OL-KWM - International Symposium on Organizational Learning and Knowledge Work Management, Bucuresti, 2005**.**

189. Pecheanu, E., Dumitriu, L., **Ştefănescu, D.**, Segal, C., *A Framework for Conceptually Modelling the Domain Knowledge of an Instructional System*, Book Series Lecture Notes in Comp. Sc., Publisher Springer Berlin / Heidelberg, Volume 3992/2006, ISBN 3-540-34381-4, pp. 199-206, **Journal ISI Quoted (Web of Science),** 2006.

190. Pecheanu, E. and **Ştefănescu, D.,** *On Modeling Unstructured Data Collections in Computer Assisted Instruction*, 5th EuropeanConference on Intelligent Systems and Technologies (ECIT'2008) 2008.

191. Pecheanu, E., **Ştefănescu, D.**, Istrate, A, *On Modeling Instructional Content in Computer Assisted Learning*, The Annals of "Dunărea de Jos" University of Galaţi, Fascicle III, Electrotechnics, Electronics, Automatic Control, Informatics, Vol. 32, Nr. 2, ISSN 1221-454X, Paginile: 21-24, http://www.ann.ugal.ro/eeai/archives/2009/Lucrare-4-EPecheanu-p21-24.pdf, (**Indexare BDI: DOAJ - Directory of Open Access Journals, 2010, Lund University Libraries**),http://www.doaj.org/doaj?func=openurl&issn=1221454X&genre=journal, 2009.

192. Pecheanu, E., **Ştefănescu, D.,** Popescu, F., *A Solution for Modeling the Instructional Content in Computer Assisted Learning*, Proc. of the 6th International Conference on New Horizons in

Industry, Business and Education – NHIBE 2009, Santorini, Greece, I, ISBN 978-960-88785-8-7, pp 171-175, 2009.

193. Sowa, J.F., *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole, chapt. 3 & 4, 2000.

194. Sowa, J.W., *Conceptual Graphs – Bridging the Gap Between Language and Logic*, VivoMind LLC, URL:http://www.jfsowa.com/cg/cgonto.htm.

195. Sowa, J.F., *Chapter 5 – Conceptual Graphs*, In Handbook of Knowledge Representation, , ed. F. van Harmelen, V. Lifschitz and B. Porter, Elsevier, pp.213-237, 2008.

196. Genest, D., Chein, M., *A content-search information retrieval process based on conceptual graphs*, Knowledge on Information Systems, Vol. 8(3), 2005.

197. Kabbaj, A., *From PROLOG++ to PROLOG+CG: A CG Object-Oriented Logic Programming Language*, Proc. of the International Conference on Conceptual Structures, pp. 540-544, Springer-Verlag (LNAI 1867), 2000.

198. [198]*** Amine Paltform, URL:http://amine-platform.sourceforge.net/.

199. Mugnier, M.L., Chein, M., *Représénter des connaissances et raisonner avec des graphes*, R.I.A., vol. 10, nr. 1, pp 7-56, 1996.

200. Mugnier, M.L., Chein, M., *Knowledge Representation and Reasonings Based on Graph Homomorphism*, Proc. of International Conference on Conceptual Structures (ICCS-2000), Springer, pp. 172-192, 2000.

201. Mugnier, M.L., *Knowledge Representation and Reasonings Based on Graph Homomorphism*, Research Report LIRMM 00-098, 2000.

202. CoGITaNT - *Conceptual Graphs Integrated Tools allowing Nested Typed graphs***,** URL: http://cogitant.sourceforge.net/(u.v. aug. 2010).

203. Chein, M., Mugnier, M.L., *CGs Applications: Where Are We 7 Years after the Fist ICCS?*, The 8th International Conference on Conceptual Structures, ICCS 2000, Germany, Proc.Lecture Notes in Computer Science, Vol. 1867, pp 127-139, 2000.

204. Fensel, D., Horrocks, I., van Harmelen, F., McGuinness, D., Patel-Schneider, P.F., OIL: Ontology Infrastructure to Enable the Semantic Web, IEEE Intelligent Systems, 2001.

205. Corcho, C., Gómez-Pérez, A., *A Roadmap to Ontology Specification Languages*, In Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management, pp. 80-96, 2000.

206. Berners-Lee, *Weaving the Web. The original design and ultimate destiny of the World Wide Web, by its inventor*, Publisher: Harper Paperbacks, ISBN-10: 006251587X, USA, 2007

207. Gómez-Pérez, A., Corcho, O., *Ontology Languages for the Semantic Web*, IEEE Intelligent Systems, pp. 54-60, 2002.

208. Corcho, O., Fernandez-Lopes, M.,Gómez-Pérez, A., *Methodologies, tools and languages for building ontologies. Where is their meeting point?*, In Data&Knowledge Engineering 46, pp. 41-64, Elsevier, 2003.

209. Hayes, P., Menzel, C., *IKL Specification Document*, http://www.ihmc.us:16080/users/phayes/IKL/SPEC/SPEC.html, 2009, (u.v. apr. 2009).

210. *Common Logic – Motivations and Some Gentle Theory*, SemTech 2008 Workshop, http://cl.tamu.edu/docs/cl/SemTech2008/Menzel-SemTech2008.pdf

211. *ISO/IEC 24707:2007 – Information Technology – Commn Logic (CL) – A framework for a family of logic-based languages*, http://www.iso.org/iso/catalogue_detail.htm?csnumber=39175.

212. [212]***, RDF, http://www.w3.org/TR/rdf-schema/.

213. Baget, J.F., *RDF Entailment as a Graph Homomorphism*, The 4th International SemanticWeb Conference (ISWC), Ireland, Proc.: Lecture Notes in Computer Science 3729, pp. 82-96, ISBN 9783540297543, 2005.

214. Corby, O., Dieng, R., Hébert, C., *A Conceptual Graph Model for W3C Resource Description Framework*, In 8th International Conference on Conceptual Structures (ICCS'00), pp. 468-482, Germany, 2000.

215. Wielemaker, J., Schreiber, G., Wielinga, B., *Using triples for implementation: the {Triple20} ontology-manipulation tool*, ISWC 2005, Galway, Ireland, Springer-Verlag, LNCS 3729, pp. 773-785, 2005.

216. *DAML+OIL Technical Detail,* Talk given at W3C Web Ontology Working Group meeting, Bell Laboratories, Murray Hill NJ, 2002, http://www.cs.man.ac.uk/~horrocks/Slides/wowg-talk.pdf

217. Horrocks, I., *OWL: A Description Logic Based Ontology Language*., Talk at CISA, Edinburgh, 2006, http://www.cs.man.ac.uk/~horrocks/Slides/cisa06.ppt.

218. Sleeman, D., Reul, Q., *CleanONTO: Evaluating Taxonomic Relationships in Ontologies*, EON2006 (Evaluation of Ontologies for the Web), 4th International EON Workshop, 2006, (u.v. aug. 2010).

219. Sowa J.F., *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA., 1984.

220. Chein, M., Mugnier, M.L., *Positive Nested Conceptual Graphs*, Proc. Of the 5thInternational Conference on Conceptual Structures", Springer Verlag, pp. 95-109, 1997.

221. Chein, M., Mugnier, M.L., *Nested Graphs: A Graph-based Knowledge Representation Model with FOL Semantics*, KR, , pp 524-535, 1998.

222. Chein, M., Mugnier, M.L., *Concept Types and Coreference in Simple Conceptual Graphs*, Proc. of the 9th International Conference (KR2004): Principles of Knowledge Representation and Reasoning, pp. 303-318, Canada, 2004.

223. Baget, J.F., Genest, D., Mugnier, M.L., *A Pure Graph-Based Solution to the SCG-1 Initiative*, The 8th International Conference on Conceptual Structures, Proc. of ICCS '99 (vol. 1640), Virginia, USA, pp. 355-376, 1999

224. Chein, M., Mugnier, M.L., *Conceptual Graphs are also Graphs*, Research Report LIRMM, 950003, 1995.

225. Baget, J.F., *Extending the CG Model by Simulation*, The 8th International Conference on Conceptual Structures, ICCS 2000, Germany, Proc.Lecture Notes in Computer Science Vol. 1867, pp. 277-291, 2000.

226. Baget, J.F., Mugnier, M.L., *The SG Family: Extensions of Simple Conceptual Graphs*, The 17th International Joint Conference on AI, Proc. of IJCAI'2001 (vol. 1), Seatle, USA, pp. 205-212, Morgan Kaufmann, 2001.

227. Baget, J.F., Mugnier, M.L., *Extensions of Simple Conceptual Graphs: the Complexity of Rules and Constraints*, Journal of AI Research 16, pp. 425-465, 2002.

228. Baget, J.F., *Improving the Forward Chaining Algorithm for Conceptual Graphs Rules*, Proc. of the 9th International Conference (KR2004): Principles of Knowledge Representation and Reasoning, pp. 407-414, Canada, 2004.

229. Mugnier, M.L., Leclère, M., *On Querying Simple Conceptual Graphs with Negation*, Rapport de recherche, LIRMM 05-051, 2005.

I. ANEXA I – STUDIU DETALIAT AL ELEMENTELOR INGINERIEI ONTOLOGICE

AI.1. OBIECTIVELE INGINERIEI ONTOLOGICE

Ingineria ontologică este definită ca “mulţimea activităţilor implicate în procesul de dezvoltare a ontologiilor şi în ciclul de viaţă al acestora, metodologii, instrumente şi limbaje pentru construirea ontologiilor”.

Ontologiile se doresc reprezentări ale cunoştinţelor construite la nivel conceptual, care permit utilizarea cunoştinţelor pentru raţionament. Ontologia trebuie să fie o construcţie inginerească, formată din vocabularul utilizat pentru descrierea realităţii, împreună cu axiomele referitoare la semnificaţia intenţională a vocabularului. Ca urmare, trebuie să integreze cunoştinţele terminologice ale unui domeniu şi semantica acestuia, păstrând totodată independenţa de utilizarea operaţională a sistemului bazat pe cunoştinţe.

Practic, ingineria ontologică oferă posibilităţile de a obţine:
- Vocabular comun şi o structurare superioară a definiţiilor conceptelor;
- Interoperabilitate semantică şi expresivitate;
- Coerenţa şi sistematizarea cunoştinţelor;
- Meta-modele pentru rezolvarea problemelor într-o varietate de contexte.

Proiectele actuale de cercetare în ingineria ontologiilor vizează:
- Obţinerea unor metodologii de proiectare a ontologiilor şi limbajele lor de reprezentare;
- Stabilirea principiilor ingineriei ontologice;
- Dezvoltarea, partajarea şi reutilizarea cunoaşterii din bazele de cunoştinţe existente;
- Fructificarea avantajelor oferite de web-ul semantic prin modalităţi superioare de achiziţie, gestiune, partajare şi prelucrare a cunoştinţelor pe web.

În concluzie, ingineria ontologică vizează *dezvoltarea de metodologii, modele, instrumente şi limbaje pentru ontologii*.

AI.2. MODELE ŞI LIMBAJE DE REPREZENTARE A CUNOŞTINŢELOR UTILIZATE ÎN INGINERIA ONTOLGICĂ

Modelele de reprezentare a cunoştinţelor utilizate în ingineria ontologică pot fi grupate după paradigmele conceptuale reliefate:

- Modele bazate pe **cadre**;
- Modele bazate pe **logici de descriere**;
- Modele bazate pe **grafuri conceptuale**.

Aceste formalisme permit specificarea cunoştinţelor despre obiectele din domeniu, relaţiile dintre concepte şi semantica primitivelor de modelare. Pentru fiecare model există unul sau mai multe **limbaje** care implementează *parţial* sau *total* un anumit model. În familia limbajelor de reprezentare a cunoştinţelor, limbajelor ”clasice” li se alătură limbajele adaptate web, care utilizează sintaxa xml. Unele limbaje sunt *operaţionale* (oferă mecanisme de raţionament), altele permit doar *specificarea declarativă* a cunoştinţelor.

AI.2.1 Modele bazate pe cadre

Modelul cadrelor (frame-uri sau scheme) - introdus de către Minsky - a fost propus iniţial de către Gruber pentru reprezentarea ontologiilor. Modelul permite reprezentarea cunoştinţelor despre o situaţie (obiect) printr-o schemă. *Principiul de bază* al acestei familii de modele este descompunerea cunoaşterii în clase (cadre) care reprezintă concepte ale domeniului. Fiecare cadru are ataşat un anumit număr de atribute (slot-uri) pentru proprietăţile obiectului reprezentat, iar fiecare atribut poate primi o valoare dintr-o mulţime de sloturi (faţete). O *alternativă a modelului bazat pe cadre* este aceea în care atributele sunt privite ca relaţii binare între clase, argumentele relaţiei fiind numite domeniu, respectiv rang. Instanţele claselor corespund extensiunilor conceptelor.

La acest model pot fi adăugate funcţii care să reprezinte tipuri particulare de relaţii care leagă un ansamblu de clase cu o valoare calculată pe baza atributele claselor. Specificarea proprietăţilor conceptuale ale atributelor (sau ale relaţiilor) se realizează prin formulele logicii de ordinul întâi (FOL[56]).

Semantica de subsumare este pur extensională, considerându-se că un cadru, F_1, este mai specific decât un alt cadru, F_2, dacă orice instanţă a lui F_1 este instanţă a lui F_2.

Aşadar, *raţionamentul* poate fi unul de clasificare a cadrelor (dacă un cadru este mai specific sau mai general decât altul) sau de descoperire a mulţimii de valori ale proprietăţilor pentru un anumit obiect.

Limbajele pentru modelele bazate pe cadre sunt:

- **KIF** (**K**nowledge **I**nterchange **F**ormat) - limbaj neoperaţional care implementează modelul cadrelor în FOL, conceput cu scopul de a rezolva problema heterogenităţii limbajelor de reprezentare a cunoaşterii.
- **OKBC** (**O**pen **K**nowledge **B**ase **C**onnectivity, numit anterior “Generic Frame Protocol”) specifică un protocol (nu limbaj) şi API[57] pentru interogări şi interfaţă, furnizează accesul la ontologiile implementate într-un limbaj al cadrelor.
- **F-Logic**, **Cycl**, **Ontolingua**, **OCML** - limbaje operaţionale bazate pe cadre şi FOL:

[56] FOL – logica de ordinul întâi, logica predicatelor (engl., ”First Order Logic”)

[57] API – Interfața aplicației (engl., ”Application Programming Interface”)

- Ontolingua permite construirea ontologiilor fie utilizând doar vocabularul din modelul cadrelor (în care axiomele nu pot fi exprimate), fie utilizând expresii KIF, fie folosind ambele limbaje.
- OCML[58] (**O**perational **C**onceptual **M**odeling **L**anguage) – limbaj operaţional, permite exprimarea relaţiilor, funcţiilor şi regulilor (raționament cu înlănţuire înainte şi înapoi), a claselor şi a instanţelor, dar şi ataşarea unor proceduri.
- F-Logic (**F**rame **Logic**), integrează un limbaj bazat pe cadre cu FOL.

AI.2.2 Modele bazate pe logici de descriere

Logicile de descriere[59] combină reprezentările intensionale şi extensionale ale cunoştinţelor.

Nivelul terminologic (T-box) conţine declaraţiile intensionale ale cunoştinţelor, descrise prin primitivele *concepte* şi *roluri* (relaţii binare între conceptele domeniului, similare cu atributele din modelul bazat pe cadre).

Nivelul aserţional (A-box) conţine declaraţiile extensionale ale instanţelor conceptelor definite la nivelul terminologic şi permite specificarea unor constrângeri asupra conceptelor sau a rolurilor.

Specificarea conceptelor şi a rolurilor se realizează fie prin constructori din teoria mulţimilor, fie prin constructori logici, interpretarea putând fi cea logică sau cea din teoria mulţimilor (teoria modelului). *Raţionamentul principal* este cel de subsumare. Au fost propuse numeroase modele (în funcţie de constructorii consideraţi) şi limbaje ale acestora.

Cele mai cunoscute **limbaje** pentru logicile de descriere sunt **LOOM**, **CLASSIC** și **KL-ONE**.

AI.2.3 Modele bazate pe grafuri conceptuale

Formalismul grafurilor conceptuale, introdus de către Sowa reprezintă o simbioză între aspectele cognitive şi cele computaţionale legate de reprezentarea cunoaşterii. Fiind derivate din reţelele semantice şi grafurile existenţiale introduse de Peirce, grafurile conceptuale (CG) furnizează nu numai un mod de *reprezentare grafică* a *cunoaşterii declarative* şi *procedurale* (cu ajutorul grafurilor conceptuale dinamice), ci şi un *model bazat pe teoriile matematice ale grafurilor şi ale logicii*.

Un graf conceptual (CG) este un graf bipartit, în care nodurile sunt conectate între ele prin arce sau prin muchii etichetate (în funcţie de abordare). Nodurile unui CG sunt de două tipuri:

- noduri care reprezintă tipuri de concepte (pe care le vom numi ***c-noduri***);
- noduri care reprezintă tipuri de relaţii (***r-noduri***) (figura AI - 1).

În capitolul al III-lea din [193], autorul subliniază: *"a conceptual graph has no meaning in isolation. Only through the semantic network are its concepts and relations linked to context, language, emotion and perception"*. Ideea evidenţiată este că CG au semnificaţie doar dacă sunt introduse într-o reţea semantică (altfel, sunt doar grafuri bipartite). De aceea, CG sunt

[58] OCML – limbaj considerat "Ontolingua operaţional"

[59] LD – logici de descriere (engl., "Description Logics")

"legate" de un aşa-numit **suport** (numită de către Sowa **canon** sau **bază canonică**), care reprezintă **ontologia** domeniului.

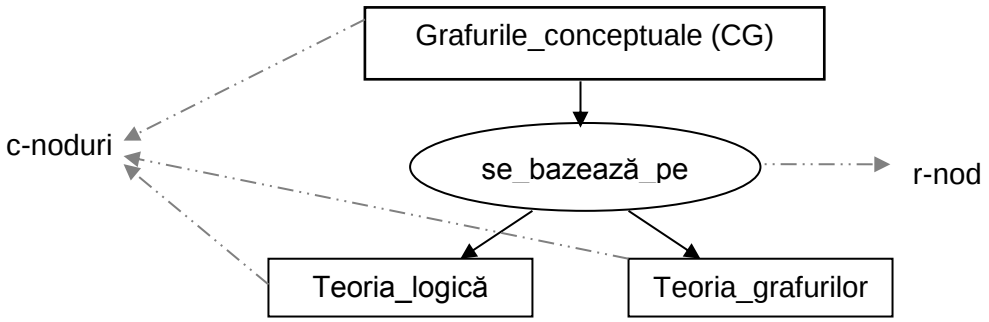


Figura AI - 1 Exemplu de CG

Există două forme principale de prezentare a CG [194] :

- Forma grafică (graf neorientat etichetat sau digraf) în care nodurile concepte sunt reprezentate prin dreptunghiuri, iar relaţiile care interconectează nodurile concepte sunt reprezentate prin ovale (figura AI - 2, respectiv figura AI - 3);
- Forma lineară (figura AI - 4, respectiv figura AI - 5)

Grafurile conceptuale sunt privite de Sowa [195] ca *"system of logic that can express the propositional content of sentences in natural language in as simple and direct manner as possible"* (*un sistem al logicii care poate exprima în cel mai simplu şi direct mod, conţinutul propoziţiilor din limbajul natural*).

Figura AI - 2 Grafuri conceptuale (neorientate, etichetate) echivalente, în notaţie grafică

Figura AI - 3 Grafuri conceptuale (digrafuri) echivalente, în notație grafică

Figura AI - 4 Grafuri conceptuale (neorientate, etichetate) echivalente, în notaţie lineară

Figura AI - 5 Grafuri conceptuale (digrafuri) echivalente, în notaţie lineară

Fiecare nod al unui CG este etichetat cu:

- Eticheta ($\mathbf{E_r}$) unui nod relaţie conceptuală[60] (r-nod) reprezintă numele tipului relaţiei;
- Eticheta[61] ($\mathbf{E_c}$) unui *nod tip conceptual* (c-nod) este formată din (cel puţin):
  - Tipul conceptului (obligatoriu), $\mathbf{t_c}$;
  - Referent (opţional), marcator, $\mathbf{ref_c}$ – care diferenţiază instanţele unui tip concept - care poate fi specificat prin:
    - *Referent generic* (poate lipsi sau este notat cu *) care nu identifică o anumită instanţă şi este echivalentul cuantificatorului exstenţial (∃); semnificaţia este că există cel puţin un individ (o instanţă) de tipul respectiv;
    - Un *referent individual* desemnat printr-un identificator sau variabilă (“conform” unui tip de concept), care identifică o anumită instanţă – entitate, obiect, proces, stare, eveniment, etc.– din universul de discurs;

Ca exemplu, figura AI - 6 prezintă sub formă grafică, respectiv lineară, fraza *“Profesorul Popa predă disciplina Programare”*.

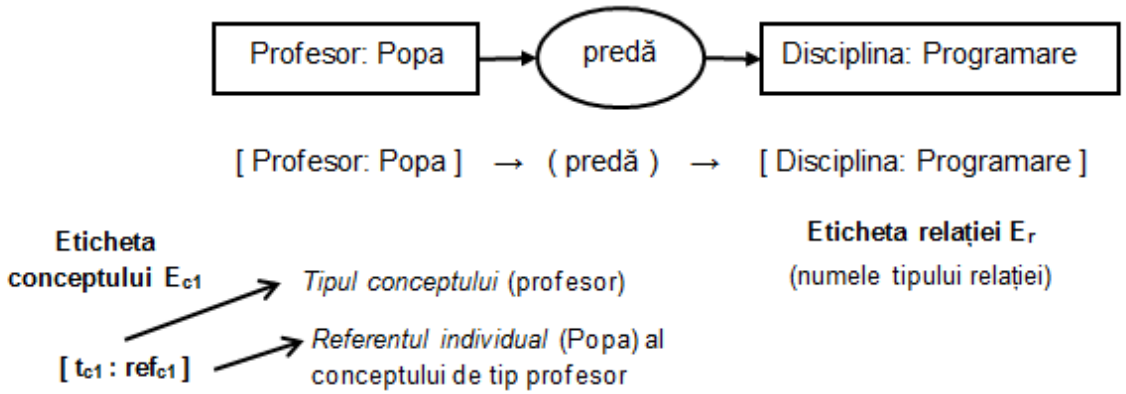


Figura AI - 6 Etichetarea nodurilor concepte şi relaţii

Fraza *“Există un profesor care predă disciplina Programare”* poate fi reprezentată *în formă lineară* (de cele mai multe ori, referentul generic, *, este omis, ca în prima formă) astfel:

```
[Profesor] –> (Predă) –> [Disciplina: Programare]
```

sau

```
[Profesor: *] –> (Predă) –> [Disciplina: Programare]
```

Observaţie:

Modul de etichetare a nodurilor care reprezintă tipuri conceptuale, sub forma (t_c, ref_c), este cel pentru *grafurile conceptuale simple*; în cazul *grafurilor conceptuale imbricate*, eticheta are forma (t_c, ref_c, g_c), unde g_c este un alt graf conceptual, numit în unele abordări *context*, în altele – *modul*.

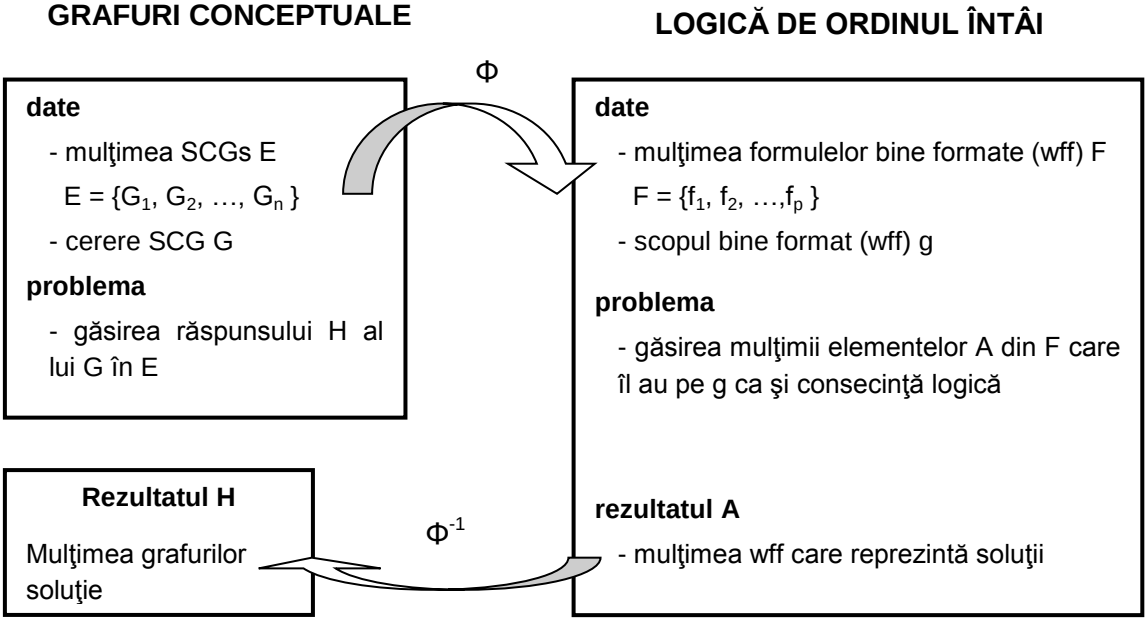
[60] Vom folosi termenul “prescurtat” *nod relaţie*, în locul celui de *nod relaţie conceptuală*.

[61] Modul prezentat de etichetare a unui c-nod, este cel din modelul de bază; în modelele extinse pot apărea şi alte tipuri de referenţi, chiar alte grafuri conceptuale (desemnate printr-un descriptor, ca în cazul grafurilor imbricate (numite şi contexte sau module).

Grafurile conceptuale pot constitui baza unui sistem operaţional de reprezentare a cunoştinţelor (figura AI - 9), putând fi folosite atât în prelucrarea limbajului natural [193], logică, căutarea informaţiilor [196] sisteme formale şi raţionamentele asociate.

Pornind de la **legătura dintre CG şi logică** (ilustrată şi în figura AI - 9, prin legăturile II şi 1), în ***utilizarea limbajului grafurilor conceptuale ca limbaj de reprezentare***, există *două direcţii importante de cercetare*:

- În prima direcţie de cercetare limbajul CG este privit ca o ***simplă notaţie grafică*** a unei teorii FOL (engl., "***F****irst* ***O****rder* ***L****ogic"*). Deoarece un CG este mai uşor de "citit" decât formula FOL corespunzătoare (*"un bon dessin vaut mieux qu'un long discours"*), CG sunt transformate în formule logice[62], iar raţionamentul se realizează prin algoritmii logici de rezoluţie (figura AI - 7) [193]. Această abordare a apărut ca urmare a cercetărilor legate de grafurile existenţiale ale lui Peirce (*grafurile exstenţiale* reprezintă formule conjunctive pozitive, închise existenţial; în aceste formule FOL pot apărea co-referinţe reprezentând variabile sau negaţii). Deducţiile logice asupra acestor obiecte constau în manipulări grafice (duplicare, deplasare, ştergere, etc.). După translatarea grafurilor conceptuale în formulele logice corespunzătoare (figura AI - 7), raţionamentul se realizează prin algoritmi logici de rezoluţie. Sistemul Prolog+CG [197] adoptă această abordare [198].



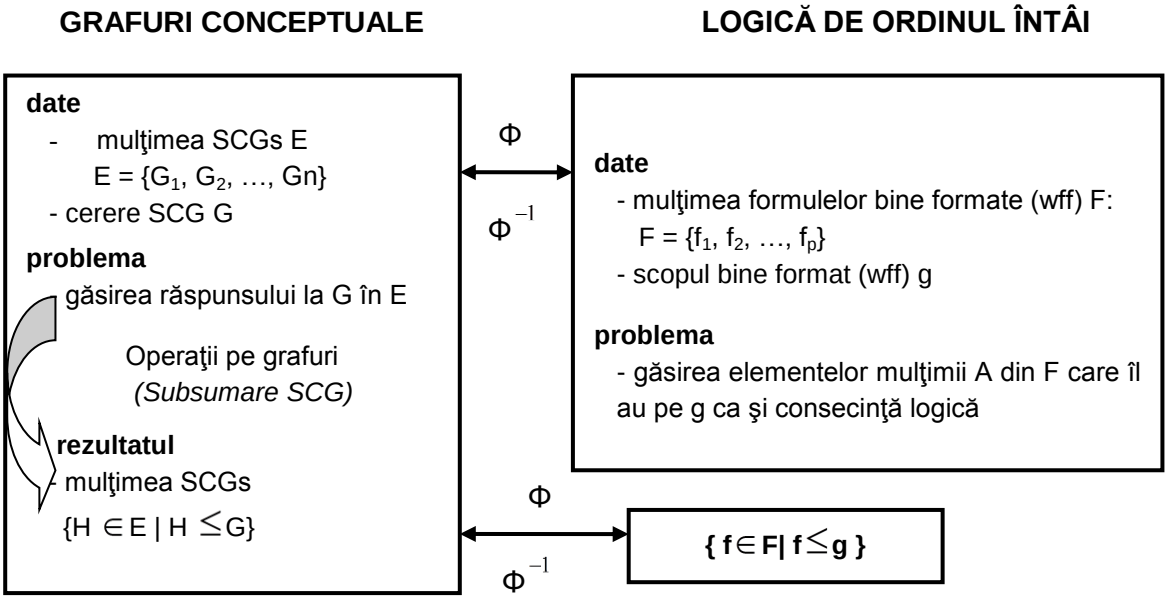
Fie E mulţimea faptelor și interogarea G (exprimate ca grafuri conceptuale simple). Operatorul (funcţia) Φ asociază grafurilor formule logice, calculul constă în demonstrarea logică a unei teoreme (demonstrator logic).

Figura AI - 7 Grafurile conceptuale ca reprezentare grafică a logicii [199]

- În ***a doua direcţie*** (abordarea din această lucrare) limbajul CG este privit ca sistem formal [200] în care primitivele sunt grafuri conceptuale, iar raţionamentul se bazează pe operaţii

[62] Oricărui CG i se poate asocia o **formulă logică**, utilizând funcţia ϕ propusă de Sowa sau operatorul Ψ propus de Chein.

interne (practic, un morfism al grafurilor) care păstrează corectitudinea şi completitudinea în raport cu FOL (figura AI - 7). Corespondenţa între inferenţa logică şi operaţiile de generalizare asupra grafurilor conceptuale a fost stabilită de către Chein şi Mugnier [175]. Fiecărui CG G i se asociază o formulă $\Phi(G)$ a logicii de ordinul întâi şi se aplică teorema (Sowa, 1984): Fiind date 2 grafuri G şi H (legate de acelaşi suport S), *dacă* $G \leq H$, *atunci*[63] $\Phi(S), \Phi(G) \vdash \Phi(H)$. În plus, în anumite situaţii[64], *dacă* $\Phi(S), \Phi(G) \vdash \Phi(H)$, *atunci* $G \leq H$. Relaţia $\leq$ între CG este numită relaţie de specializare [200] şi corespunde existenţei unui morfism – care satisface o anumită proprietate a etichetelor - între grafurile H şi G (mai precis, unei proiecţii de la H la G). În plus, morfismul poate fi definit constructiv cu ajutorul unor operaţii simple asupra grafurilor (identificarea a două noduri, suprimarea nodurilor identice, etc.). Funcţia Φ furnizează o ***interpretare semantică logică consistentă şi completă*** [201] a grafurilor conceptuale.



Fie E mulţimea faptelor și interogarea G (exprimate ca grafuri conceptuale simple). *Operatorul (funcţia) Φ asociază grafurilor formule logice, însă calculul se realizează printr-un demonstrator CG (algoritmi asupra grafurilor, mai precis prin operația de proiecție).*

Figura AI - 8 Grafurile conceptuale - sistem formal cu semantică logică [200]

Din acest punct de vedere, modelul grafurilor conceptuale prezintă avantajele:

- Un model declarativ de reprezentare a cunoştinţelor bazat pe grafurile etichetate;
- Calculul inferenţelor se realizează prin algoritmi ai grafurilor (fără a folosi un rezolvitor de teoreme, ca în logică) [201];
- Are o semantică logică consistentă şi completă.

Sistemul **CoGITaNT** (ales ca mediu de dezvoltare în această lucrare) adoptă **a doua abordare** [202].

[63] Φ(S) – mulţimea formulelor logice asociate suportului

[64] G şi H – grafuri bine formate, în formă normală (definite în Anexa III).

Pe lângă interpretarea logică pe care am evidenţiat-o, este posibil, deasemenea, să dăm grafurilor conceptuale o ***interpretare în sensul teoriei mulţimilor*** [200] (figura AI - 9, legăturile I, 3). Această interpretare furnizează un model cu o a doua semantică (similară celei din logicile de descriere), care corespunde cu semantica din teoria modelului (asociată modelului formal al teoriei logice exprimate în calculul FOL şi reprezentată în figura AI - 9 prin legătura 2).

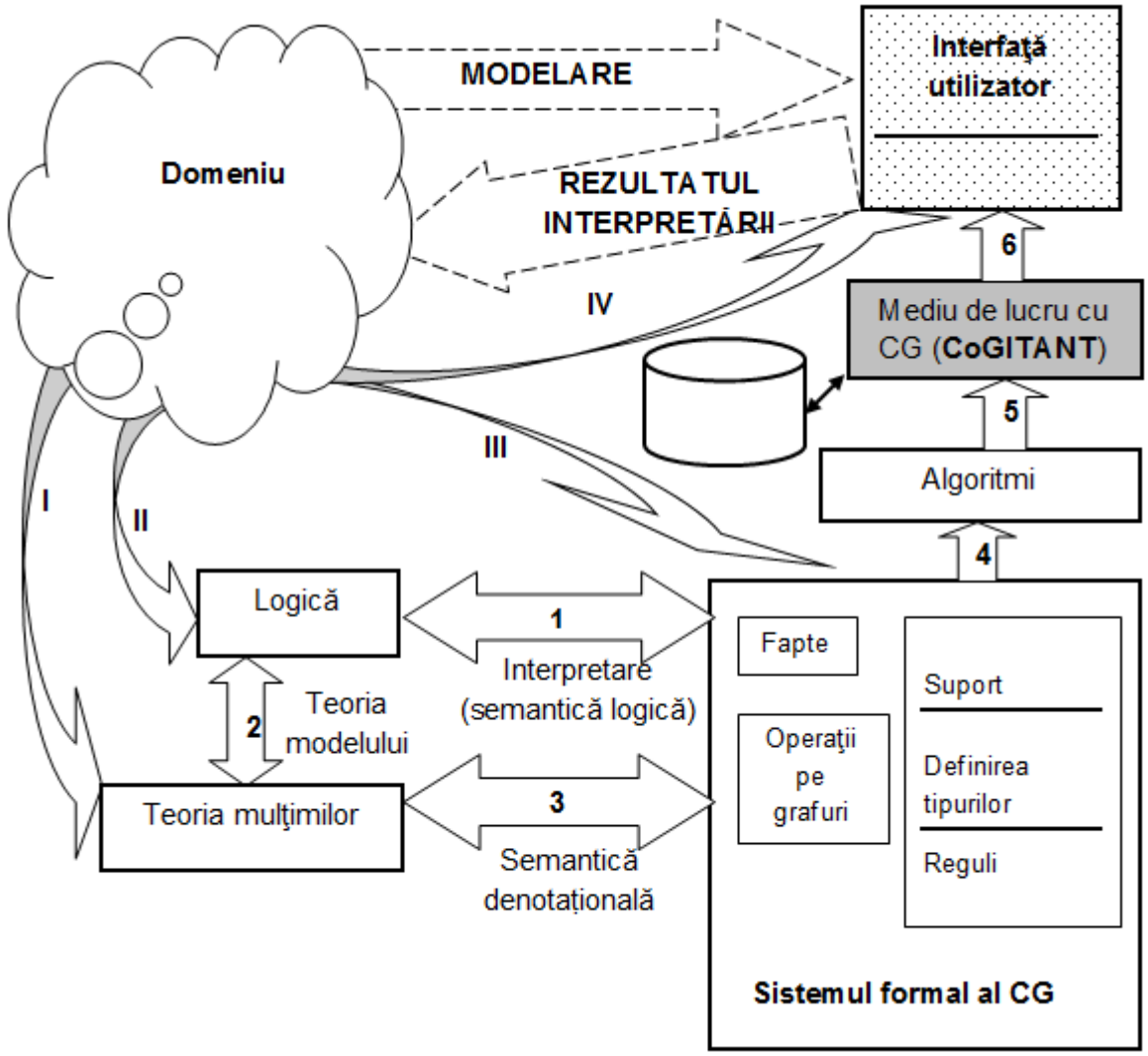


Figura AI - 9 Posibilităţi de utilizare a sistemului formal al grafurilor conceptuale

În cadrul suportului, tipurile de concepte sunt organizate într-o ierarhie de tip latice, pe baza unei relaţii de ordine parţială – relaţia semantică AKO – relaţie de specializare/generalizare (subsumare) (ca în figura AI - 10 şi). Şi tipurile de relaţii conceptuale pot fi organizate ierarhic pe baza unei relaţii de subsumare. Clasificarea conceptelor şi a relaţiilor este tratată ca o ***consecinţă logică***. Şi suportului i se poate da o interpretare logică sau una în sensul teoriei mulţimilor (aşa cum se ilustrează în capitolul 3 din lucrare).

O reprezentare intuitivă a unui graf conceptual legat de suport este redată în figura AI - 10.

Aşa cum deja am evidenţiat, raţionamentul în formalismul grafurilor conceptuale se realizează prin **proiecţie**, operaţie care corespunde deducţiei pentru formulele existenţiale conjunctive din FOL [175].

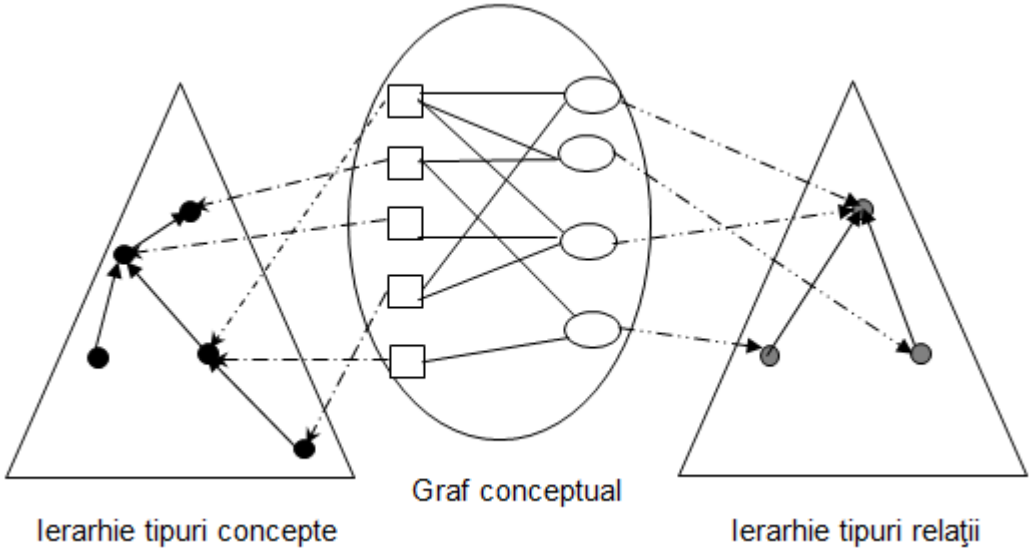


Figura AI - 10 Graful conceptual simplu: mulţime de pointeri semantici către ontologie

Ca exemplu, fie ierarhia de concepte din figura AI - 11 a (reprezentând suportul, ontologia domeniului). Relaţia de subsumare (AKO) între tipurile de concepte definite în ontologie (suportul S) - *"Orice logică ierarhică este o (cu sensul de subtip, AKO) logică ŞI orice logică este un formalism ŞI orice OntologieAI este un formalism"* (figura AI - 11- a este exprimată prin mulţimea de formule Φ_0 (notată şi Φ(S)). Fie graful H care exprimă faptul că *"Formalismul CG combină logica grafurilor existenţiale (PeirceEG) într-o logică a ierahiilor de clase, care ea însăşi combină o OntologieAI cu logica FOL"* (figura AI - 11- b căruia îi corespunde Φ_1(notată şi Φ(H)); fie graful G care exprimă faptul că *"Formalismul CG combină două (nu neapărat diferite) logici"* (figura AI - 11- c căruia îi corespunde. Între grafurile G şi H există relaţia G≥H. Atunci, formula Φ_2 poate fi obţinută din Φ(S) şi Φ(H), prin ***derivare***.

$\Phi_0 = \forall x(\text{Logica\_cu\_ierarhie\_de\_clase}(x) \rightarrow \text{Logica}(x)) \wedge$

$\wedge \forall x(\text{Logica}(x) \rightarrow \text{Formalism}(x)) \wedge$

$\wedge \forall x(\text{OntologieAI}(x) \rightarrow \text{Formalism}(x))$ (=Φ(S))

$\Phi_1 = \exists x(\text{Formalism}(CG) \wedge \text{Logica}(\text{PeirceEG}) \wedge$

$\wedge \text{Logica\_ierarhica}(x) \wedge \text{combina}(CG, \text{PeirceEG}, x)) \wedge$

$\wedge \exists y\ (\text{OntologieAI}(y) \wedge \text{Logica}(FOL) \wedge \text{combina}(x, y, FOL))$ (= Φ(H))

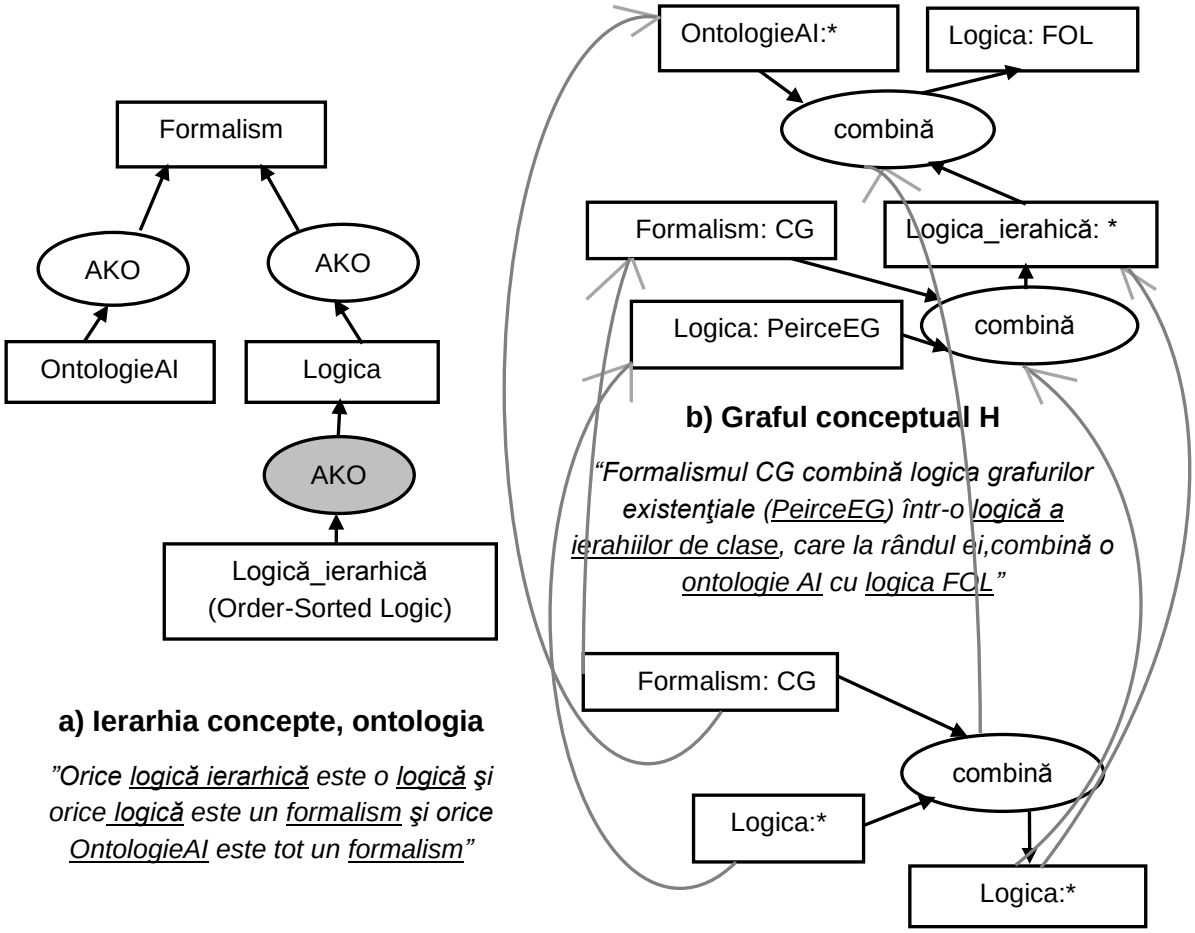
$\Phi_2 = \exists x \exists z(\text{Formalism}(CG) \wedge \text{Logica}(x) \wedge \text{Logica}(z) \wedge \text{combina}(CG, x, z))$ (= Φ(G))

$\Phi_0 \wedge \Phi_1 \vdash \Phi_2$

Observaţii:

1. Folosind operatorul Φ, propus de Sowa, formula ataşată grafului G este Φ_2.
2. Folosind operatorul Ψ, propus de Chein, formula ataşată grafului G este:

   $\Psi(G) = \exists x_1 \exists x_2 \exists x_3 \exists x \exists z\, \text{Formalism}(CG, x_1) \wedge \text{Logica}(x, x_2) \wedge$
   $\wedge \text{Logica}(z, x_3) \wedge \text{combina}(x_1, x_2, x_3)$



a) Ierarhia concepte, ontologia

"Orice logică ierarhică este o logică şi orice logică este un formalism şi orice OntologieAI este tot un formalism"

b) Graful conceptual H

"Formalismul CG combină logica grafurilor existenţiale (PeirceEG) într-o logică a ierahiilor de clase, care la rândul ei,combină o ontologie AI cu logica FOL"

c) Graful conceptual G

"Formalismul CG combină două (nu neapărat diferite) logici"

Pornind de la ontologia din figura a şi un fapt existent (reprezentat prin graful din figura b), prin derivare, se obţine un nou fapt (reprezentat prin graful din figura c)

Figura AI - 11 Un exemplu de raţionament cu grafuri conceptuale

Aşa cum se va demonstra în lucrare, puterea formalismului grafurilor conceptuale este dată şi de multitudinea posibilităţilor de a combina şi interpreta [203] un grup restrâns de primitive, cu scopul de a reprezenta cunoştinţe (pedagogice) diferite, la niveluri diferite.

AI.2.4 Limbaje de reprezentare a ontologiilor pentru web

Pentru reprezentarea informaţiei[65] din web-ul semantic şi interoperabilitatea acesteia atât din punct de vedere sintactic, cât şi semantic, se utilizează ontologii ("*Ontolgiile aplicate web-ului au*

[65] În 2002, în web existau peste 3 bilioane de documente statice şi peste 300 de milioane de utilizatori.

creat web-ul semantic" [204]) şi limbaje specifice [205]. Tim Berners-Lee propune un model stratificat al limbajelor pentru web-ul semantic [206], [168] prezentat în figura AI - 12.

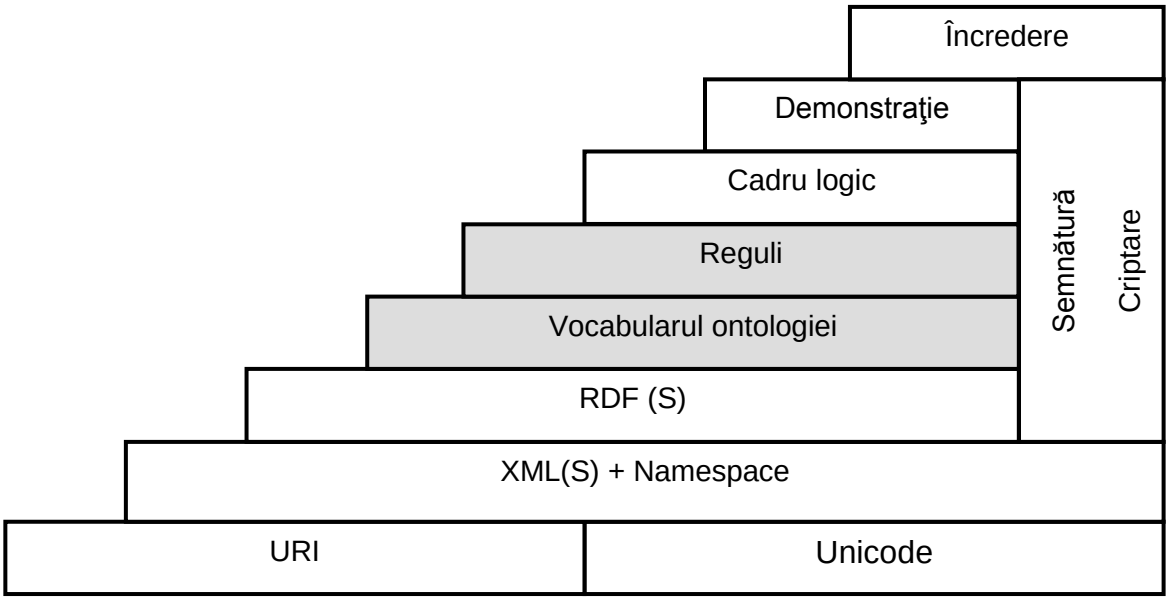


Figura AI - 12 Limbaje pentru web-ul semantic, adaptat după *[190]* (Berners-Lee, 2000)

Nivelul de la baza stratificării din figura AI - 12 este furnizat de către URI (**U**niform **R**esource **I**dentifier), care pemite referirea unei resurse web (anonime sau nu) pe baza unei scheme[66]. Folosind Unicode şi URI, *nivelul XML* (propus ca standard pentru interschimbul informaţiei din web) defineşte o reprezentare generică a documentelor, structurate sub formă de arbore; mecanismul spaţiilor numelor permite combinarea documentelor (eterogene) xml şi facilitează integrarea informaţiilor provenite din surse diferite.

Nivelul RDF(S) reprezintă cadrul pentru reprezentarea metadatelor (RDF), furnizează mecanismele pentru descrierea resurselor identificabile prin URI şi defineşte vocabularul folosit (RDF(S)).

La *nivelurile următoare* (cele reprezentate în gri în figura AI - 12) sunt situate limbajele ("***veritabile"***) (Studii comparative ale acestor limbaje în [205], [207], [208]) de reprezentare a ontologiilor pe web:

- SHOE, XOL [207], OIL, OIL+DAML, bazate pe sintaxa XML/RDF.
- **OWL** (succesorul DAML+OIL), bazat pe logicile de descriere, compus din OWL LITE, OWL DL şi OWL FULL.

Următoarele niveluri nu sunt încă deplin conturate, constituind încă direcţii de cercetare.
În ceea ce priveşte nivelul logic, se lucrează la o **logică comună**, CL (**C**ommon **L**ogic) [210], gândită ca un cadru al limbajelor bazate pe logică. Hayes [209] defineşte un model foarte general pentru teoria modelului pentru CL şi foloseşte CL pentru definirea semanticii RDF(S) şi

[66] Exemple de scheme URI: http, mailto, ldap, ftp; schemele definesc semantica şi structura unui URI.

OWL. În anul 2007 a apărut standardul ISO/IEC24707 [211] cu specificațiile pentru trei **dialecte:**

- **CLIF (Common Logic Interchange Format),**
- **CGIF (Conceptual G**raph **I**nterchange **F**ormat) și
- **XCL** (notație bazată pe XML pentru CL).

RDF şi OWL pot fi considerate dialecte care exprimă submulţimi ale semanticii CL, deoarece orice propoziţie din RDF sau OWL poate fi translatată în CLIF, CGIF sau XCL, însă doar o submulţime a logicii comune poate fi translatată în RDF, OWL [210].

Aşa cum se observă şi din figura AI - 13, limbajele care exploatează caracteristicile web-ului sunt limbaje de marcare, fondate pe xml. Chiar dacă *nu* toate sunt *veritabile limbaje de reprezentare a ontologiilor*, vom evidenţia câteva caracteristici ale fiecăruia dintre aceste limbaje.

- Limbajul **XML** (e**X**tensible **M**arkup **L**anguage), dezvoltat de XML Working Group de la W3C[67], furnizează nivelul sintactic de structurare a informaţiei din web. XML-ul poate constitui *baza unui limbaj de specificare* a ontologiilor: specificaţia sintactică se realizează în cadrul unui DTD (Document Type Definition), care desemnează un format standard al unui fişier text în care sunt reprezentate structuri arborescente. XML(S) defineşte tipurile care pot fi utilizate într-un RDF. La ora actuală există numeroase instrumente care permit utilizatorului să-şi definească propriile tag-uri şi atribute, dar şi propriile structuri de date (inclusiv cele imbricate), intrumente de extragere a datelor din documentele .xml, instrumente de validare a documentelor .xml. XML-ul oferă un mod de specificare a sintaxei pentru un limbaj ontologic, însă este nevoie de crearea unor instrumente care să permită inferenţe în limbajele bazate pe xml.

- Limbajul **RDF (S)** (**R**essource **D**escription **F**ramework (**S**chema)) propus de W3C [212] furnizează un model de descriere a semanticii datelor bazate pe xml.

  RDF este limbaj de tip reţea semantică pentru descrierea resurselor web. Modelul datelor RDF conţine *resurse (subiecte)* identificate prin URI, *proprietăţile (predicate)* (caracteristici, atribute sau relaţii care descriu resursele) lor şi *aserţiuni* de forma unor triplete (*subiect, predicat, valoare)*.

  RDF Schema (RDF Vocabulary Description Language) este un *limbaj declarativ bazat pe cadre*, care permite definirea relaţiilor între proprietăţi şi resurse, dar şi organizarea acestora în taxonomii (prin relaţia *subProprietyOf* şi *subClassOf*). Specifică schema de reprezentare a cunoştinţelor ontologice folosite în propoziţiile RDF (deoarece permite definirea unui vocabular).

  De aceea, RDF(S) poate fi considerat un limbaj minimal pentru reprezentarea ontologiilor. Cum au remarcat numeroşi autori [213], [214], limbajul RDF(S) prezintă multe similarităţi cu limbajle grafurilor conceptuale: reprezentarea cunoştinţelor sub formă de grafuri [215], semantica din teoria modelului.

[67] W3C - World Wide Web

- Limbajele **SHOE, XOL, OIL, OIL+DAML**, fondate pe sintaxa XML/RDF:

  - **SHOE** (**S**imple **H**TML **O**ntology **E**xtension), dezvoltat iniţial ca extensie a HTML-ului, adaptat acum pentru XML, permite definirea unei ontologii care descrie clasificarea obiectelor şi a relaţiilor dintre acestea şi adnotarea paginilor html. O ontologie este specificată în limbajul SHOE ca o ierarhie de clase (numite categorii), relaţii între clase şi reguli inferenţiale exprimate printr-o formă simplificată a clauzelor Horn.

  - **XOL**, limbaj care permite doar specificarea conceptelor, taxonomiilor şi a relaţiilor binare, fără mecanisme de raţionament.

  - **OIL** (**O**ntology **I**nterchange **L**anguage) - Combină primitivele de modelare din limbajele bazate pe cadre cu raţionamentul formal din logicile de descriere. Se bazează pe OKBC, XOL şi RDF. O ontologie în OIL este organizată pe trei niveluri: nivelul obiectelor (al instanţelor), primul meta-nivel care conţine definiţii ale ontologiei şi al doilea meta-nivel care conţine informaţii despre ontologie (autor, etc.). Limbajul OIL [204] permite definirea (parţială) a conceptelor, relaţiilor, funcţiilor şi a axiomelor.

  - **DAML+OIL** - Limbajul DAML+OIL [216] este un limbaj care extinde RDFS, cu semantică formală clară, bazat pe logicile de descriere. Permite definirea claselor, proprietăţilor, specificarea unor constrângeri de cardinalitate sau ale proprietăţilor şi ale unor caracteristici ale acestora (tranzitivitate, inversă, etc.), dar şi operaţii asupra claselor cu interpretare în teoria modelului.

O ontologie dezvoltată în aceste limbaje (SHOE, XOL, OIL, OIL+DAML) este convertită, de obicei, într-o formă xml/rdf.

- Limbajul **OWL** (**W**eb **O**ntology **L**anguage), compus din OWL Lite, OWL DL şi OWL Full.
  Începând din 2004 limbajul OWL a fost recomandat de către W3C ca limbaj *standard* de reprezentare a ontologiilor pe web. A fost dezvoltat pornind de la logicile de descriere şi DAML+OIL şi constă într-o mulţime de elemente XML şi atribute, cu semnificaţie bine definită, care specifică termenii şi relaţiile dintre aceştia (*Class*, *equivalentProperty*, *intersectionOf*, *unionOf*, etc.). Limbajul oferă şi posibilităţi de integrare a ontologiilor existente [217].

  Permite diferite tipuri de raţionament:
  - OWL Lite – este limbajul cel mai simplu, mai puţin expresiv, care permite clasificări ierahice şi specificarea constrângerilor de cardinalitate 0 şi 1.
  - OWL DL – este un limbaj cu grad mai mare de expresivitate, raţionamentul (decidabil) fiind cel din logicile de descriere.
  - OWL Full – este limbajul cu expresivitate maximă şi mecanisme de raţionament complexe, dar fără garanţia decidibilităţii (OWL Full $\supset$ OWL DL $\supset$ OWL Lite).

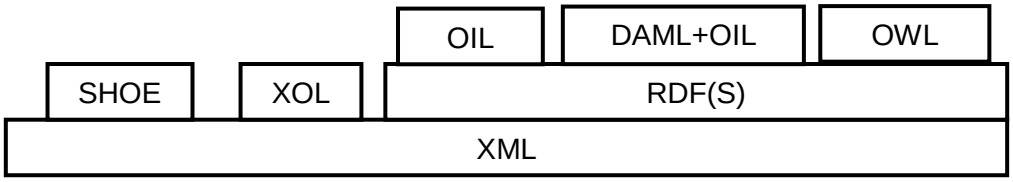


Figura AI - 13 Limbajele de reprezentare a ontologiilor pentru web

AI.3. METODOLOGII, METODE ŞI INSTRUMENTE ÎN INGINERIA ONTOLOGICĂ

Cercetătorii din ingineria ontologică recomandă ca la dezvoltarea unei ontologii să se aplice metodologiile prezentate în secțiunea 3 a anexei și să se repecte o serie de principii (schematizate în figura AI - 17).

Fiind componente software, ontologiile trebuie privite ca obiecte tehnice evolutive, al căror ciclu de viaţă trebuie specificat. Activităţile din ingineria ontologică (figura AI - 14) sunt: *activităţi de pre-dezvoltare*, de gestiune a proiectului (planificare, control, asigurarea calităţii); *activităţi de dezvoltare* (specificare, conceptualizare, formalizare, implementare); *activităţi de post-dezvoltare* (întreţinere, utilizare); *activităţi complementare* (validare, evaluare, documentare).

Astfel, metodologiile şi instrumentele folosite în ingineria ontologică pot viza diverse aspecte din cele enumerate anterior; un studiu al metodologiilor şi instrumentelor folosite în ingineria ontologică poate fi găsit în [161], [208] aşa că în această anexă **vom prezenta doar aspectele importante din punctul de vedere al acestei lucrări**, *evidenţiind în principal metodologiile şi instrumentele pentru construirea ontologii lor şi evaluarea acestora*.

Au fost propuse numeroase metodologii[68], ***fără a se impune*** una anume sau a se ajunge la un consens, aşa că am prezentat aici doar ***elementele care sunt utile în demersul nostru***.

În [161] **metodologiile şi metodele** sunt clasificate în categoriile: *pentru construirea ontologiilor*, pentru *reinginerie*, pentru *construirea cooperativă*, pentru *unificare şi fuzionare*, pentru *evoluţie* şi pentru *evaluare*, iar **instrumentele** - în categoriile: instrumente pentru *construcţia ontologiilor*, pentru *fuziune şi integrare*, pentru *evaluare*, pentru *adnotare* pe baza ontologiei, pentru *memorare şi interogări*, pentru *întreţinere şi evoluţie*. Deşi cea mai folosită clasificare a instrumentelor [161] se bazează pe funcţionalităţile acestora, trebuie subliniat faptul că, de cele mai multe ori, un instrument înglobează mai multe funcţionalităţi (vezi şi secțiunea 3.2.5.1.), sub forma unor module de aplicaţie (modul pentru raţionament, modul pentru adnotare, etc.).

[68] Conform standardelor IEEE95 și IEEE90 o ***metodologie*** reprezintă o succesiune cuprinzătoare şi integrată de tehnici sau metode pentru creearea unei teorii generale despre modul în care ar putea fi realizată o categorie de activităţi; o ***metodă*** reprezintă un proces *ordonat* sau o procedură folosită în ingineria unui produs sau realizarea unui serviciu; o ***tehnică*** este o procedură tehnică şi managerială folosită pentru atingerea unui anumit obiectiv. Metodele şi tehnicile sunt componente ale unei metodologii; metoda necesită o ordine, tehnica – nu.

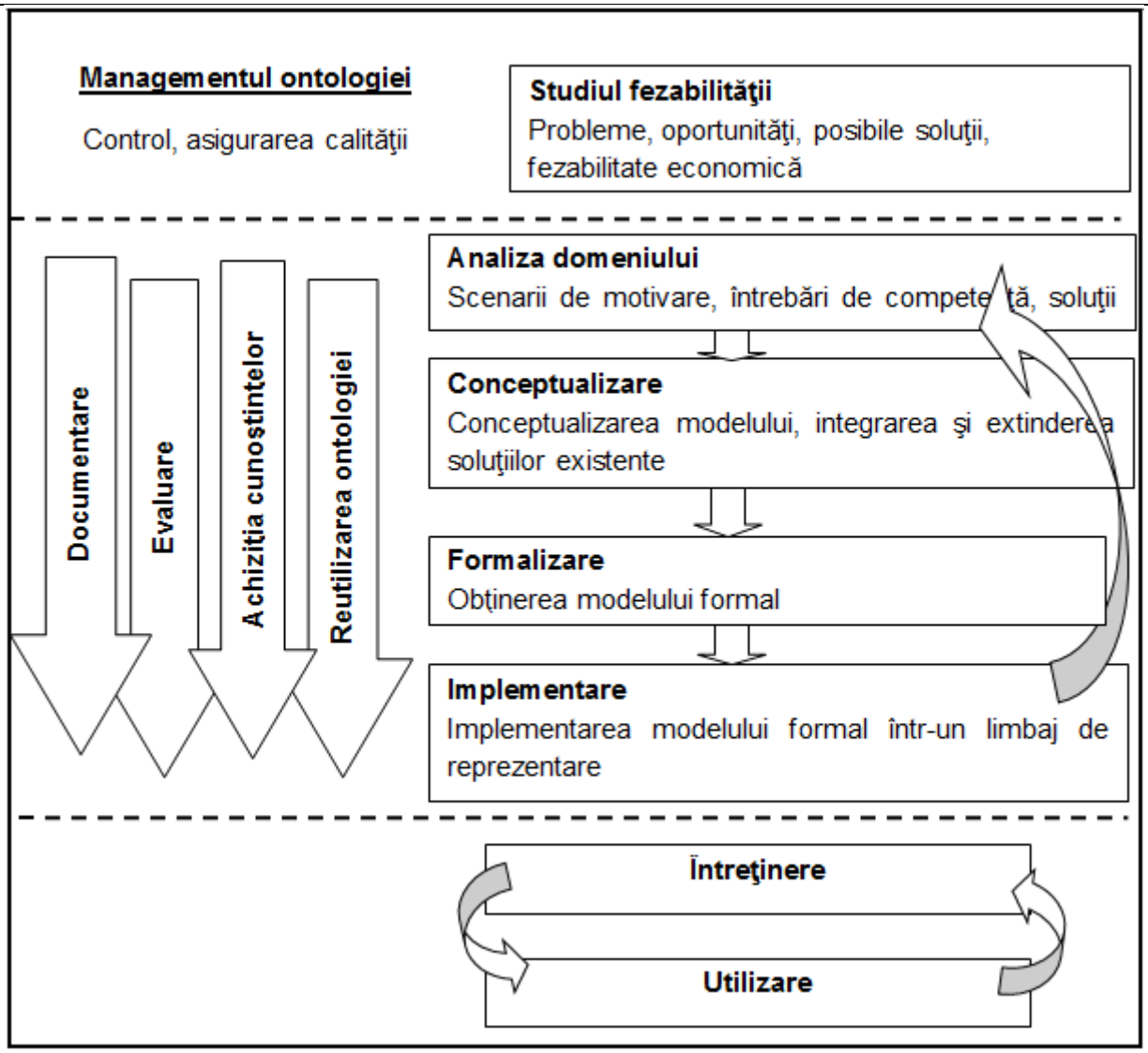


Figura AI - 14 Activităţi în ingineria ontologică

AI.3.1 Metodologii, metode şi instrumente pentru construirea ontologiilor

A. Metodologii şi metode pentru construirea ontologiilor

Metodologiile şi metodele existente pentru construirea ontologiilor (pornind de la zero sau reutilizând ontologiile existente) sunt evaluate [161] în funcţie de aspectele considerate în construcţia ontologiei[69] (ciclul de viaţă, *gradul de dependenţă faţă de aplicaţia care foloseşte ontologia* (dependente de aplicaţie, semi-dependente de aplicaţie sau cât mai puțin dependente de aplicaţie), *utilizarea unei ontologii nucleu ca punct de plecare în dezvoltarea unei ontologii a domeniului, modul de identificare a conceptelor* (bottom-up, top-down, middle-out (identificarea celor mai importante concepte, apoi specializarea sau generalizarea acestora)), *modalităţi de obţinere a unor taxonomii cât mai corecte* şi de acoperirea procesului de dezvoltare a ontologiei (bazat pe standardul IEEE 1075-1995 [171] pentru dezvoltarea software-ului): procese de

[69] Aspectele care ne interesează sunt scrise cu caractere italice.

management al proiectului, procese de dezvoltare (pre-dezvoltare (studiul mediului, studiu de fezabilitate, etc.), dezvoltare (cerinţe-specificaţii, proiectare, implementare) şi post-dezvoltare (instalare, mentenanţă, retragere)) şi *procese complementare* (evaluare, validare, documentare).

Vom prezenta, în continuare, cele mai reprezentative metode şi metodologii, insistând asupra celor de *construire a ontologiilor pornind de la zero*. În cadrul acestor metode/metodologii se urmăresc, în special, etapele propuse pentru construirea/dezvoltarea ontologiei, care sunt ontologiile dezvoltate pe baza unei anumite metode/metodologii şi domeniile lor de utilizare, ce aplicaţii care folosesc ontologiile respective au fost dezvoltate, precum şi existenţa unor instrumente suport.

Metododlogia Cyc[70]

- *Construirea unei ontologii* constă în etapele: (1) codificarea manuală a cunoştinţelor implicite şi explicite din diferite surse; (2) codificarea cunoştinţelor din KB Cyc cu instrumentele Cyc; (3) delegarea către instrumente a codificării. Fiecare etapă necesită dezvoltarea unei reprezentări a cunoaşterii (atribut, valoarea-atributului) şi utilizarea unei ontologii top-level (cu concepte abstracte, ca obiect, mulţime, etc).
- Ontologii dezvoltate aplicând metodologia: Cyc, Cyc.daml, folosite în proiectul HPKB (High Performance Knowledge Base – program de cercetare privind achiziţia, reprezentarea şi operaţionalizarea cunoaşterii).
- Domenii ale ontologiilor: Cyc are diferite micro-teorii, foloseşte o ontologie nucleu.
- Aplicaţii în care sunt utilizate ontologiile Cyc sunt module integrate într-o KB[71] Cyc şi motor de inferenţă: (1) *Sistem de integrare a bazelor de date eterogene* (mapează vocabularul Cyc în schemele bazelor de date); (2) *Modul de interogare în limbaj natural a unei baze de date de imagini*; (3) *Modul de integrare ghidată a terminologiei structurate* (permite importul, managementul şi integrarea dicţionarelor) (4) *Agenţi Cyc* (cunoştinţe din nucleul unei KB Cyc şi cunoştinţe din domeniul specific agentului); (5) *Modul de regăsire a informaţiei web* (acces KB în limbaj natural, extinderea KB cu informaţii web).
- Instrumente suport: instrumente Cyc.

Metoda Uschold&King[72]

- Propune pentru *dezvoltarea ontologiei*, următoarele etape: (1) identificarea scopului ontologiei; (2) construcţia ontologiei, constând în: (2a) ”capturarea ontologiei” (engl. “ontology capture”, echivalentă conceptualizării (identificarea conceptelor şi a relaţiilor din domeniu, definiţii (text) neambigue ale acestora, identificarea termenilor pentru concepte, relaţii); (2b) codificare (reprezentarea în limbaj formal); (2c) integrarea ontologiilor existente; (3) evaluare; (4) documentare.
- Ontologii dezvoltate prin această metodă: Ontologia întreprinderii (colecţie de termeni şi definiţii relevante în modelarea întreprinderii), în proiectul “Enterprise Project”, în domeniul întreprinderii.
- Aplicaţii: *Enterprise Toolset*, cu arhitectură multi-agent, compus din: *Procedure Builder* (construirea modelelor procesului), *Agent Toolkit* (dezvoltarea agenţilor), *Task Manager*

[70] http://www.cyc.com

[71] KB – bază de cunoștințe (engl., ”Knowledge Base”)

[72] http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html

(integrarea, vizualizarea şi suportul pentru hotărârea procesului) şi *Ontologia întreprinderi*i pentru comunicaţie.

Metodologia METHONTOLOGY

- *Construirea ontologiilor* în ingineria cunoştinţelor (construirea de la zero, reutilizarea altor ontologii existente, sau prin proces de reinginerie), propusă pentru construirea ontologiilor prin FIPA (Foundation for Intelligent Physical Agent) urmează etapele: (1) Identificarea proceselor de dezvoltare a ontologiilor (independente de aplicaţie) prin: (1a) gestiunea proiectului; (1b) activităţi de dezvoltare; (1c) activităţi ajutătoare; (2) ciclul de viaţă al ontologiei bazat pe prototipuri evolutive.
- Proiecte: MKBEEM, OntoWeb, Esperonto, Utilizarea ontologiilor în gestiunea cunoştinţelor, GUME, Prototipuri de ontologii pentru mediul ambiant.
- Ontologii: (1) CHEMICALS (conţine cunoştinţe din domeniul chimiei), cu: Monoatomic Ions Ontology, Environmental Polluants, Reference Ontology; (2) (KA)2 (conţine cunoştinţe ştiinţifice, subiecte de cercetare, proiecte, universităţi, etc.) în achiziţia cunoştinţelor.
- Domenii: chimie, mediu, ontologii, managementul cunoştinţelor, ştiinţa calculatoarelor, călătorii, etc.
- Aplicaţii: (1) *Ontogeneration*, sistem care foloseşte o ontologie a domeniului (CHEMICALS) şi una lingvistică (GUM) pentru răspunsuri sub forma unor descrieri text în spaniolă; (2) *Onto2Agent*, broker WWW bazat pe ontologii, care folosind Reference-Ontology regăseşte descrieri ale ontologiilor care satisfac anumite constrângeri; (3) *OntoRoadMap*, aplicaţie web bazată pe ontologii care permite membrilor unei comunităţi înregistrarea, căutarea şi navigarea prin ontologii, metodologii, instrumente şi limbaje, aplicaţii pentru web-ul semantic, e-commerce, managementul cunoştinţelor, prelucrarea limbajului natural, conferinţe, workshop-uri; (4) *ODEClean*; (5) prototipul *MKBEEM*.
- Instrumente: *ODE*, *WebODE* (platformă integrată de dezvoltare a ontologiilor).

Metodologia On-To-Knowledge[73]

- *Dezvoltarea unei ontologi*i conţine etapele: (1) Studiul fezabilităţii: (1a) identificarea problemei, (1b) studiul oportunităţii, (1c) alegerea publicului; (2) Conceptualizare, obţinându-se o descriere semi-formală a ontologiei (2a) specificarea cerinţelor, (2b) analiza surselor de informare, (2c) crearea ontologiei iniţiale; (3) Rafinare (3a) extragerea cunoştinţelor (3b) dezvoltarea taxonomiei de bază (3c) formalizare; (3d) adăugarea relaţiilor şi axiomelor; (4) Evaluare – revizuire şi extindere bazată pe acţiuni inverse (retroacţiuni) – bazată pe tehnologie, pe utilizatori, pe ontologie; (5) Întreţinere.
- Proiecte: On-To-Knowledge, OntoWeb, SemiPort, AIFB Website, COST – Modeling Real-Property Transactions.
- Ontologii: Skills Management @ SwissLife, Virtual Organization @ EnerSearch, OntoShare @ BT, OntoWeb Portal, AIFB Portal, în proiectul On-To-Knowledge.
- Instrumente: (1) *OntoEdit*, veritabil mediu de inginerie ontologică, sprijină toate fazele metodologiei, cu diverse plugin-uri: OntoMap, OntoFiller, Domain Lexicon, Inferencing; (2) *OntoMat-Annotizer*, (3) *OntoBroker*, (4) *MindManager2002 Business Edition*, (5) *Sesame*, (6) *Spectacle*, (7) *OntoShare*, (8) *OMM (Ontology Middleware Module)*.

[73] http://www.ontoknowledge.org

Metodologia Grüninger&Fox[74]

- Propune *construirea* unui model (integrat) logic (formalizat în FOL) al cunoaşterii unui domeniu (întreprindere) specificate printr-o ontologie.
- Construirea ontologiei (semi-dependente de aplicaţie) urmează etapele: (1) stabilirea scenariilor de motivare; (2) formularea întrebărilor de competenţă în limbaj natural; (3) specificarea terminologiei ontologiei în limbaj formal; (3) terminologia informală; (3b) terminologia formală; (4) specificarea formală a axiomelor şi a definiţiilor termenilor; (5) stabilirea condiţiilor de completitudine a ontologiei.
- Ontologii şi proiecte: TOVE (Toronto Virtual Enterprise) în domeniul întreprinderii.
- Aplicaţii: (1) *Enterprise Design Workbench* (mediu de proiectare a întreprinderii care permite explorarea şi analiza comparativă unor soluţii de proiectare existente, proiectare, reproiectare şi ghidarea acestora); (2) *Integrated Supply Chain Management Project Agents* (organizarea unei reţele de agenţi inteligenţi care cooperează în realizarea unor activităţi logistice, de transport, management, etc).

Metoda KAKTUS

- Investighează fezabilitatea reutilizării cunoaşterii în sisteme tehnice complexe, sprijinite de ontologii; metoda este condiţionată de *dezvoltarea unor ontologii pentru aplicaţii*.
- Metoda conţine etapele: (1) specificarea aplicaţiei (context, componente); (2) proiectarea preliminară bazată pe categoriile ontologice top-level; (3)rafinarea/structurarea ontologiei.
- Ontologii şi proiecte: ontologii ale reţelelor electrice, proiectul KAKTUS.
- Aplicaţii: (1) *Diagnoza defectelor în reţele electrice*; (2) *Programarea reluării serviciilor după o defecţiune*; (3) *Controlul reţelelor electrice pe baza aplicaţiilor anterioare*.

Metoda SENSUS

- Propune o *abordare top-down pentru obţinerea ontologiilor specifice domeniului* (prin construirea arborelui conceptual), pornind de la ontologii existente, ”mai generale” (obţinute prin extragerea şi unificarea informaţiilor din diferite surse (electronice) de cunoaştere).
- Etapele de *construire*: (1) identificarea termenilor cheie ai domeniului; (2) legarea manuală a acestora la SENSUS; (3) adăugarea conceptelor aflate în calea de la termen la rădăcina arborelui SENSUS; (4) selectarea dintre conceptele adăugate a celor care pot fi relevante pentru domeniu; (5) adăugarea sub-arborilor corespunzători conceptelor pentru care se identifică numeroase căi.
- Ontologii de planificare a campaniilor militare aeriene într-un proiect DARPA.
- Instrumente: *OntoSaurus*.

B. Medii şi instrumente pentru construirea ontologiilor

OilED: OilEd[75] (**Oil Ed**itor) este un editor de ontologii bazat pe formalismul DAML+OIL şi a logicilor de descriere. Permite în principal construirea ontologiilor, testarea coerenţei (cu ajutorul motorului de inferenţă FACT, bazat pe OIL) şi exportul ontologiei în alte limbaje[76].

[74] http://www.ie.utoronto.ca/EIL

[75] http://oiled.man.ac.uk

[76] Chiar şi limbaje mai puţin cunoscute, ca SHIQ

OntoEdit: OntoEdit[77] - distribuit atât ca versiune free, cât şi ca versiune profesională – este un mediu de inginerie ontologică pentru dezvoltarea şi întreţinerea ontologiilor, bazat pe metodologia On-TO-Knowledge. Permite editarea grafică a ierarhiilor de concepte şi relaţii. Utilizează un model de reprezentare a cunoaşterii bazat pe cadre şi permite exprimarea unor axiome algebrice asupra relaţiilor şi a proprietăţii de genericitate a unui concept [168](Buraga, 2007). Versiunea comercială oferă servicii de editare colaborativă, testarea coerenţei ontologiei, posibilitatea de generare a specificaţiilor[78] ontologiei prin intermediul întrebărilor de competenţă şi import/export în diferite limbaje de reprezentare a ontologiei.

ODE şi **WebODE**: ODE (**O**ntology **D**esign **E**nvironment) şi WebODE[79] sunt instrumente de inginerie ontologică care furnizează suportul şi serviciile necesare activităţilor implicate în procesul de dezvoltare şi utilizare a ontologiilor: interfaţă grafică de editare colaborativă şi de testare a ontologiei (verificarea constrângerilor tipurilor, a valorilor numerice, a cardinalităţii sau verificări de consistenţă taxonomică), memorarea ontologiei într-o bază de date relaţională, import/export XML, translatare în alte limbaje sau sisteme[80]. Metodologia aplicată în construirea ontologiei este METHONTOLOGY, modelul de reprezentare a cunoştinţelor este unul bazat pe cadre (exprimat în limbajul OCML), mecanismele operaţionale de raţionament sunt reguli logice (motor de inferenţă dezvoltat în Ciao Prolog: AxiomBuilder transformă axiomele FOL şi regulile în Prolog).

Protégé-2000: Protégé-2000[81] este un mediu grafic şi interactiv pentru proiectarea ontologiilor şi un mediu de dezvoltare a bazelor de cunoştinţe. Permite editarea, vizualizarea şi verificarea constrângerilor ontologiei, extragerea unei ontologii pornind de la surse textuale şi fuziunea semi-automată a ontologiilor. Modelul cunoştinţelor este bazat pe cadre, conţinând clase (concepte), slot-uri (proprietăţi) şi faţete (valori ale proprietăţilor şi ale constrângerilor). Deşi se doreşte un model compatibil OKBC, apar unele diferenţe: de exemplu, în Protégé-2000, un cadru – fie o clasă, fie un slot, fie o faţetă – este instanţa unei singure clase; în OKBC acelaşi cadru poate reprezenta mai multe obiecte conceptuale. Permite definirea meta-claselor.

Constrângerile pot fi exprimate în limbajul PAL[82]. Diferite motoare de inferenţă pot fi dezvoltate utilizând Flora (pentru F-logic) şi Jess.

Plugin-urile OntoViz şi Jambalaya permit obţinerea unor view-uri grafice asupra bazei de cunoştinţe, respectiv navigare interactivă, zooming asupra anumitor elemente din baza de date, evidenţierea conexiunilor între clustere de date. Plugin-ul PROMPT furnizează un mediu de management al ontologiilor, oferind instrumente de unificare (prin găsirea similarităţilor între diferite ontologii), de versionare, de extragere semantică a unor sub-părţi complete dintr-o ontologie sau de rearanjare a cadrelor în diferite ontologii relaţionate.

---

[77] http://www.ontoprise.de/com/start_downlo.htm

[78] Prin pulgin-ul ONTOKICK

[79] http://webode.dia.fi.upm.es/webODE

[80] Java sau Jess

[81] http://protege.stanford.edu

[82] PAL (Protégé Axiom Language) este un sub-limbaj al limbajului KIF: *fără* unele constante şi predicate din KIF, *fără* defrelation şi deffunction.

Ontolingua Server: Ontolingua Server[83] este un server de editare colaborativă a ontologiilor. Modelul cunoştinţelor este unul bazat pe cadre, în limbajul ONTOLINGUA (extensie a limbajului KIF). Într-o astfel de ontologie constituentele sunt clase, relaţii, funcţii, instanţe şi axiome. Furnizează accesul la o bibliotecă de ontologii (prin protocolul OKBC), un editor pentru crearea/consultarea ontologiilor şi numeroase translatoare între diferite limbaje de reprezentare (Prolog, CORBA IDL, CLIPS, LOOM, etc.). Permite includerea unei ontologii existente în ontologia în curs de construcţie, prin adăugarea la cea din urmă a unor axiome de "tranducere" (prin stabilirea unei relaţii de identitate între termenii care definesc aceleaşi clase sau relaţii în cele două ontologii).

OntoSaurus: OntoSaurus[84] constă în două module: un server de ontologii în care reprezentarea cunoştinţelor foloseşte LOOM şi un server de consultare a ontologiilor care oferă funcţionalităţi de afişare a ierarhiilor prin generarea dinamică a unor pagini html. Oferă translatoare din LOOM în Ontolingua, KIF, KRSS şi C++.

WebOnto: WebOnto[85] este un instrument pentru crearea, modificarea şi consultarea colaborativă a ontologiilor. Reprezentarea cunoştinţelor utilizează paradigma bazată pe cadre şi limbajul OCML. Instrumentul permite managementul ontologiilor prin intermediul unei interfeţe grafice, suport pentru rezolvarea problemelor şi modelarea task-urilor, verificarea consistenţei, realizarea unor adnotări (folosind Tadzebao).

DOE: DOE[86] (**D**ifferential **O**ntology **E**ditor), conceput mai degrabă ca şi complement al altor editoare de ontologii, permite construirea ontologiilor conform metodologiei propuse de Bachimont, în care ierarhiile de concepte şi de relaţii sunt construite iniţial folosind principiile diferenţiale, apoi se adaugă principiile referenţiale. Semantica relaţiilor este precizată prin semnătura acestora. Este posibilă exportarea ontologiei în xml, CGXML, DAML+OIL, RDFS şi OWL.

AI.3.2 Metodologii, metode şi instrumente pentru evaluarea ontologiilor

Evaluarea unei ontologii constă în validarea şi verificarea acesteia:

- **Validarea**: pentru a ne asigura că este corect construită din punct de vedere al modelului de reprezentare a cunoştinţelor adoptat (ontologia este conformă cu modelul formal de reprezentare); ca urmare, la acest nivel, evaluarea constă într-un proces formal *dependent de modelul de reprezentare* şi *independent de domeniul modelat*. La acest nivel sunt testate trei tipuri de proprietăţi ale ontologiei:
  - *Conformitatea* ontologiei cu modelul de reprezentare a cunoştinţelor (de exemplu, într-un model bazat pe grafuri conceptuale sau un model entitate/relaţie, este necesară precizarea semnăturii fiecărei relaţii);

[83] http://ontolingua.stanford.edu

[84] http://www.isi.edu/isd/ontosaurus.html

[85] http://kmi.open.ac.uk/projects/webonto, http://webonto.open.ac.uk

[86] http://opales.ina.fr/public

- *Coerenţa ontologiei*: absenţa contradicţiilor logice (de exemplu, două axiome sintactic corecte pot fi logic contradictorii) ;
- *Minimalitatea ontologiei*: nu conţine cunoştinţe inutile (cum ar fi concepte specificate de mai multe ori, axiome care pot fi deduse dintr-o combinaţie de alte axiome).

- **Verificarea**: pentru a ne asigura că semantica exprimată prin ontologie este conformă cu domeniul modelat. La acest nivel care vizează, practic, o evaluare din punct de vedere conceptual, sunt testate:
  - *Completitudinea ontologiei* faţă de domeniul vizat (dacă toate cunoştinţele domeniului sunt incluse în ontologie);
  - *Conformitatea ontologiei* faţă de domeniul vizat (cunoştinţele reprezentate în ontologie corespund întru totul semanticii domeniului).

Metodologia lui Goméz-Pérez de evaluare a taxonomiilor

Deoarece majoritatea ontologiilor sunt bazate pe un model al cadrelor, Gómez-Pérez identifică un număr de posibilile erori în construirea taxonomiilor sau într-o bază de cunoştinţe reprezentate printr-un model bazat pe cadre. Aceste erori pot fi:

- Erori de inconsistenţă:
  - Erori de circularitate;
  - Erori de partiţii (clasificarea conceptelor este definită cu ajutorul partiţiilor, într-o manieră disjunctă şi/sau completă);
  - Erori de inconsistenţă semantică.
- Erori de incompletitudine:
  - Clasificarea incompletă a conceptelor;
  - Erori de partiţii (apărute prin lipsa definiţei unei partiţii între o mulţime de clase).
- Erori de redundanţă:
  - Erori de redundanţă gramaticală;
  - Aceeaşi definiţie formală pentru mai multe clase;
  - Aceeaşi definiţie formală pentru mai multe instanţe.

Un ***instrument*** care aplică această metodologie este ***ONE-T*** (**O**ntology **E**valuation **T**ool), o aplicaţie Java bazată pe web, care permite verificarea ontologiilor Ontolingua Server şi LOOM.

Metodologia OCM (**O**ntological **C**onstraint **M**anager)

Metodologia vizează rolul axiomelor în ingineria ontologică şi restricţionarea interpretărilor pe care construcţiile ontologice le pot avea. OCM propune o arhitectură-multinivel, în care fiecare nivel reprezintă un cuplu specificaţie/ontologie. Pentru fiecare nivel (deci pentru fiecare cuplu), se verifică conformitatea specificaţiei cu ontologia corespunzătoare.

Instrumentul care aplică această metodologie este ***OCM***, o aplicaţie Java, cu motor SICTus Prolog.

Metodologia propusă de Guarino

Metodologia evaluează taxonomiile de concepte prin verificarea constrângerilor impuse prin meta-proprietăţile conceptelor (rigiditate, identitate, unitate şi dependenţă) [160].

Pentru proprietăţile Φ şi Ψ, constrângerile[87] sunt:

- Constrângeri de rigiditate ($\Phi^{\sim R}$ nu poate subsuma Ψ^{+R}, Φ^{-R} trebuie să subsumeze $\Psi^{\sim R}$, $\Phi^{\sim R}$ trebuie să subsumeze $\Psi^{\sim R}$);
- Constrângeri de identitate: proprietăţile cu condiţii de identitate incompatibile sunt disjuncte;
- Constrângeri de unitate (Φ^{+U} nu poate subsuma Ψ^{+U}, $\Phi^{\sim U}$ nu poate subsuma Ψ^{+U}, $\Phi^{\sim U}$ trebuie să subsumeze $\Psi^{\sim U}$);
- Constrângeri de dependenţă (Φ^{+D} trebuie să subsumeze Ψ^{+D}, iar Φ^{+D} nu poate subsuma Ψ^{-D}).

Instrumentul care implementează metodologia este ***OntoClean*** [218], un plug-in pentru WebODE şi pentru Protégé. Pentru evaluarea unei taxonomii, se parcurg următoarele etape: se atribuie fiecărei proprietăţi meta-proprietatea corespunzătoare; se consideră doar proprietăţile rigide (o taxonomie ***fără*** proprietăţi rigide constituie osatura); se evaluează taxonomia pe baza constrângerilor; se analizează proprietăţile rigide; se completează taxonomia cu alte concepte şi relaţii.

Chiar dacă nu aplică una dintre metodologiile prezentate aici, trebuie amintite ***alte două instrumente*** (plugin-uri pentru OntoEdit) de evaluare a ontologiilor, ***OntoAnalyzer*** şi ***OntoGenerator***.

- ***OntoAnalyzer*** evaluează proprietăţile ontologiei, conformitatea limbajului şi coerenţa. Constrângerile şi proprietăţile ontologiei sunt exprimate prin reguli, în F-Logic; motorul de inferenţă este OntoBroker.

- ***OntoGenerator*** evaluează performanţa (ca de exemplu timpul necesar pentru realizarea unui anumit task de inferenţă, timpul necesar pentru memorare, etc.) şi scalabilitatea ontologiei (comportamentul memoriei în condiţii de lucru cu ontologii de diferite mărimi). OntoGenerator testează aşa-numitele “ontologii sintetice” generate automat (nu pentru un anumit domeniu) cu anumiţi parametri tehnici (adâncimea arborelui taxonomic, un anumit număr de concepte/relaţii/instanţe, anumite tipuri de reguli, etc.).

Aceeași metodologie a fost considerată și în metoda MIIO (propusă în capitolul 2).

AI.3.3 Alte metodologii, metode şi instrumente utilizate în ingineria ontologică

A. Metode şi instrumente pentru reingineria ontologiilor

Reingineria ontologică este procesul de regăsire şi mapare a modelului conceptual al unei ontologii existente într-un alt model, care este reimplementat. Cercetătorii din “Ontology Group

[87] Notaţia Φ^{M} este interpretată astfel:

- Proprietatea Φ are *meta-proprietatea* M.
- Metaproprietăţile M, au forma: +R, -R sau ~R (ceea ce înseamnă că - în exemplul dat - o anumită proprietate *este rigidă : +R*, *nu este rigidă: -R* sau *este anti-rigidă: ~R*).

of Artificial Intelligence Laboratory at UPM“ a propus o **metodă de reinginerie**, constând în **trei activităţi**: inginerie inversă, restructurare şi inginerie (engl., “forward engineering”).

B. Metode pentru construirea cooperativă a ontologiilor

Principale metode propuse pentru construirea cooperativă a ontologiilor sunt **CO4** (pentru construirea cooperativă a bazelor de cunoştinţe consensuale) şi **(KA)2**.

- **Metoda CO4** (**Coo**perative C**o**nstruction of C**o**nsensual knowledge bases) propune organizarea bazelor de cunoştinţe într-o formă arborescentă, în care nodurile frunze sunt numite *baze de cunoştinţe utilizator*, iar nodurile intermediare – *baze de cunoştinţe de grup*. Fiecare bază de cunoştinţe de grup conţine cunoştinţele consensuale ale bazelor de cunoştinţe fii, iar o bază de cunoştinţe poate subscrie unui singur grup.

- **Metoda (KA)2**[88] îşi propune modelarea achiziţiei cunoştinţelor prin utilizarea ontologiilor dezvoltate în locaţii diferite pe baza aceloraşi şabloane sau limbaje.

C. Metodologii, metode şi instrumente pentru achiziţia automată (semi-automată) a cunoştinţelor domeniului au ca scop automatizarea parţială a procesului de achiziţie a cunoştinţelor şi utilizează în principal tehnici de prelucrare a limbajului natural şi tehnici de învăţare automată.

- **Metoda Aussenac-Gille** (numită “proiectarea ontologiilor pentru texte”) constă în următoarele etape: (1) achiziţia cunoştinţelor pornind de la texte-sursă; (2) conectarea textelor la modelele conceptuale; (3) explorarea textelor prin instrumente de prelucrare a limbajului natural şi tehnici lingvistice.
  Instrumentele care utilizează metoda sunt *LEXTER*, *LINGUAE*, *TERMINAE*.

- **Metoda Bachimont**, bazată pe semantica diferenţială, propune trei etape de construire a ontologiei: (1) alegerea de către utilizator a termenilor relevanţi şi normalizarea acestora (utilizatorul specifică locul conceptului în cadrul ierarhiei cu ajutorul semanticii diferenţiale: similarităţile cu conceptul-părinte şi diferenţele faţă de conceptele-fraţi); (2) formalizarea cunoştinţelor, inclusiv exprimarea unor constrângeri asupra domeniului unei relaţii, definirea noilor concepte, adăugarea unor proprietăţi pentru ele sau adăugarea unor axiome); (3) transcrierea ontologiei într-un limbaj de reprezentare.
  Instrument: ***DOE***

- **Metoda lui Maedche**, în care procesul de achiziţie a cunoaşterii porneşte cu selecţia unei ontologii-nucleu (engl., “core-ontology”) la care se adaugă conceptele specifice domeniului (existente în documentele electronice), utilizând doar relaţiile taxonomice. Sunt adăugate apoi relaţiile netaxonomice. Prin aplicarea unor metode de învăţare sunt deduse noi relaţii care sunt adăugate la cele existente.
  Instrument: ***Text-To-Onto***

- **Metoda Nobecourt** (numită si “metoda de construire a ontologiilor formale pornind de la texte”) constă în două activităţi: (1) Modelarea: primitivele conceptuale (termenii domeniului) sunt extrase dintr-un corpus, expertul identifică termenii relevanţi şi îi modelează ca proprietăţi sau concepte, obţinând astfel o primă schiţă a ontologiei. Conceptele, descrise în limbaj natural, constituie un nou document sursă folosit iar ca

[88] Dezvoltată de comunitatea de achiziţie a cunoştinţelor, recunoscută ca iniţiativa (KA)2.

intrare a metodei, cu scopul de a găsi noi primitive conceptuale: astfel, ontologia este rafinată succesiv. (2) Implementarea.
Instrument: ***TERMINAE***

D. Metodologii, metode şi instrumente pentru fuzionarea/alinierea ontologiilor

Alinierea a două sau mai multe ontologii, necesară în situaţiile în care se doreşte utilizarea lor în cadrul aceluiaşi SBC, constă în găsirea corespondenţei între cunoştinţele specificate în ontologii; constă, practic în alinierea sintactică (identificarea conceptelor sau relaţiilor din ontologia O_1 cu cele din ontologia O_2) şi/sau alinierea semantică (identificarea unor legături conceptuale, ca cea de subsumare, între cele două ontologii). În cazul alinierii, ontologiile iniţiale nu se modifică. ***Fuzionarea*** a două ontologii O_1 şi O_2 constă în obţinerea unei a treia ontologii, O_3, care integrează cunoştinţele conţinute în O_1 şi O_2.

Metodele aplicate pentru găsirea similarităţii între concepte şi/sau relaţii sunt:
- Metode terminologice, care compară etichetele conceptelor/relaţiilor, putând utiliza şi resurse externe (tabelele de sinonimie şi/sau de hiperonimie) ;
- Metode care compară proprietăţile interne ale conceptelor şi ale relaţiilor (atributele conceptelor, semnătura relaţiei, etc.);
- Metode care compară proprietăţile externe ale ale conceptelor şi ale relaţiilor (subsumare, alte relaţii între concepte);
- Metode care compară extensiunile conceptelor şi ale relaţiilor (instanţele lor);
- Metode care compară semantica conceptelor şi a relaţiilor.

- **Metoda FCA[89]-Merge** este o metodă de fuzionare bottom-up, care porneşte de la ontologiile sursă (O_1 şi O_2) şi de la o mulţime de documente (D) relevante pentru ambele ontologii. În prima etapă, din mulţimea documentelor, prin tehnici de prelucrare a limbajului natural, se extrag instanţele conceptelor. În a doua etapă, se determină automat contextul formal al fiecărei instanţe apoi, semi-automat - prin intervenţia inginerului de cunoştinţe - se realizează fuzionarea celor două contexte formale.

- **Metoda PROMPT** adoptă o abordare terminologică pentru identificarea analogiilor între primitivele conceptuale şi propune un algoritm semi-automat de fuzionare şi aliniere a ontologiilor: realizează automat unele task-uri, ghidează utilizatorul în realizarea unor task-uri, determină inconsistenţele din ontologie şi sugerează modalităţi de rezolvare a acestora. Paşii propuşi prin metodă: (1) fuzionarea claselor; (2) fuzionarea sloturilor; (3) fuzionarea legăturilor între un slot şi o clasă; (4) realizarea unei copii în adâncime a unei clase dintr-o ontologie în alta; (5) realizarea unei copii de suprafaţă a unei clase; (6) rezolvarea conflictelor.
  Un alt algoritm propus prin această metodă este cel în care ontologia este privită ca un graf ale cărui noduri sunt clase şi sloturile - arce. Având ca date de intrare două ancore (perechi de termeni propuşi de utilizator sau identificaţi automat prin punerea în corespondență semantică), algoritmul analizează căile din subgraful delimitat prin ancore şi determină apariţiile frecvente ale claselor în poziţii similare pe căi similare. Aceste clase reprezintă concepte similare semantic.
  Instrumente: plugin-urile ***PROMPT*** şi ***AnchorPROMPT*** pentru Protégé.

[89] FCA – analiza formală a conceptelor (engl., “Formal Concept Analysis”)

- **Metodologia propusă de Diego**, constă în următorii paşi: transformarea formatelor ontologiilor care se doresc unificate, evaluarea ontologiilor, unificarea ontologiilor, evaluarea rezultatului, transformarea ontologiei rezultate pentru adaptarea acesteia la aplicaţia în care va fi folosită. Intrări: Ontologiile sursă, O_1 şi O_2, tabela de sinonime (care conţine relaţiile de sinonimie între termenii din O_1 şi cei din O_2, tabela relaţiilor de hiperonimie (*superclasă-pentru*). Ieşire: Ontologia unificatoare, O_3.
  ***Instrument:* ODEMerge**.

AI.4. PRINCIPII DE BAZĂ ÎN INGINERIA ONTOLOGICĂ

Analiza acestor principii – prezentată în continuare - evidenţiază modul şi fazele de aplicare a acestora:

- *Claritate şi obiectivitate*: ontologia trebuie să furnizeze semnificaţia termenilor prin definiţii obiective şi documentaţie în limbaj natural [103]; principiul se aplică atât în faza de conceptualizare, cât şi în cea de evaluare tuturor entităţilor din ontologie.
- *Completitudine*: definiţiile exprimate prin condiţii necesare şi suficiente sunt preferabile definiţiilor parţiale (exprimate prin condiţii necesare sau suficiente) [103]; principiul se aplică în faza de evaluare tuturor entităţilor din ontologie.
- *Coerenţă*: să permită inferenţe consistente cu definiţiile [103]; principiul se aplică în faza de conceptualizare, în cea de formalizare şi în cea de evaluare şi vizează conceptele, instanţele şi rolurile deduse din axiome.
- *Extinderea monotonică maximă*: termeni noi (generali sau specializaţi) trebuie să poată fi incluşi în ontologie fără să necesite revizuirea definiţiilor existente [103]; principiul se aplică atât în faza de rafinare a ontologiei şi vizează toate entităţile ontologiei.
- *Angajamente ontologice minimale* [103]; se aplică în faza de cerinţe-specificaţii şi conceptualizare şi vizează întreaga ontologie.
- *Principiul distincţiei ontologice*: clase disjuncte în ontologie [208]; se aplică în special în faza de evaluare, dar şi de conceptualizare şi vizează conceptele şi relaţiile din ontologie.
- *Diversificarea ierarhiilor* prin utilizarea mecanismelor de moştenire multiplă [208]; se aplică în special la evaluare, dar şi la conceptualizare şi vizează conceptele şi relaţiile.
- *Modularitate*: minimizarea cuplajului între module (vezi ingineria software); principiul se aplică în faza de conceptualizare, în special în cazurile de integrare a unor ontologii existente şi vizează ontologiile reutilizate.
- *Minimizarea distanţei semantice dintre concepte "frate"*: conceptele similare sunt grupate şi reprezentate prin aceleaşi primitive [208]; se aplică în special la evaluare, dar şi la conceptualizare şi vizează conceptele şi rolurile (proprietăţi, atribute).
- *Standardizarea numelor*, acolo unde este posibil; se aplică la conceptualizare şi evaluare.

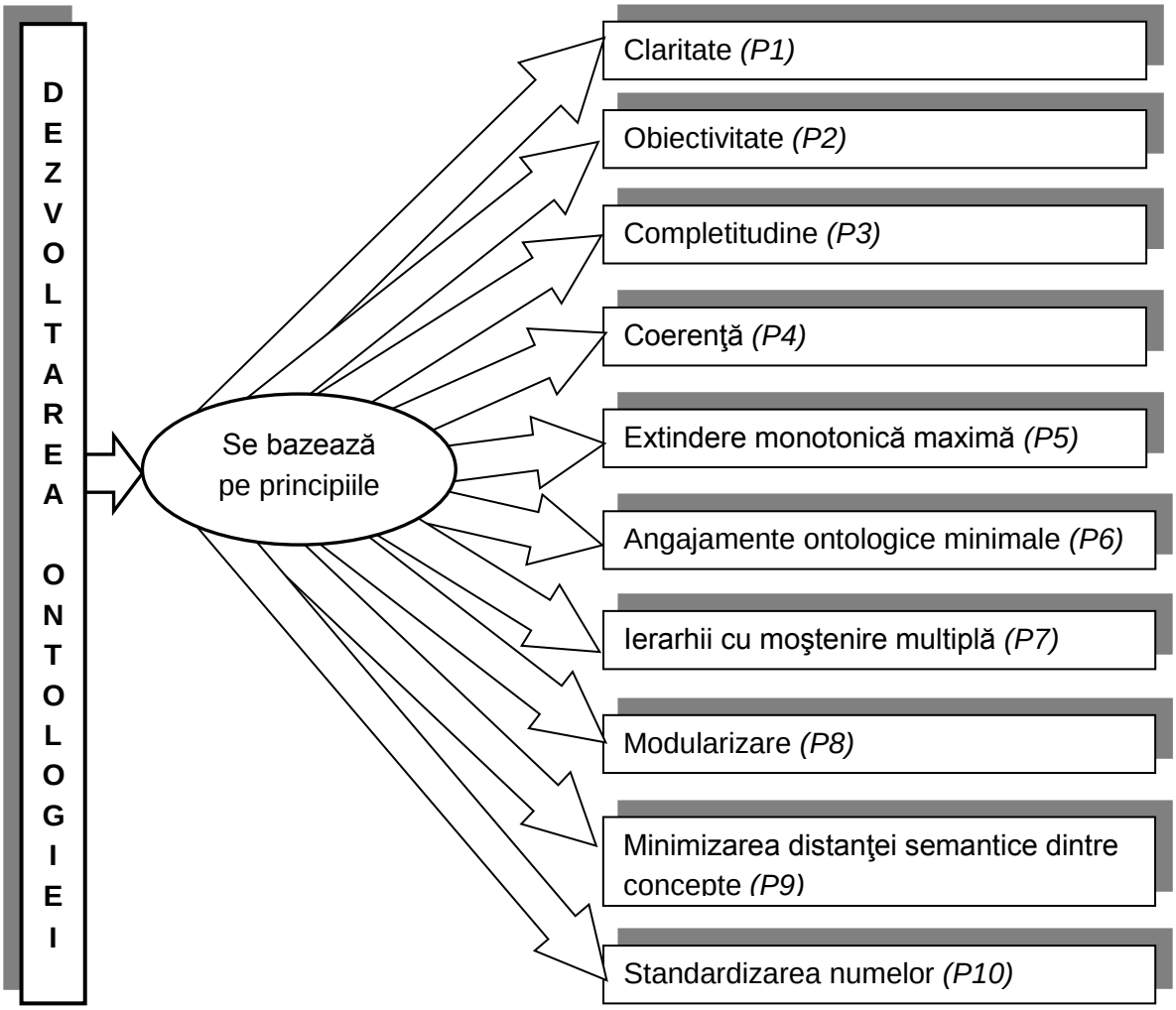


Figura AI - 15 Principiile ingineriei ontologice

II. ANEXA II - APLICAȚIA OntL_CG

Înainte de lansarea aplicației client, trebuie realizată conectarea la server-ul CoGITaNT (lansarea server-ului se poate face din linie de comandă sau din Meniul Options). Utilizatorul este informat asupra reușitei sau nereușitei conectării la server-ul CoGITaNT. În cazul în care acest lucru nu se realizează, se afișează eroarea într-o fereastră; deasemenea, dacă o conexiune există deja, utilizatorul este avertizat (figura AII - 1).

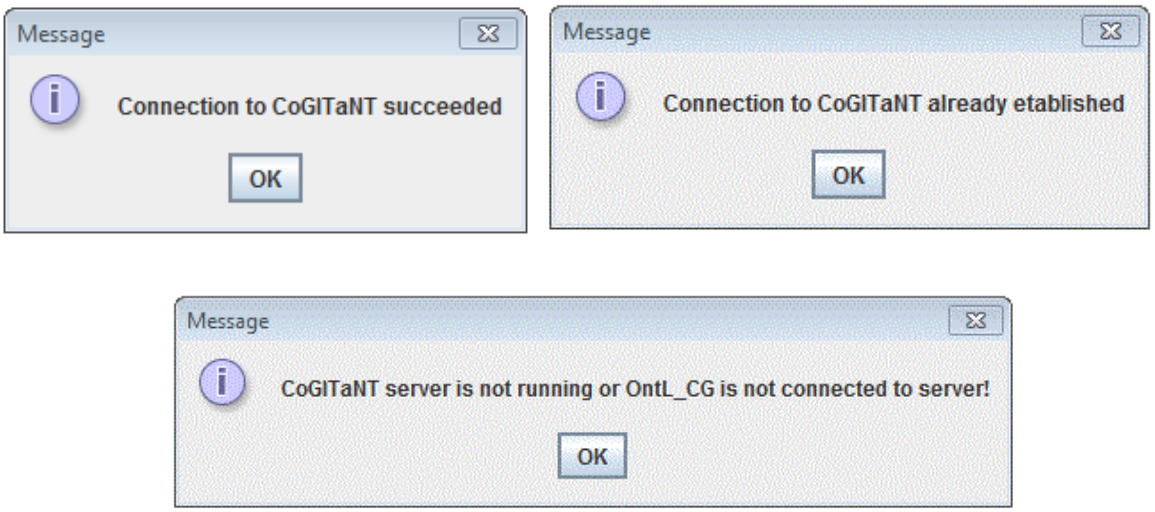
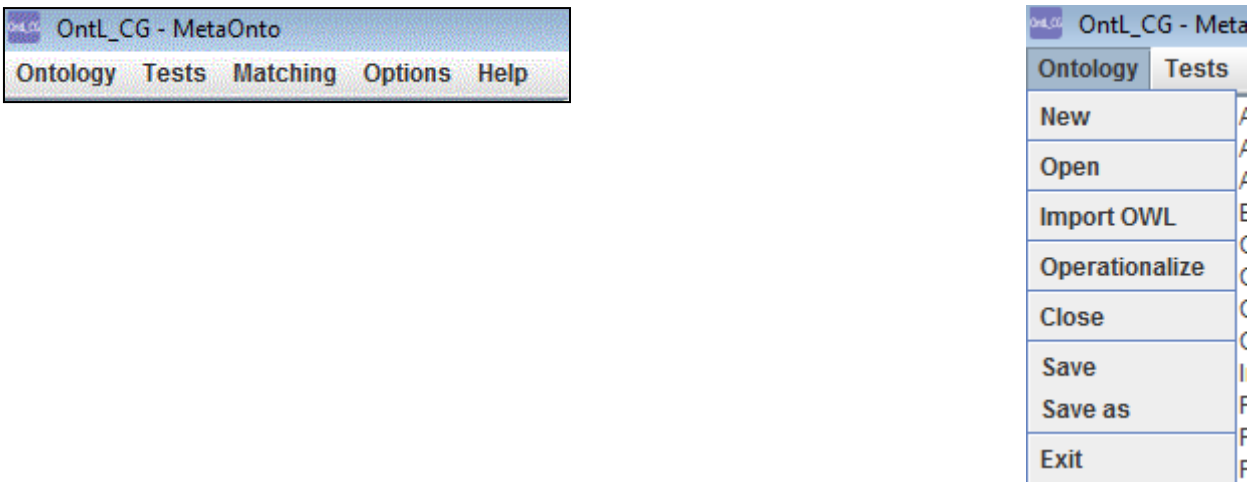


Figura AII - 1 Mesaje de conectare la serverul CoGITaNT

Meniul principal (Ontology, Tests, Matching, Options, Help) și meniurile secundare (figura AII - 2):



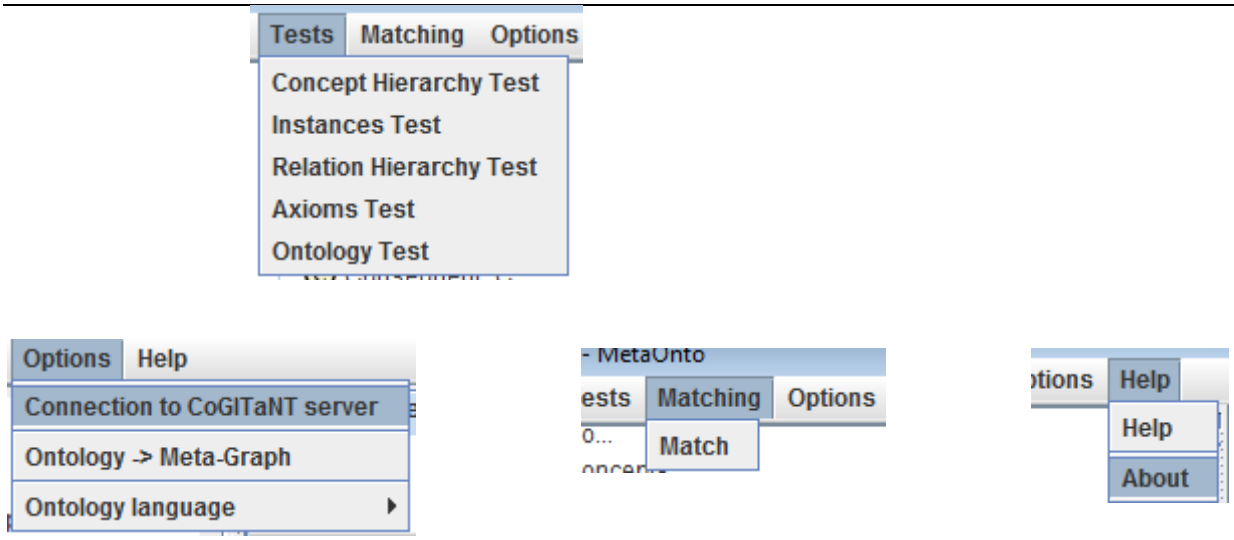


Figura AII - 2 Meniul principal și meniurile secundare

Interfața generală a aplicației (în stânga ontologia, în dreapta ierarhia tipurilor de concepte) este prezentată în figura AII - 3:

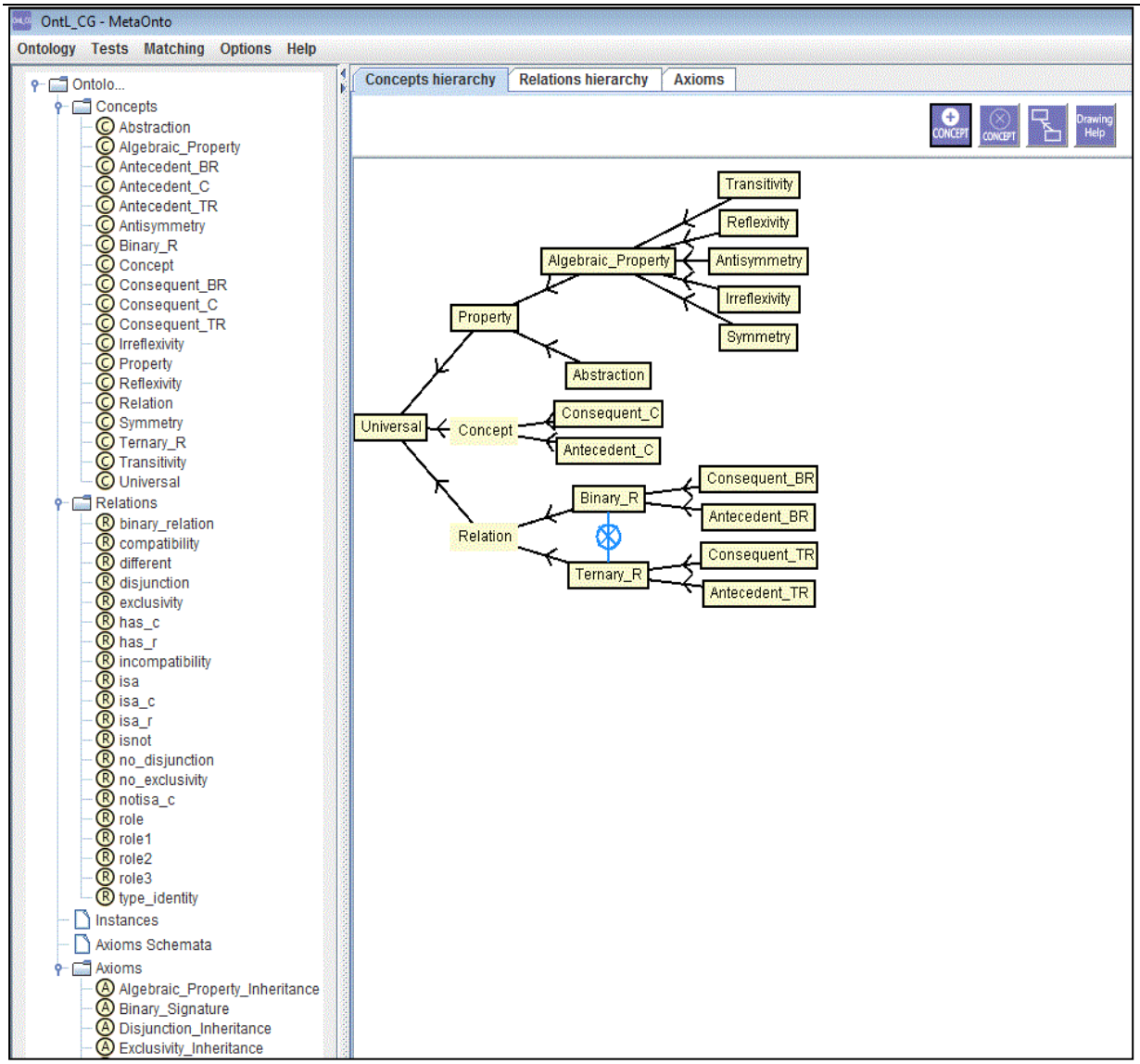
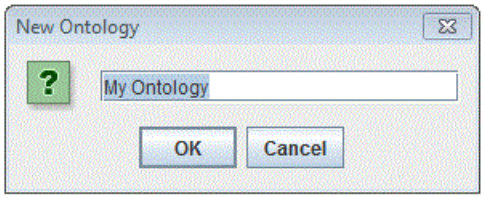
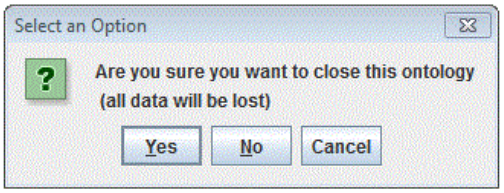


Figura AII - 3 OntL_CG – Interfața generală a aplicației

Crearea/deschiderea/salvarea/închiderea fișierului care conține ontologia (figura AII - 4):





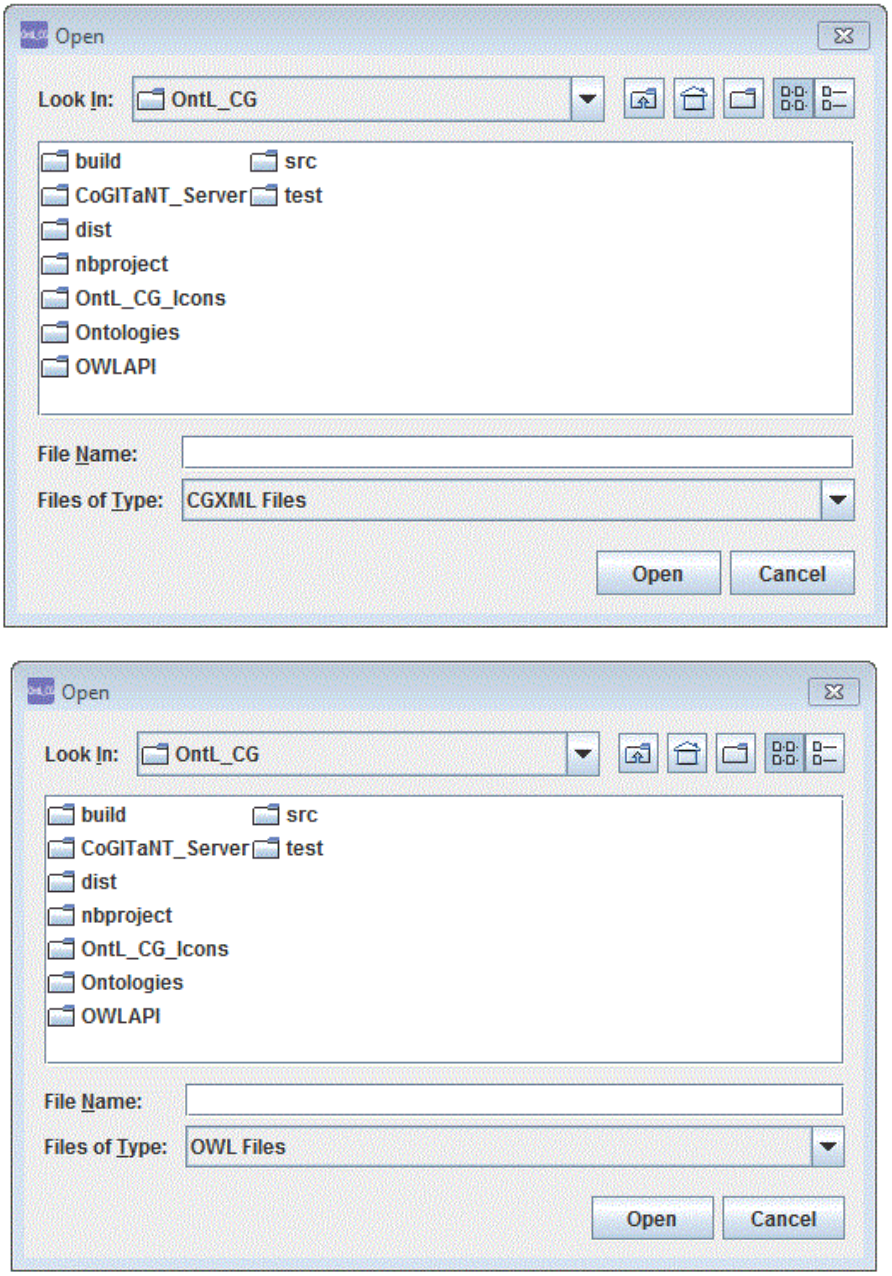
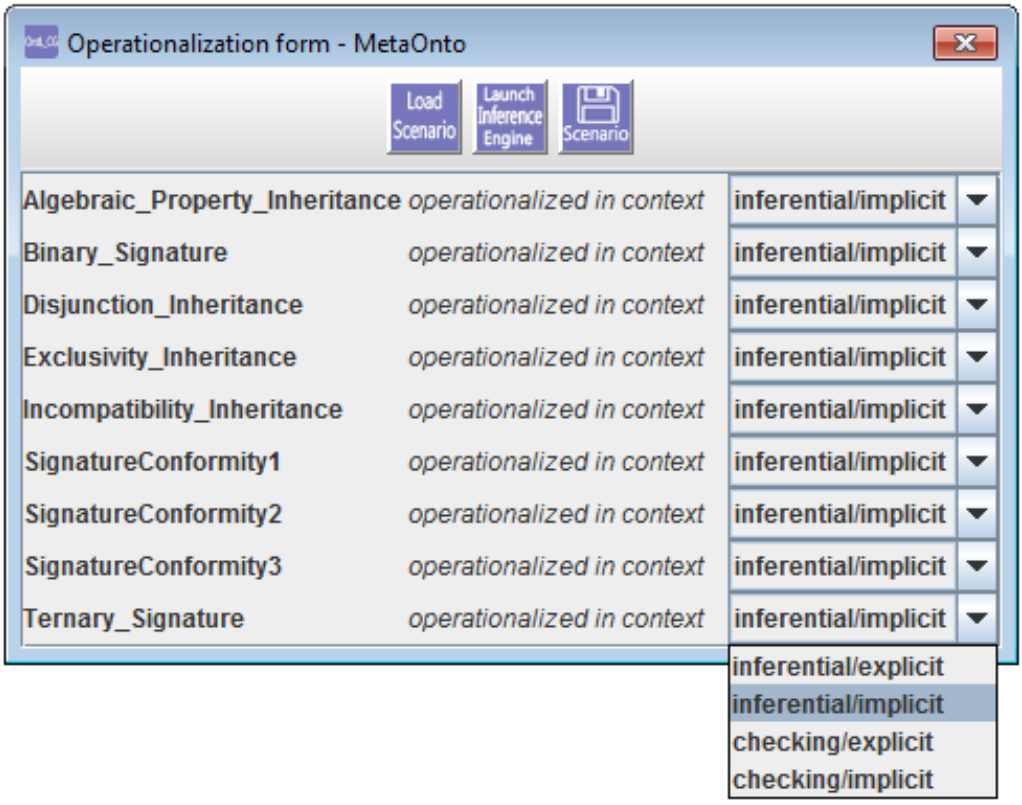
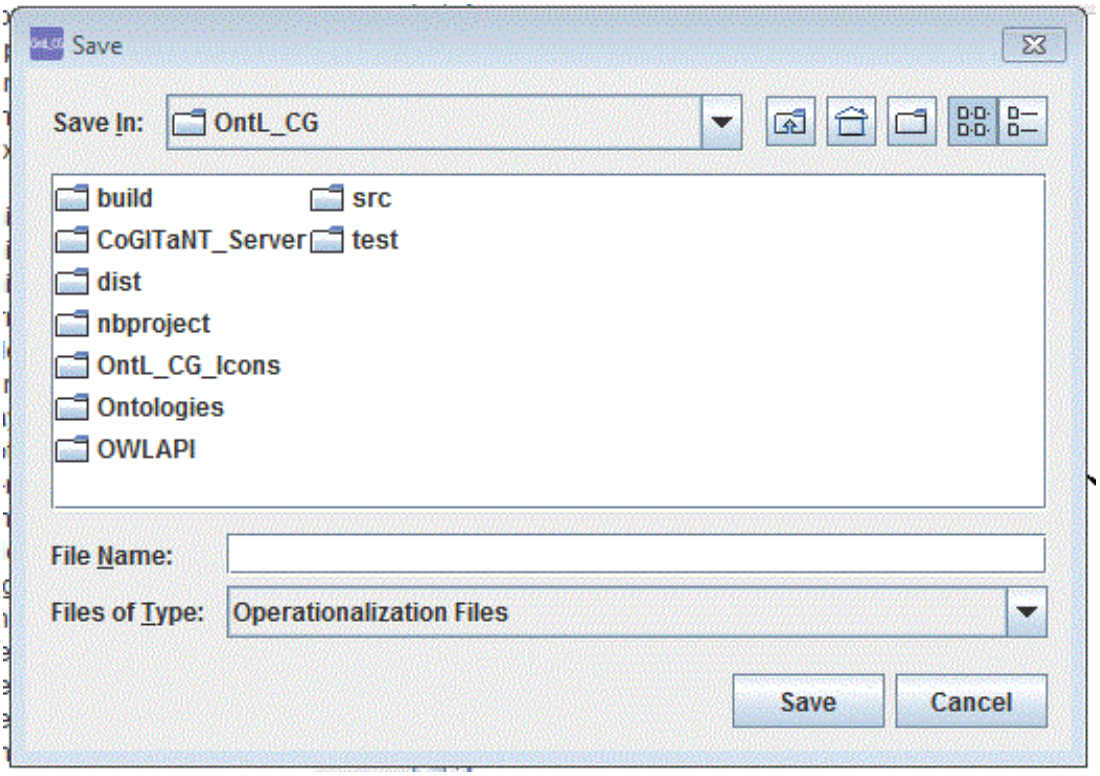


Figura AII - 4 Crearea/deschiderea/salvarea fișierului (.cgxml, .owl) cu ontologia

Specificarea și salvarea contextului de operaționalizare a ontologiei (figura AII - 5):


Operationalization form - MetaOnto
Load Scenario
Launch Inference Engine
Scenario
Algebraic_Property_Inheritance operationalized in context inferential/implicit
Binary_Signature operationalized in context inferential/implicit
Disjunction_Inheritance operationalized in context inferential/implicit
Exclusivity_Inheritance operationalized in context inferential/implicit
Incompatibility_Inheritance operationalized in context inferential/implicit
SignatureConformity1 operationalized in context inferential/implicit
SignatureConformity2 operationalized in context inferential/implicit
SignatureConformity3 operationalized in context inferential/implicit
Ternary_Signature operationalized in context inferential/implicit
inferential/explicit
inferential/implicit
checking/explicit
checking/implicit


Save
Save In: OntL_CG
build
src
CoGITaNT_Server
test
dist
nbproject
OntL_CG_Icons
Ontologies
OWLAPI
File Name:
Files of Type: Operationalization Files
Save
Cancel

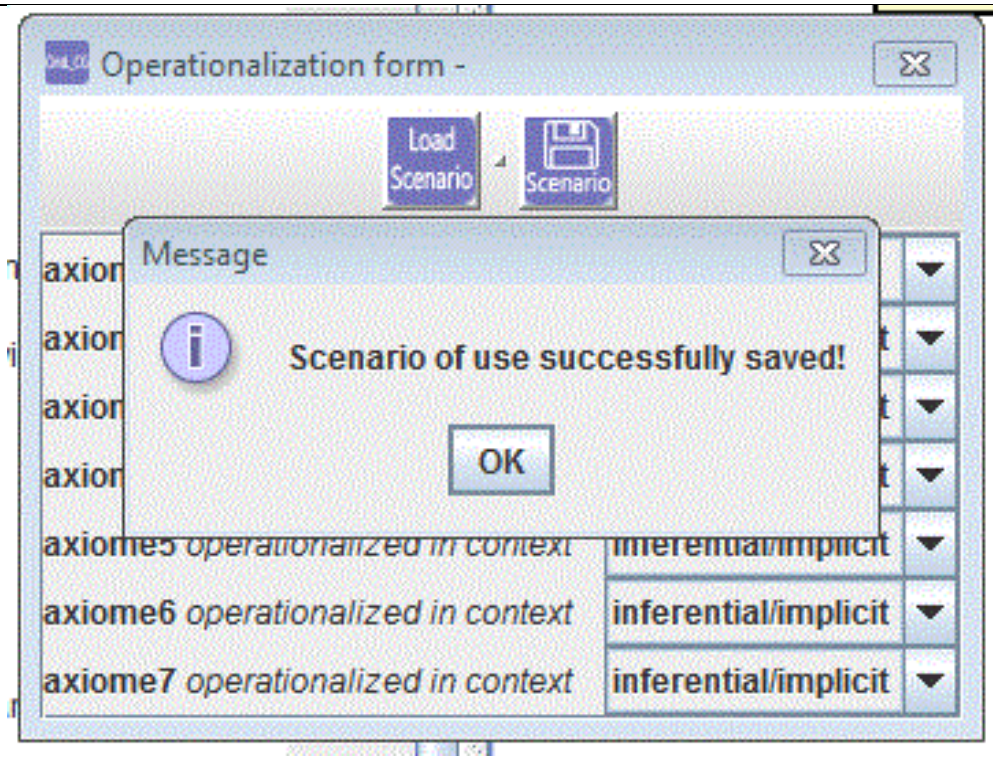


Figura AII - 5 Specificarea/salvarea contextului de operaționalizare a ontologiei

Încărcarea unui scenariu de operaționalizare existent (figura AII - 6):

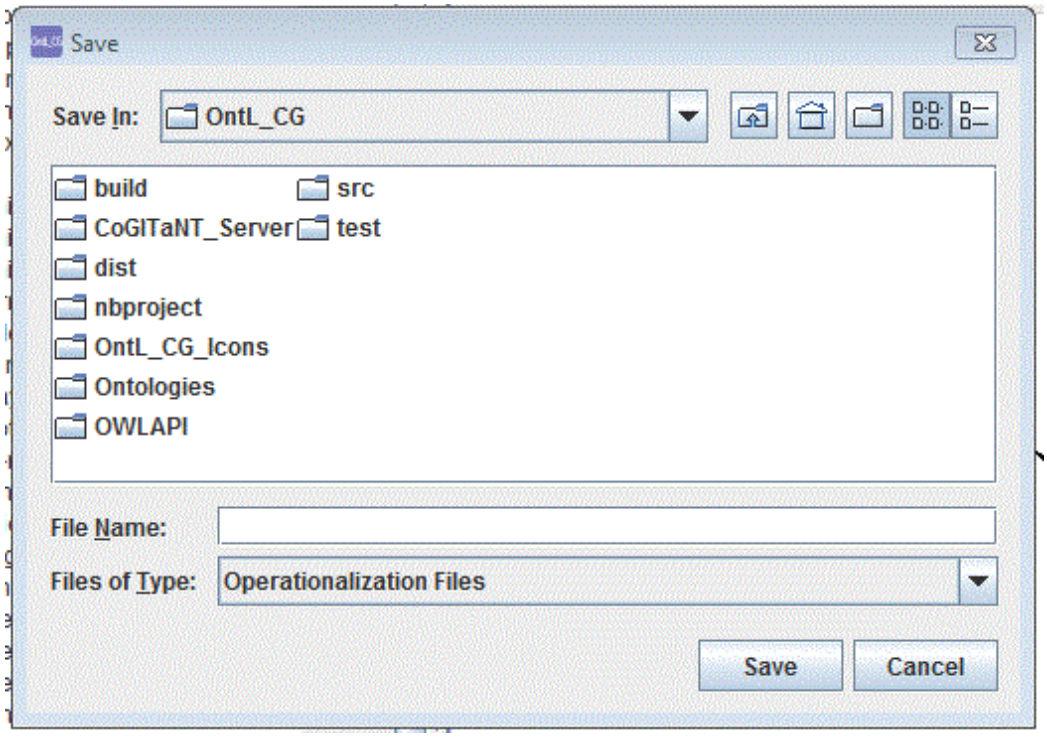


Figura AII - 6 Încărcarea unui scenariu de operaționalizare existent

Panoul din stânga (toate componentele ontologiei) (figura AII - 7):

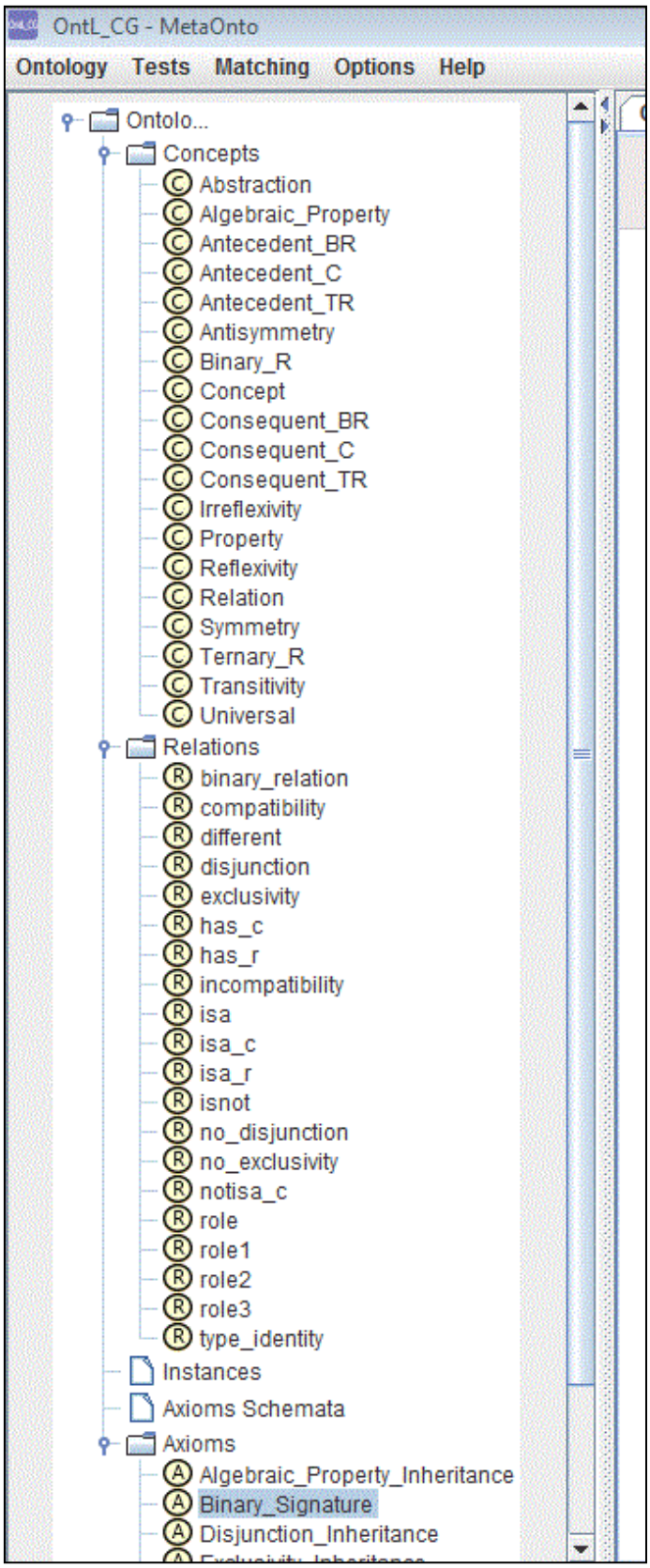


Figura AII - 7 Afișarea componentelor ontologiei în panoul din stânga

Vocabular în mai multe limbi (figura AII - 8):

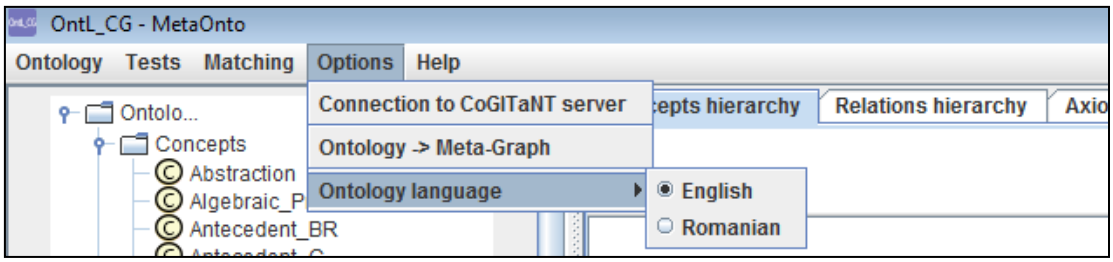


Figura AII - 8 Vocabular multi-lingual

Panoul din dreapta (ierarhie concepte/relații) (figura AII - 9, figura AII - 10):

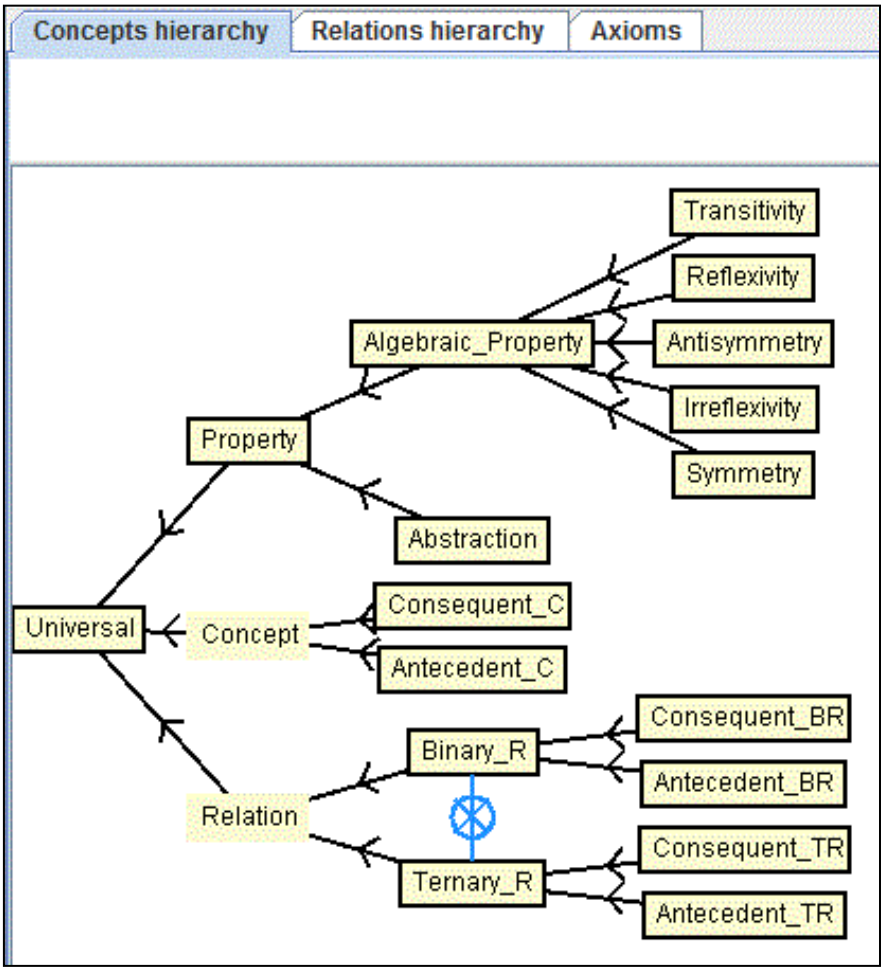


Figura AII - 9 Afișarea grafică a ierarhiei de concepte

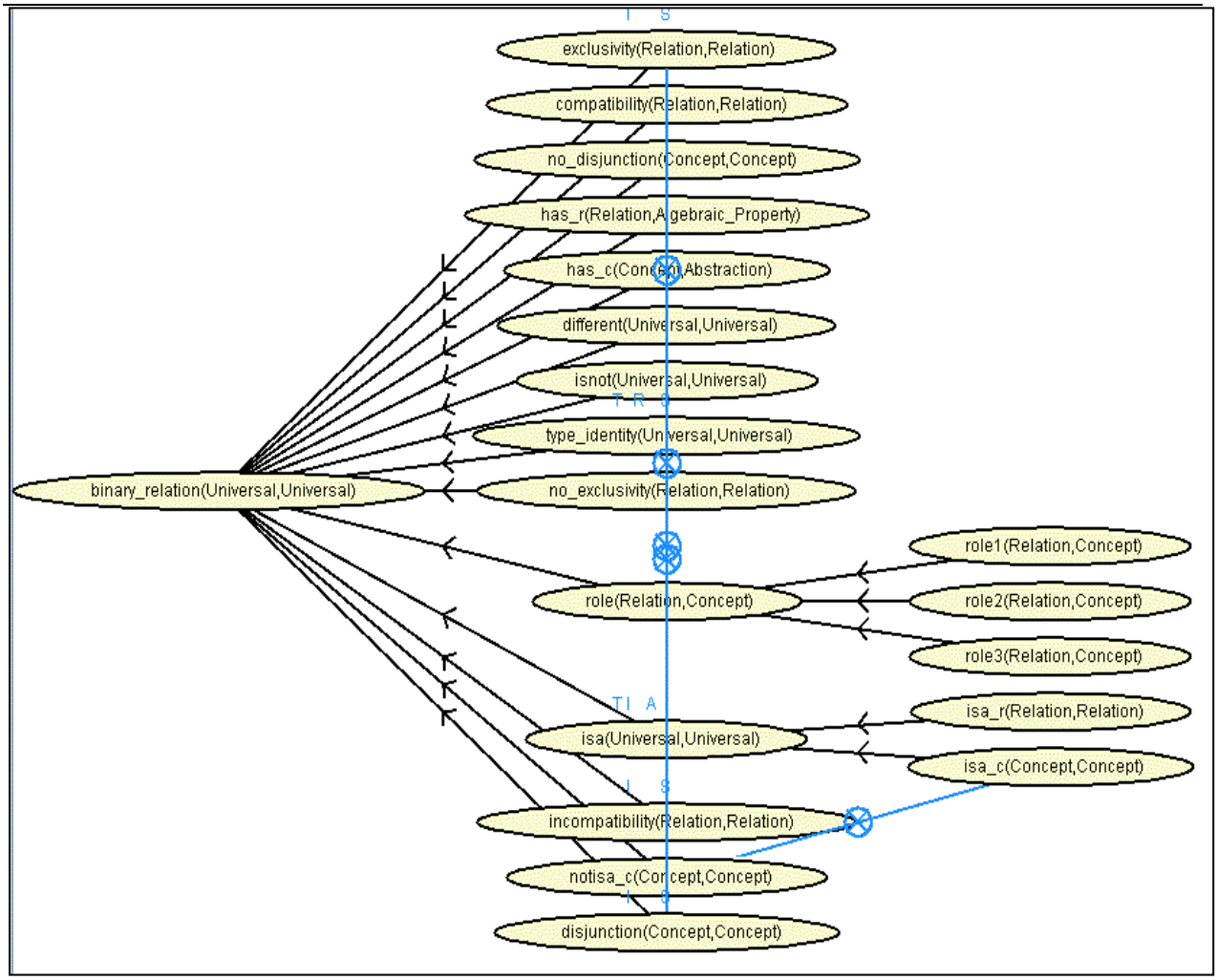


Figura AII - 10 Afișarea grafică a ierarhiei de concepte și de relații

Bare pentru instrumente de lucru pentru cu ierarhia de concepte/relații/axiome/graf (figura AII - 11):

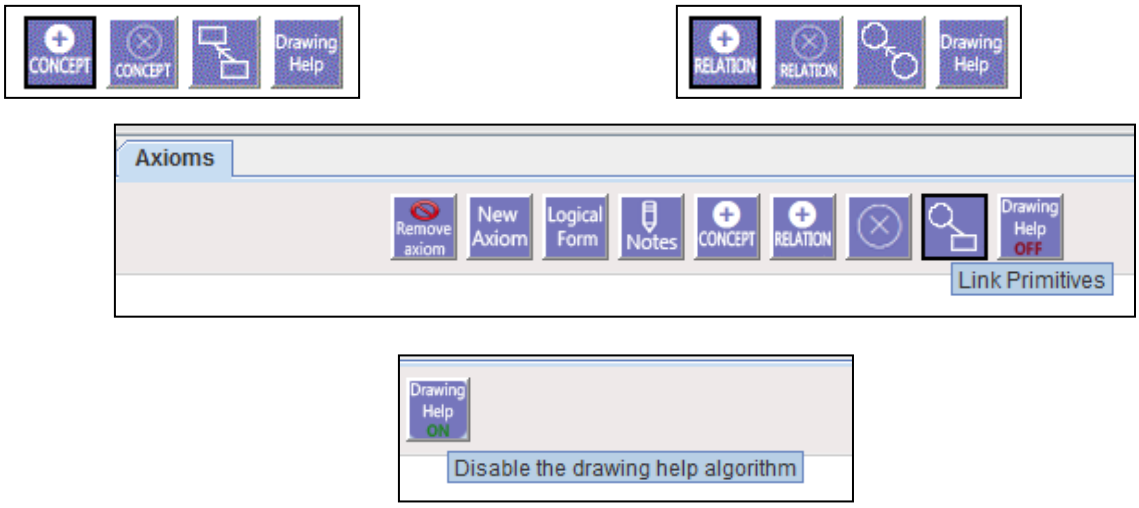


Figura AII - 11 Bare pentru instrumente de lucru

Specificarea proprietăților unui concept al domeniului de instruire (figura AII - 12)

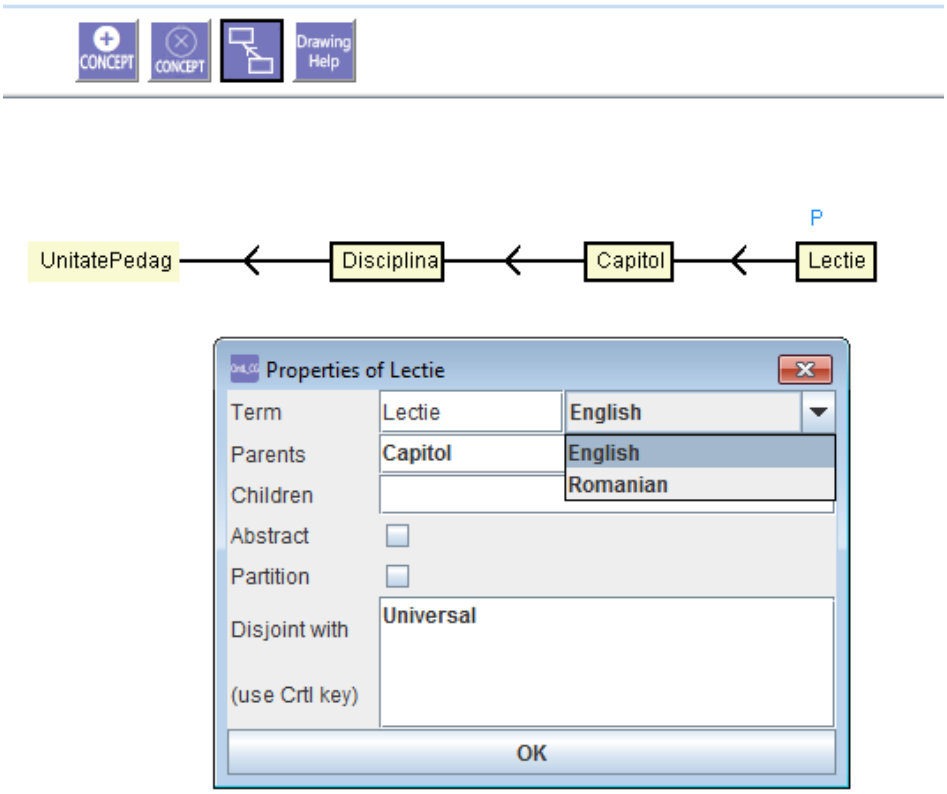
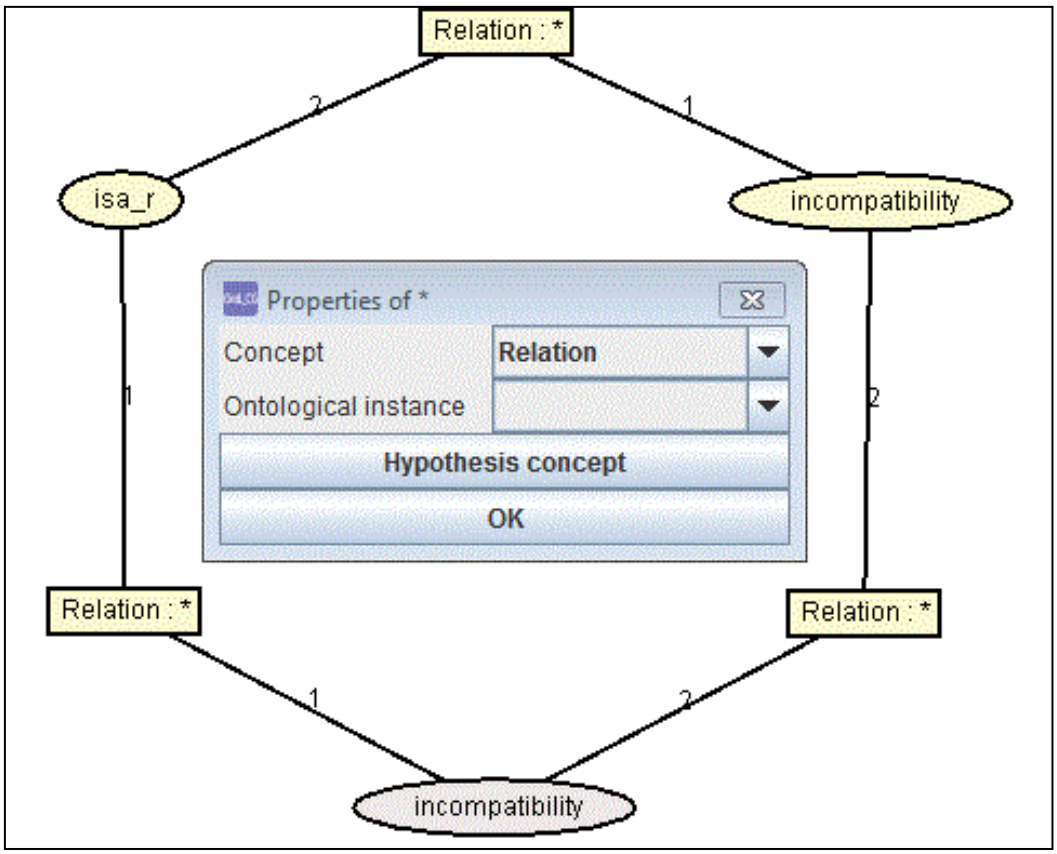


Figura AII - 12 Specificarea proprietăților unui concept din domeniul de instruire

Specificarea proprietăților unui concept din ipoteză/concluzie (figura AII - 13):



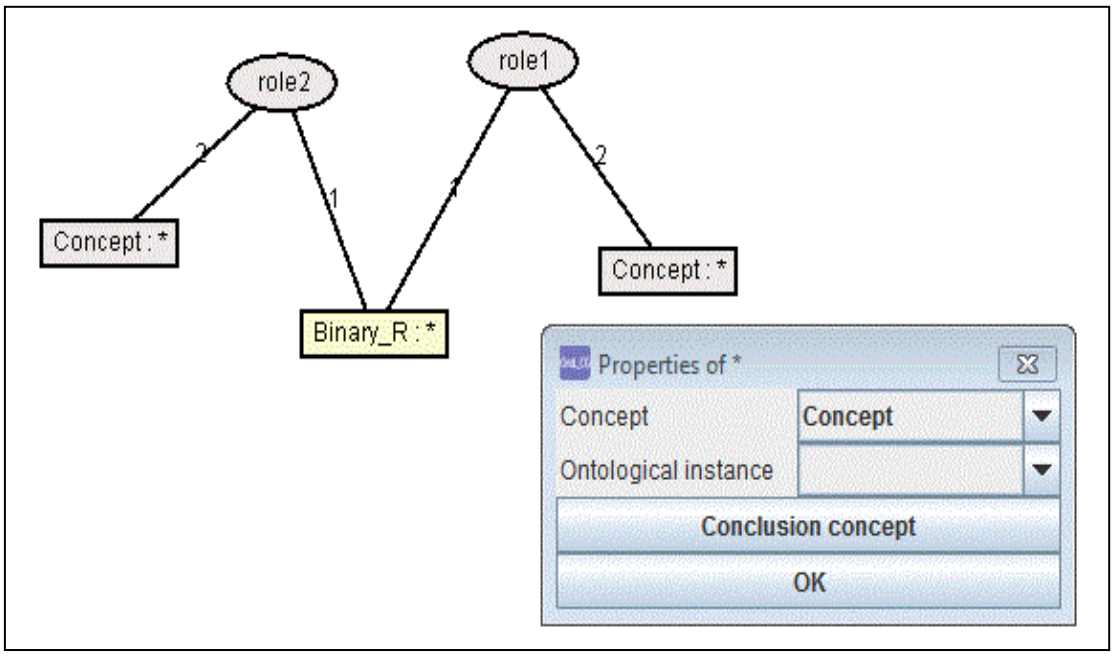
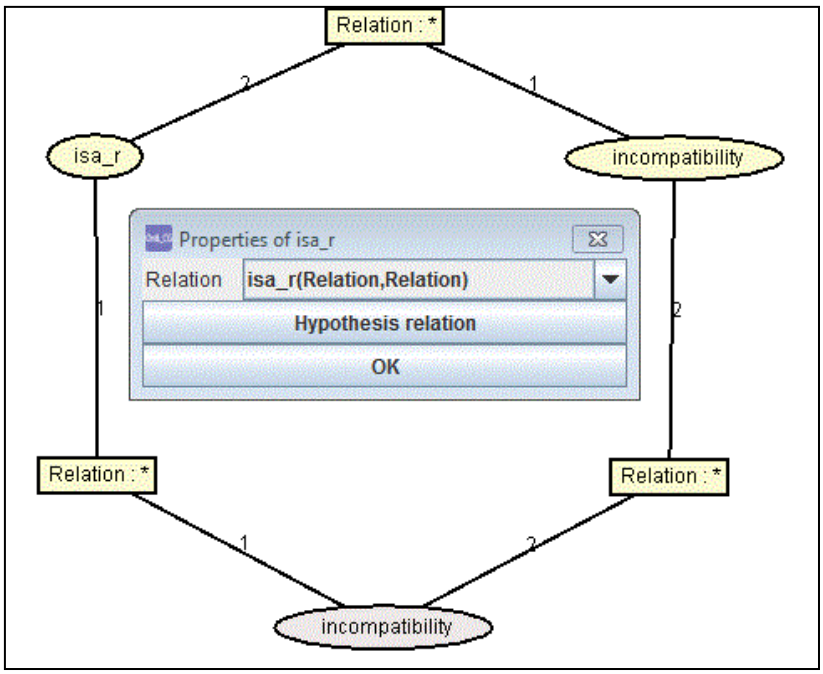


Figura AII - 13 Proprietățile unui concept din ipoteză/concluzie

Specificarea proprietăților unei relații ipoteză/concluzie (figura AII - 14):



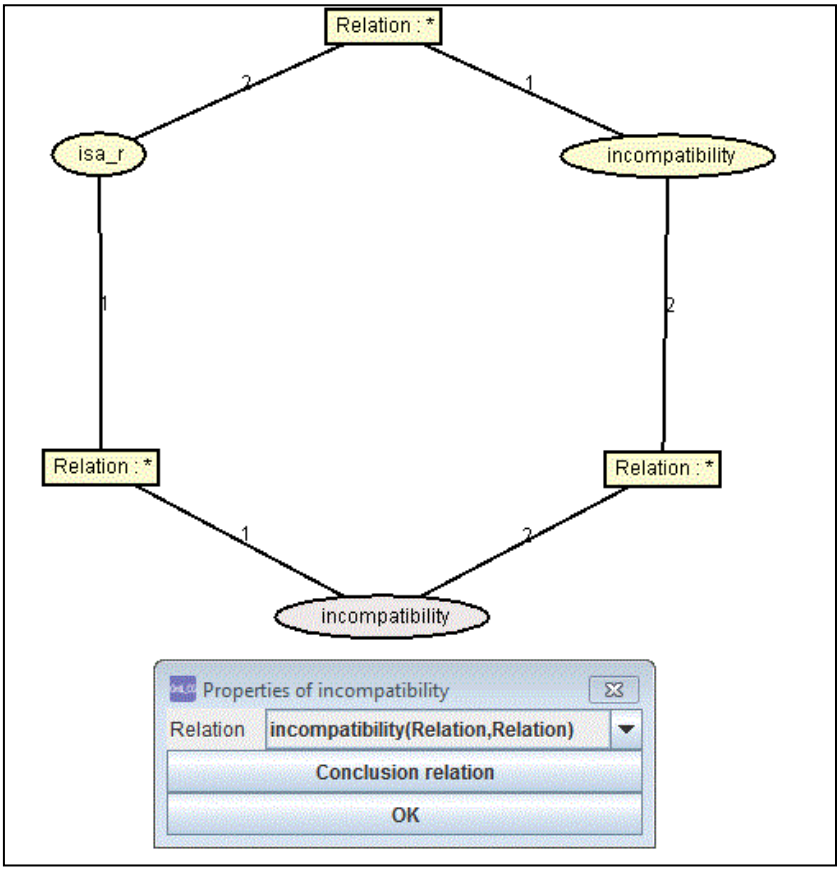


Figura AII - 14 Proprietățile unei relații ipoteză/concluzie

Specificarea proprietăților algebrice ale unei relații (figura AII - 15):

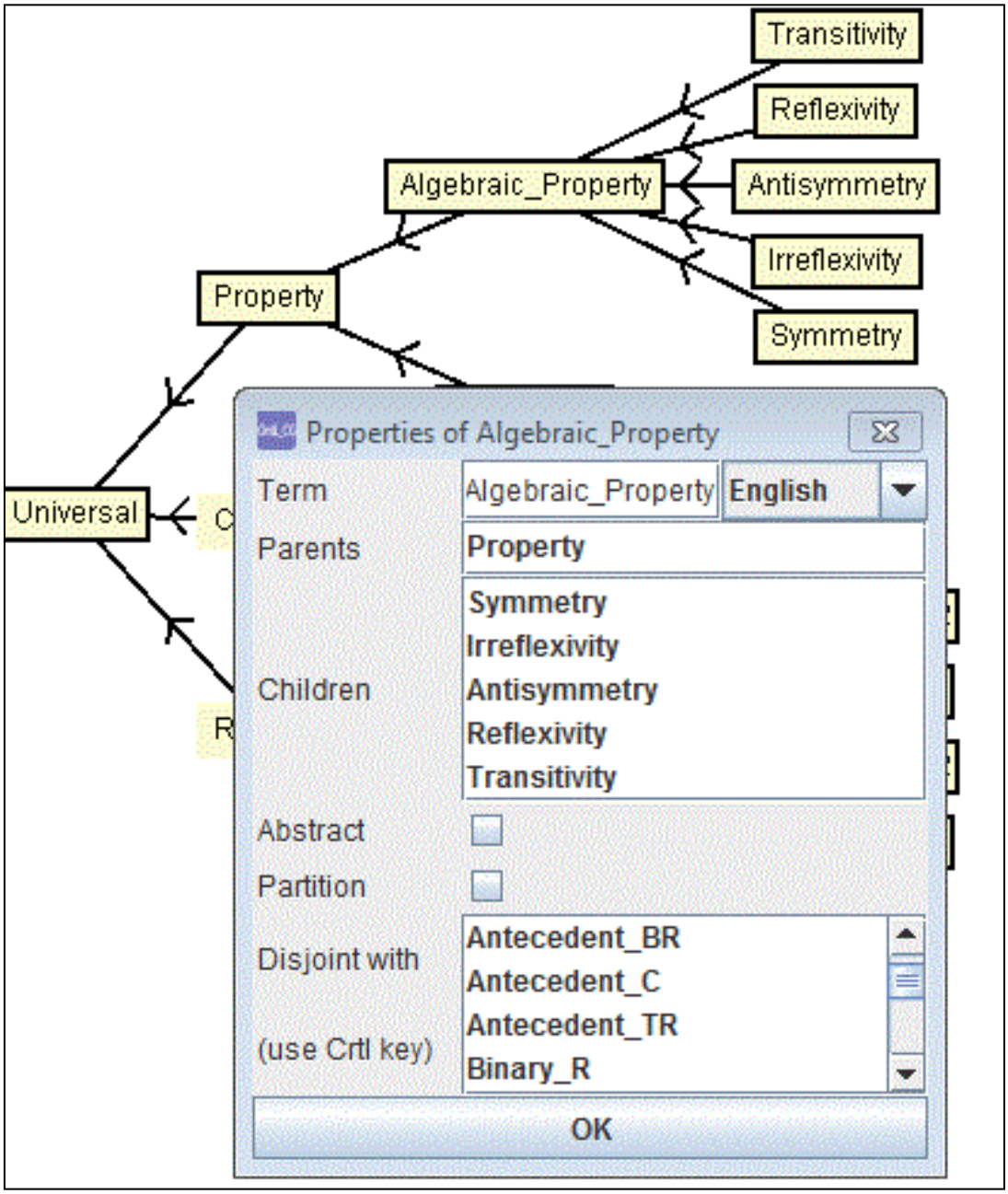
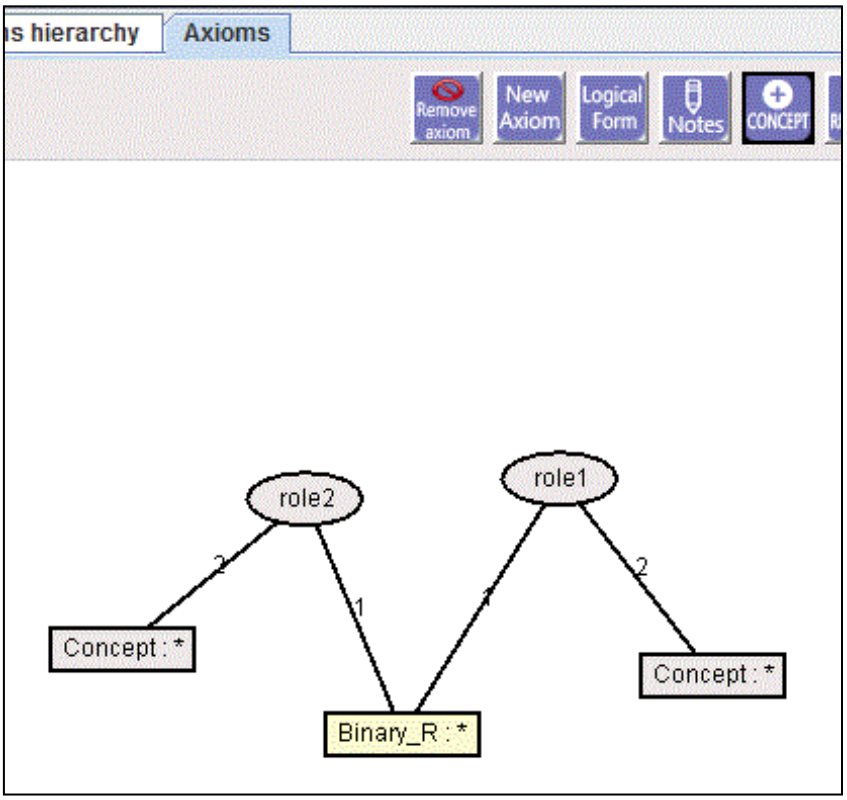


Figura AII - 15 Proprietățile algebrice ale unei relații

Specificarea proprietăților axiomelor:

Axiome pentru semnătura unei relații binare/ternare (figura AII - 16):



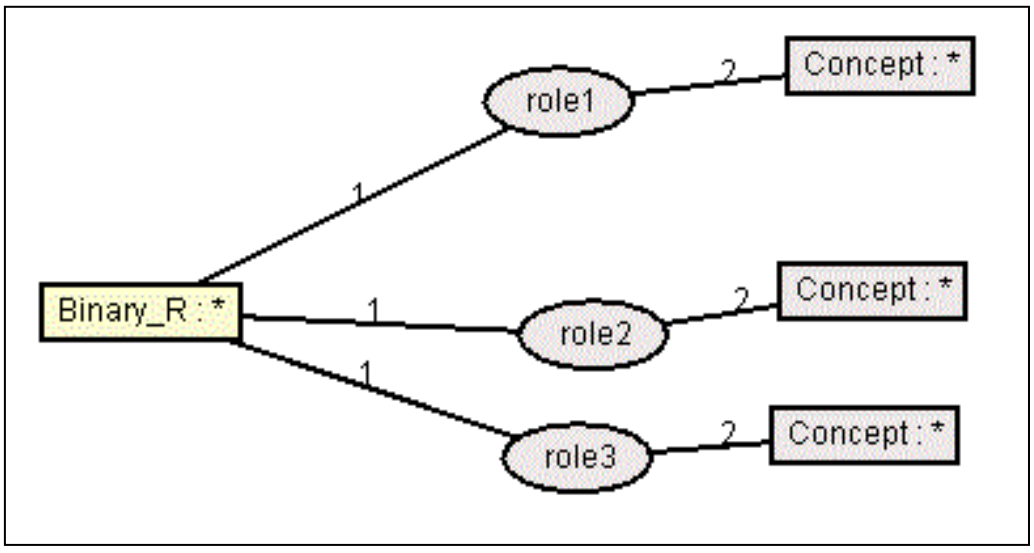
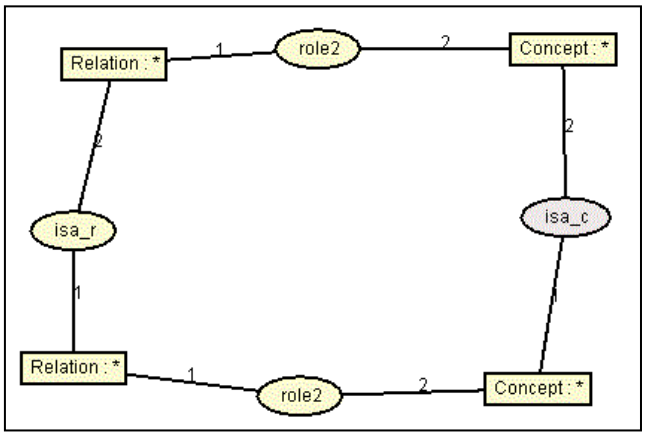
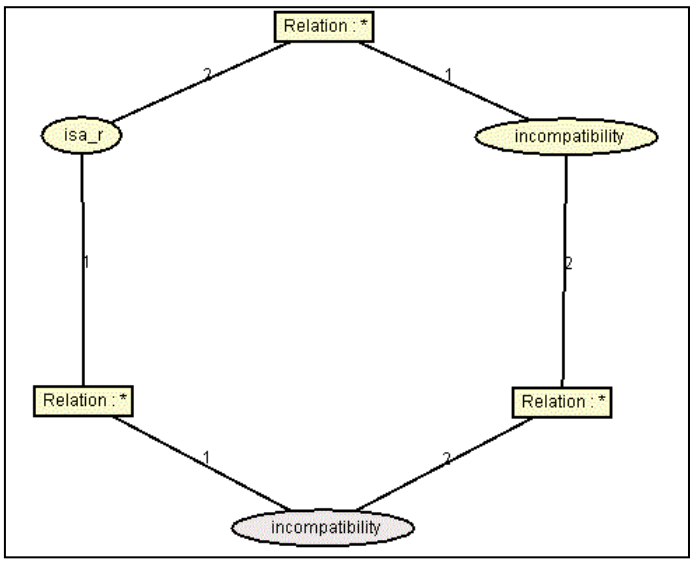


Figura AII - 16 Axiome pentru semnătura relației binare/ternare

Axiome pentru conformitatea semnăturii, moștenirea proprietăților algebrice și incompatibilitatea moștenirii (figura AII - 17):





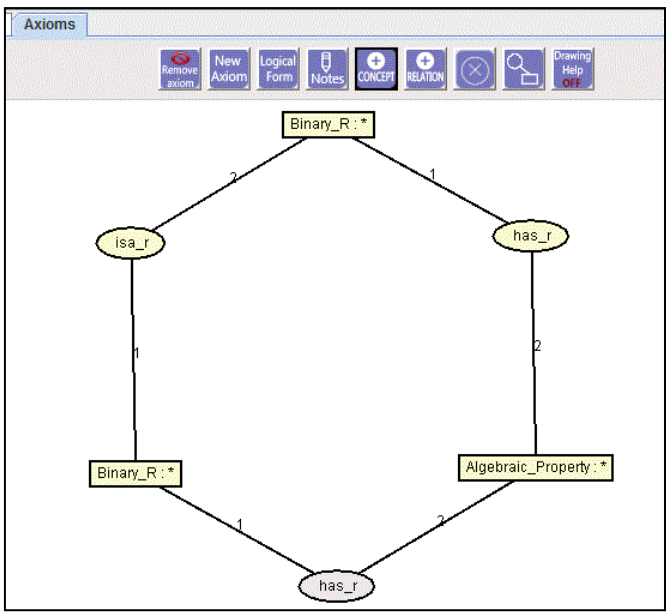
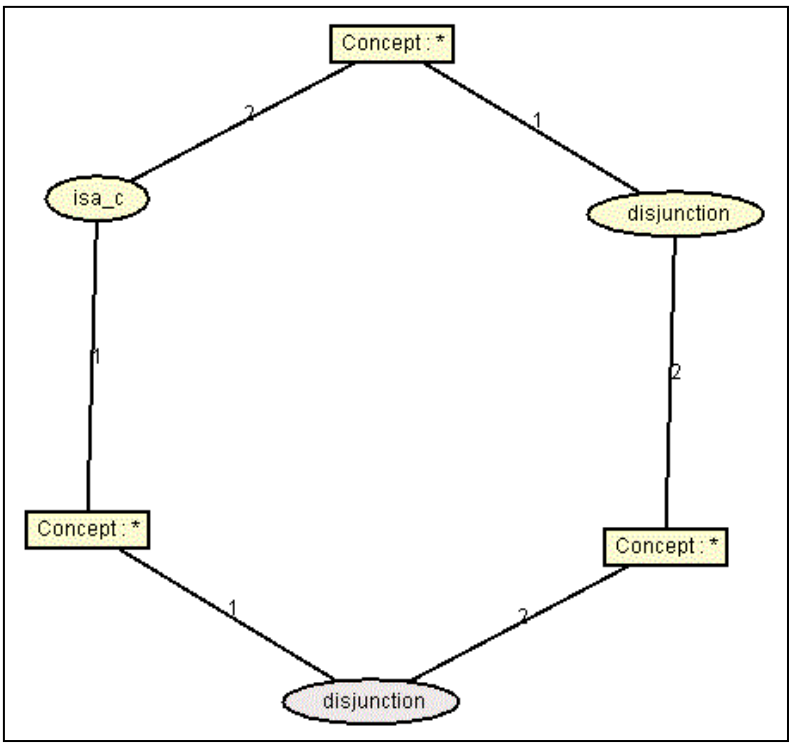
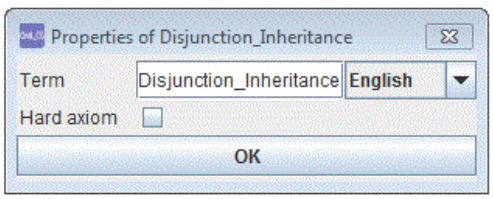


Figura AII - 17 Axiome pentru conformitatea semnăturii, moștenirea proprietăților algebrice și incompatibilitatea moștenirii

Axiome pentru moștenirea disjunctă și moștenirea exclusivă (figura AII - 18):





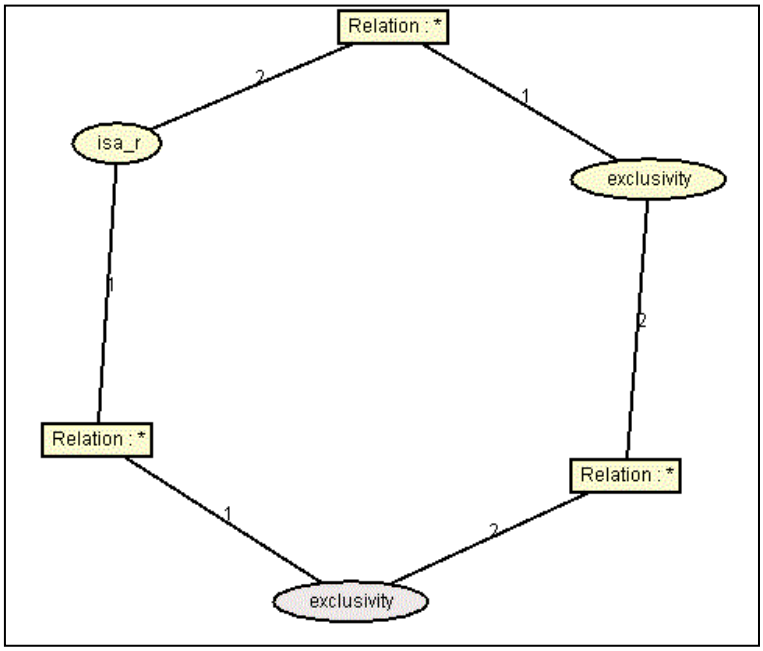


Figura AII - 18 Moștenirea disjunctă/exclusivă

Testarea ontologiei, cu opțiunea de afișare/suprimare a mesajelor de atenționare (figura AII - 19):

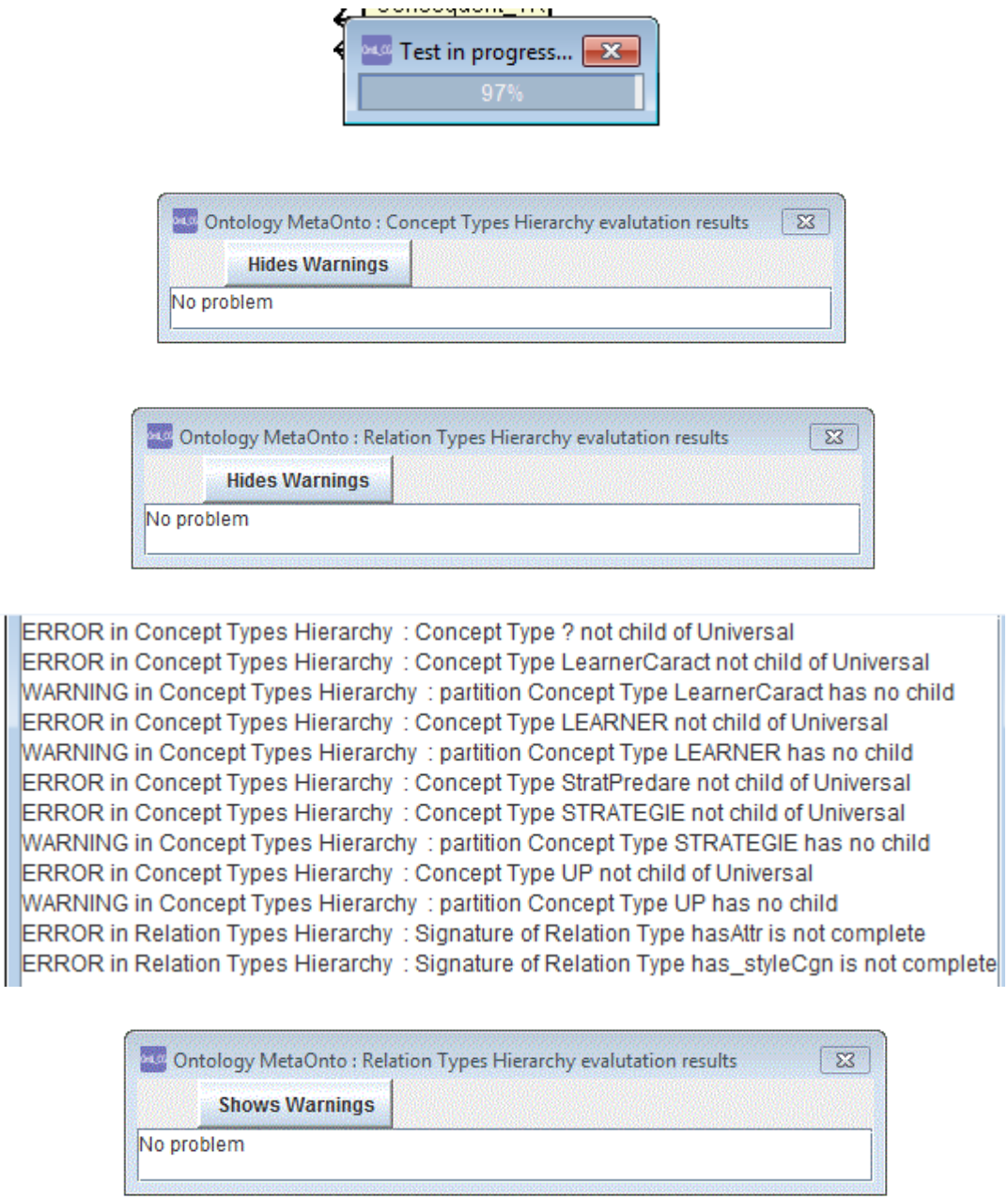
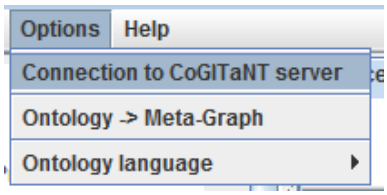


Figura AII - 19 Testarea ontologiei

Alte opțiuni (conexiunea la server, despre OntL_CG) (figura AII - 20):



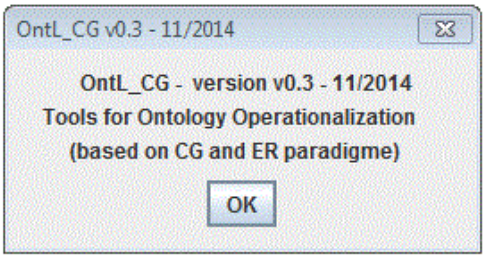


Figura AII - 20 Alte opțiuni

DOCUMENTAȚIA APLICAŢIEI OntL_CG

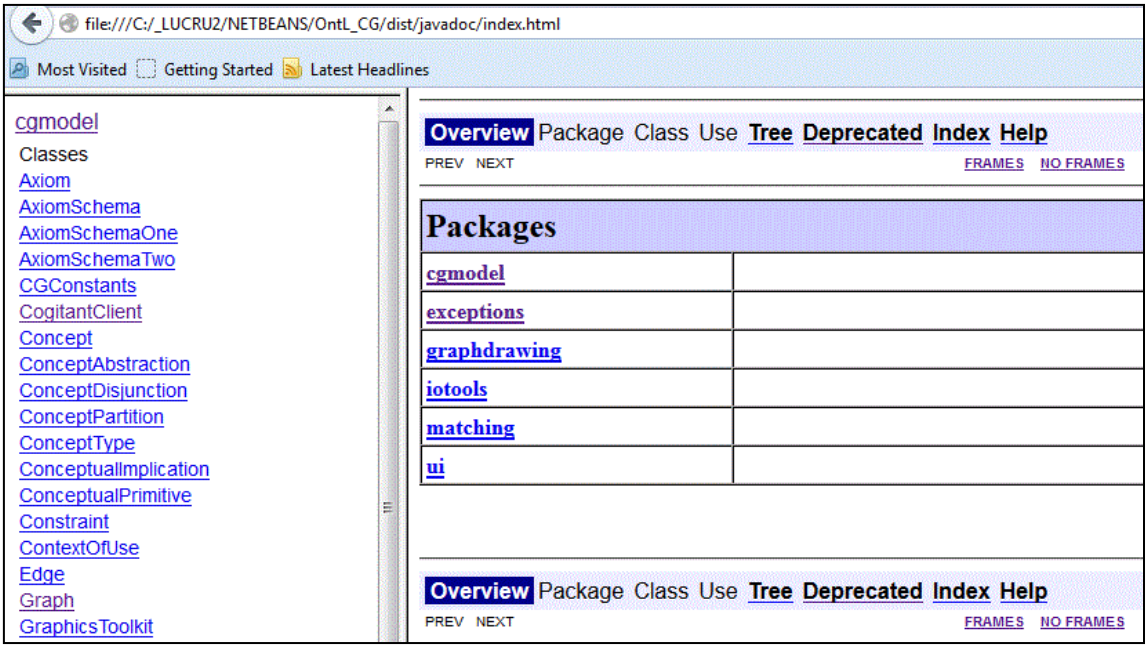


Figura AII - 21 Pachetele din OntL_CG

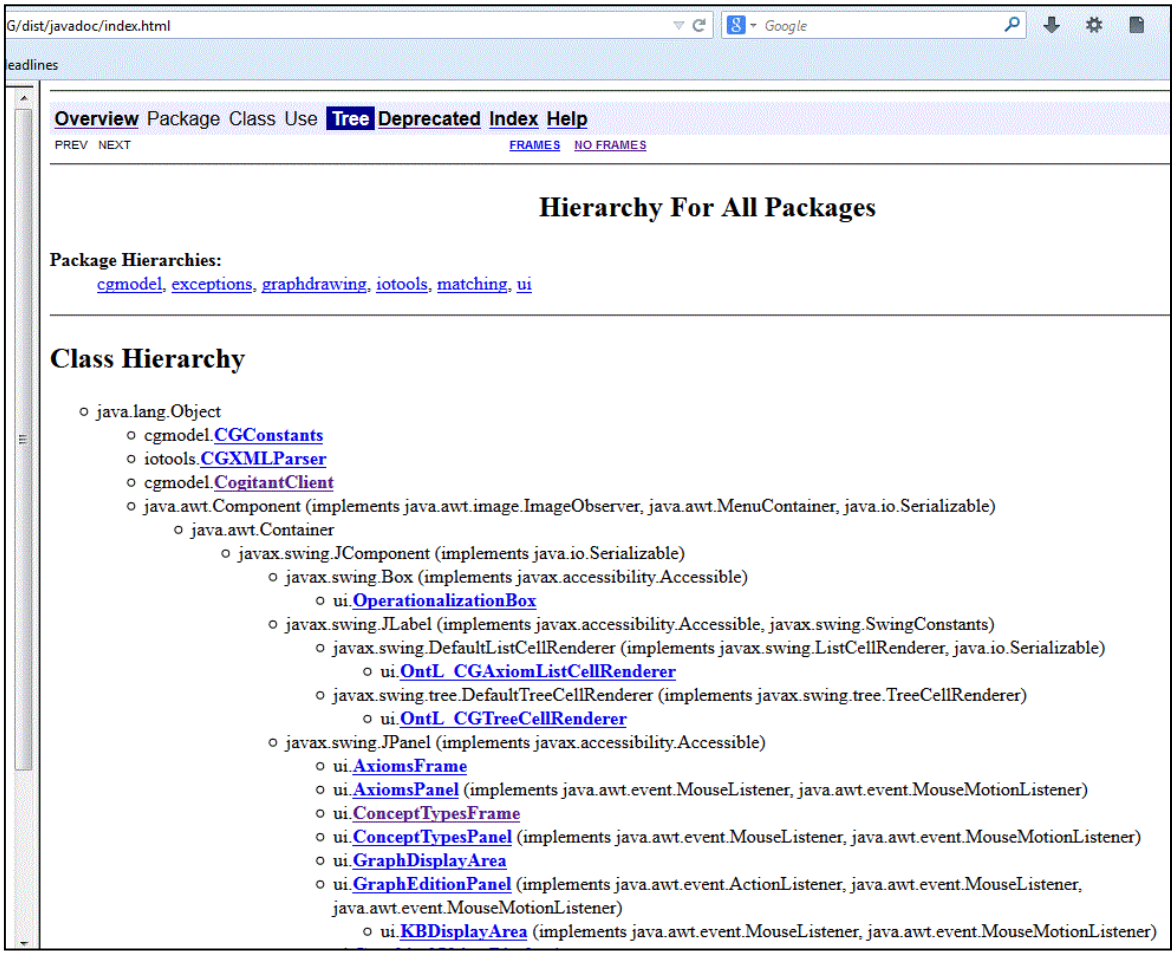
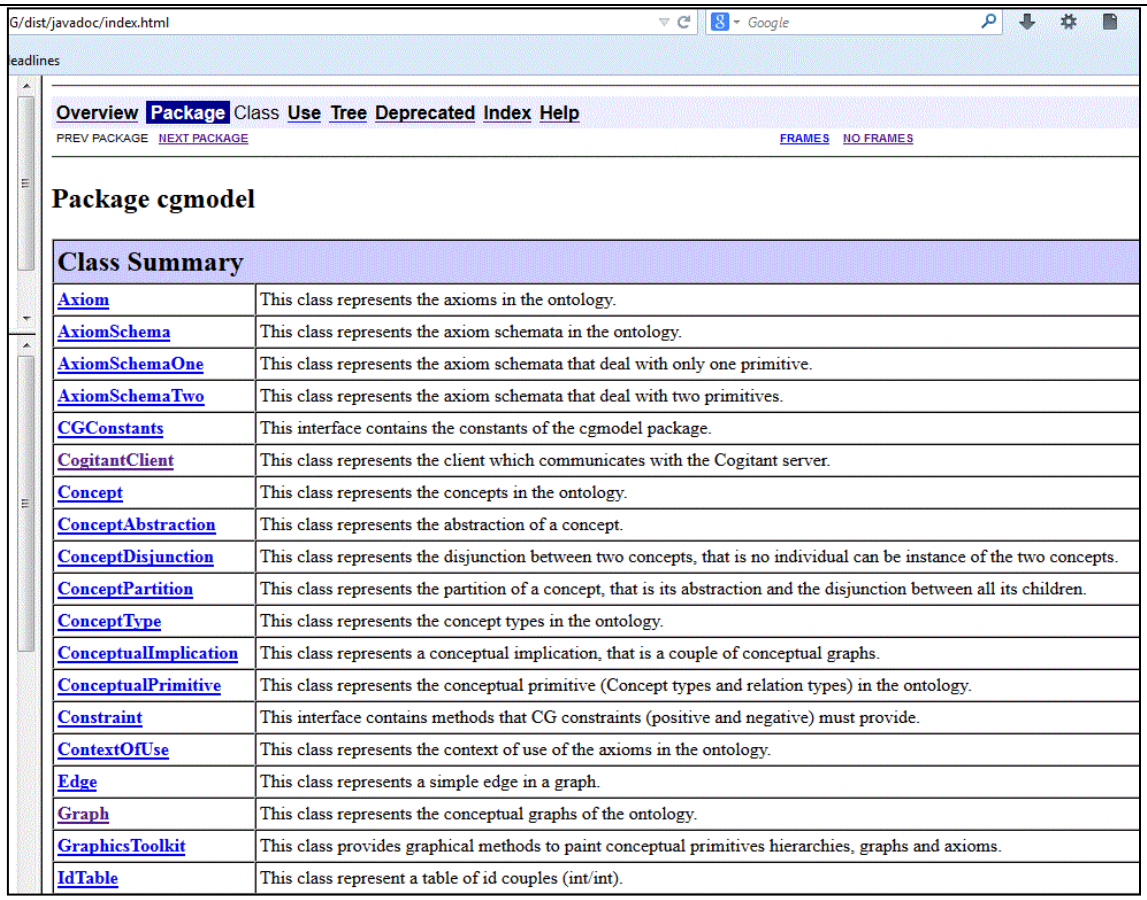


Figura AII - 22 Ierarhia claselor din pachete



G/dist/javadoc/index.html

eadlines

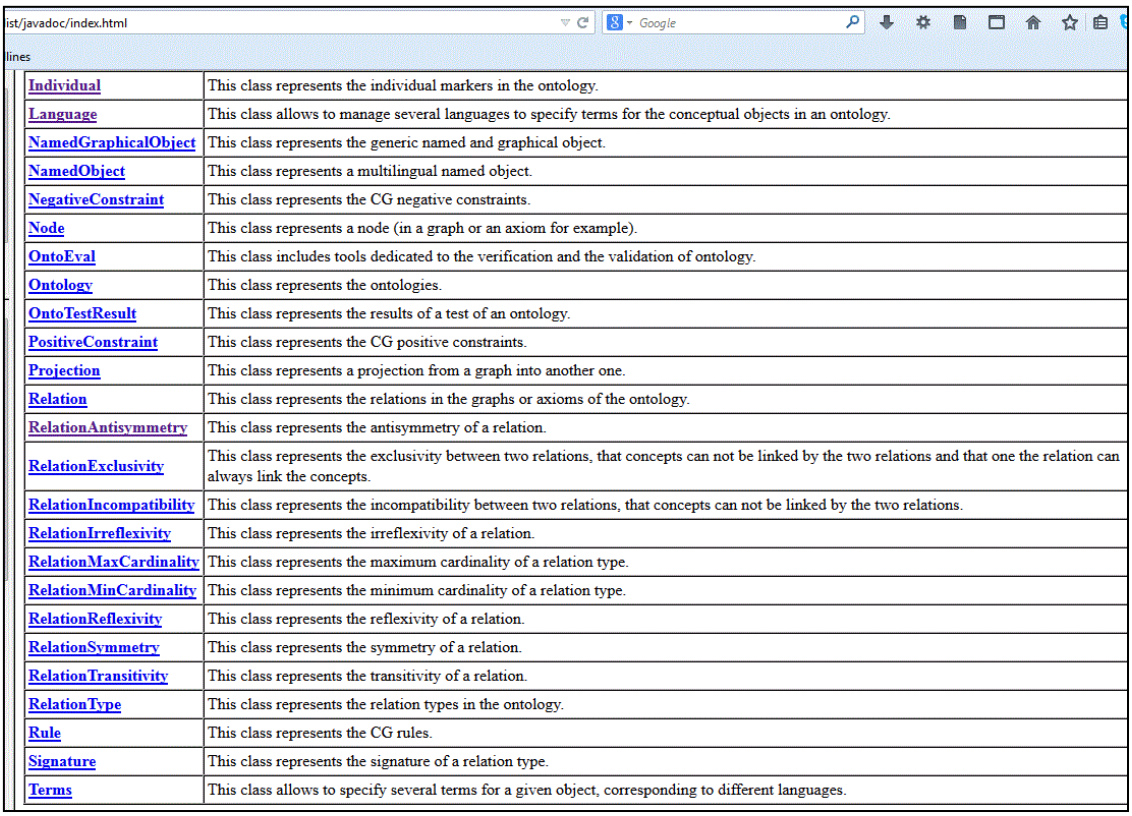
Overview Package Class Use Tree Deprecated Index Help

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

Package cgmodel

Class Summary	
Axiom	This class represents the axioms in the ontology.
AxiomSchema	This class represents the axiom schemata in the ontology.
AxiomSchemaOne	This class represents the axiom schemata that deal with only one primitive.
AxiomSchemaTwo	This class represents the axiom schemata that deal with two primitives.
CGConstants	This interface contains the constants of the cgmodel package.
CogitantClient	This class represents the client which communicates with the Cogitant server.
Concept	This class represents the concepts in the ontology.
ConceptAbstraction	This class represents the abstraction of a concept.
ConceptDisjunction	This class represents the disjunction between two concepts, that is no individual can be instance of the two concepts.
ConceptPartition	This class represents the partition of a concept, that is its abstraction and the disjunction between all its children.
ConceptType	This class represents the concept types in the ontology.
ConceptualImplication	This class represents a conceptual implication, that is a couple of conceptual graphs.
ConceptualPrimitive	This class represents the conceptual primitive (Concept types and relation types) in the ontology.
Constraint	This interface contains methods that CG constraints (positive and negative) must provide.
ContextOfUse	This class represents the context of use of the axioms in the ontology.
Edge	This class represents a simple edge in a graph.
Graph	This class represents the conceptual graphs of the ontology.
GraphicsToolkit	This class provides graphical methods to paint conceptual primitives hierarchies, graphs and axioms.
IdTable	This class represent a table of id couples (int/int).

Figura AII - 23 Pachetul cgmodel (1)



ist/javadoc/index.html

lines

Individual	This class represents the individual markers in the ontology.
Language	This class allows to manage several languages to specify terms for the conceptual objects in an ontology.
NamedGraphicalObject	This class represents the generic named and graphical object.
NamedObject	This class represents a multilingual named object.
NegativeConstraint	This class represents the CG negative constraints.
Node	This class represents a node (in a graph or an axiom for example).
OntoEval	This class includes tools dedicated to the verification and the validation of ontology.
Ontology	This class represents the ontologies.
OntoTestResult	This class represents the results of a test of an ontology.
PositiveConstraint	This class represents the CG positive constraints.
Projection	This class represents a projection from a graph into another one.
Relation	This class represents the relations in the graphs or axioms of the ontology.
RelationAntisymmetry	This class represents the antisymmetry of a relation.
RelationExclusivity	This class represents the exclusivity between two relations, that concepts can not be linked by the two relations and that one the relation can always link the concepts.
RelationIncompatibility	This class represents the incompatibility between two relations, that concepts can not be linked by the two relations.
RelationIrreflexivity	This class represents the irreflexivity of a relation.
RelationMaxCardinality	This class represents the maximum cardinality of a relation type.
RelationMinCardinality	This class represents the minimum cardinality of a relation type.
RelationReflexivity	This class represents the reflexivity of a relation.
RelationSymmetry	This class represents the symmetry of a relation.
RelationTransitivity	This class represents the transitivity of a relation.
RelationType	This class represents the relation types in the ontology.
Rule	This class represents the CG rules.
Signature	This class represents the signature of a relation type.
Terms	This class allows to specify several terms for a given object, corresponding to different languages.

Figura AII - 24 Pachetul cgmodel (2)

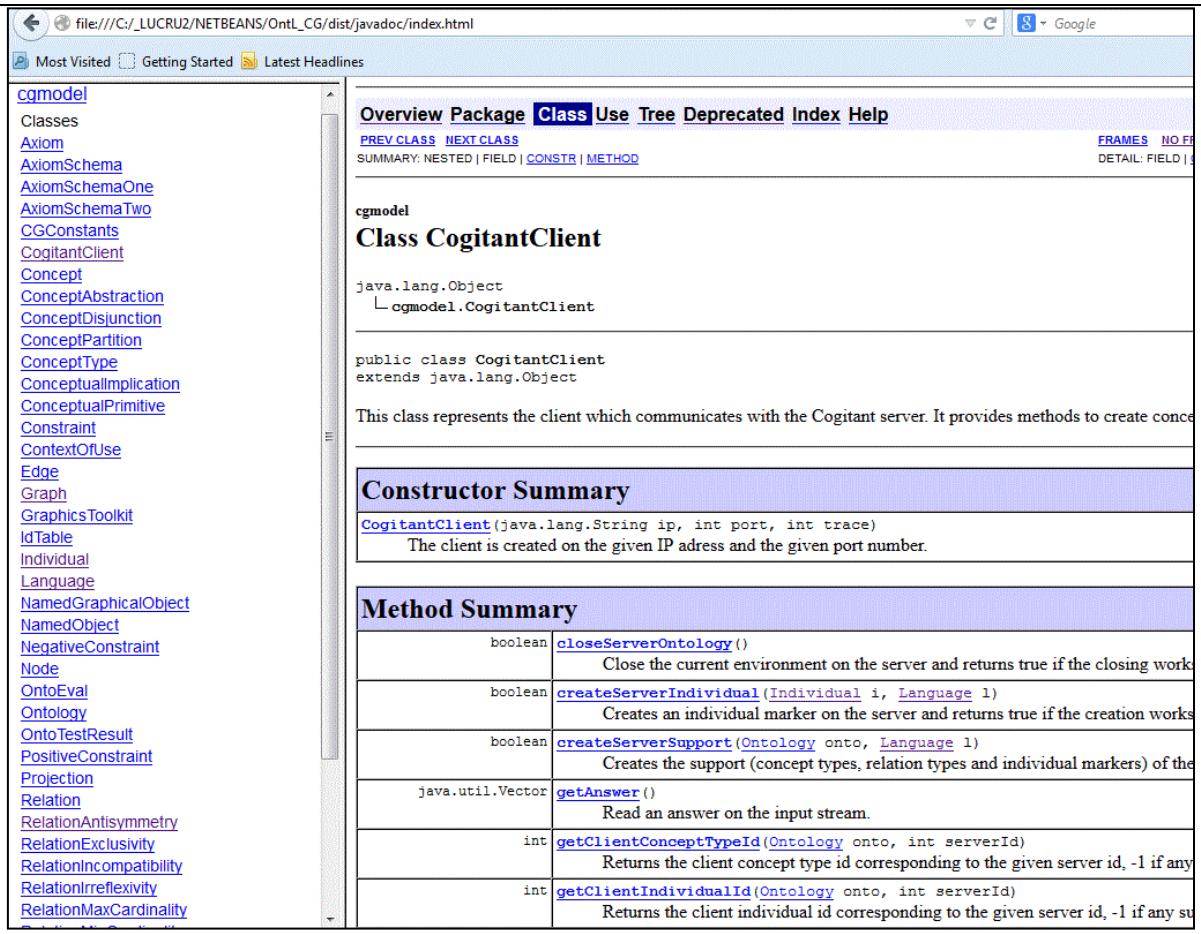


Figura AII - 25 Documentarea clasei CogitantClient din pachetul cgmodel

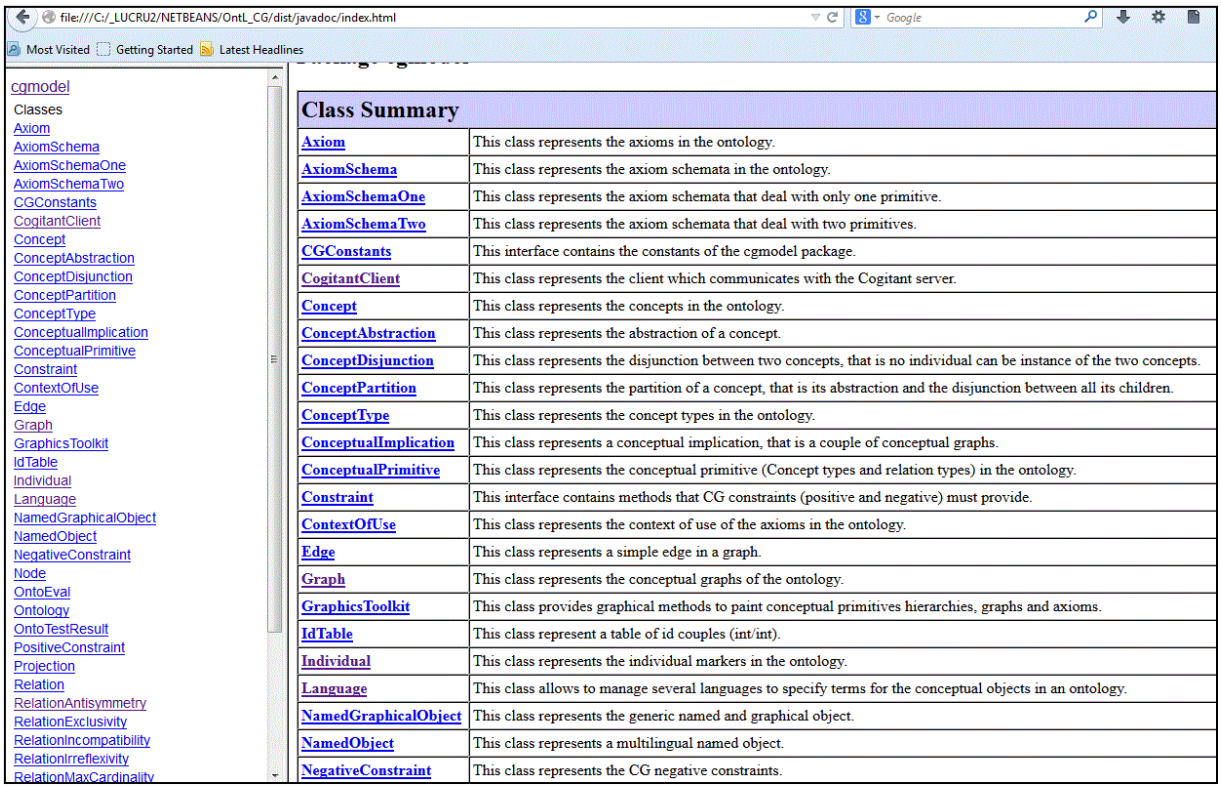


Figura AII - 26 Clasele din pachetul cgmodel și descrierea lor

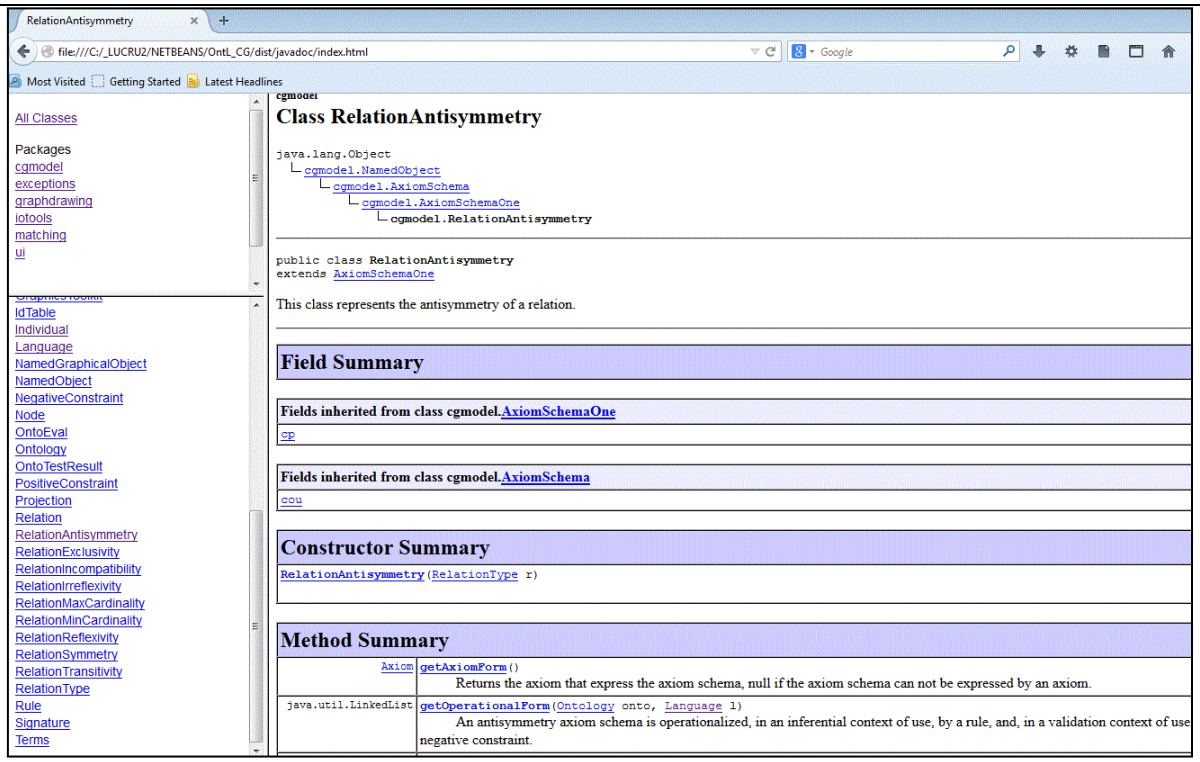


Figura AII - 27 Documentarea clasei RelationAntisymetry din pachetul cgmodel

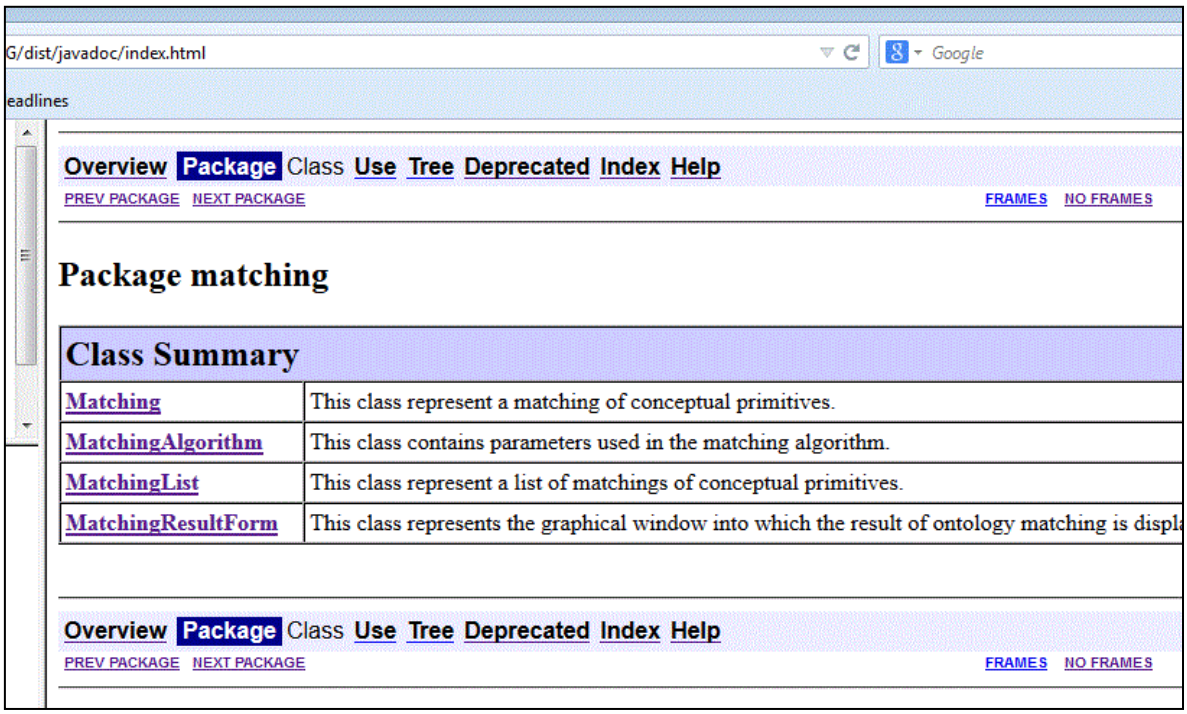


Figura AII - 28 Pachetul matching

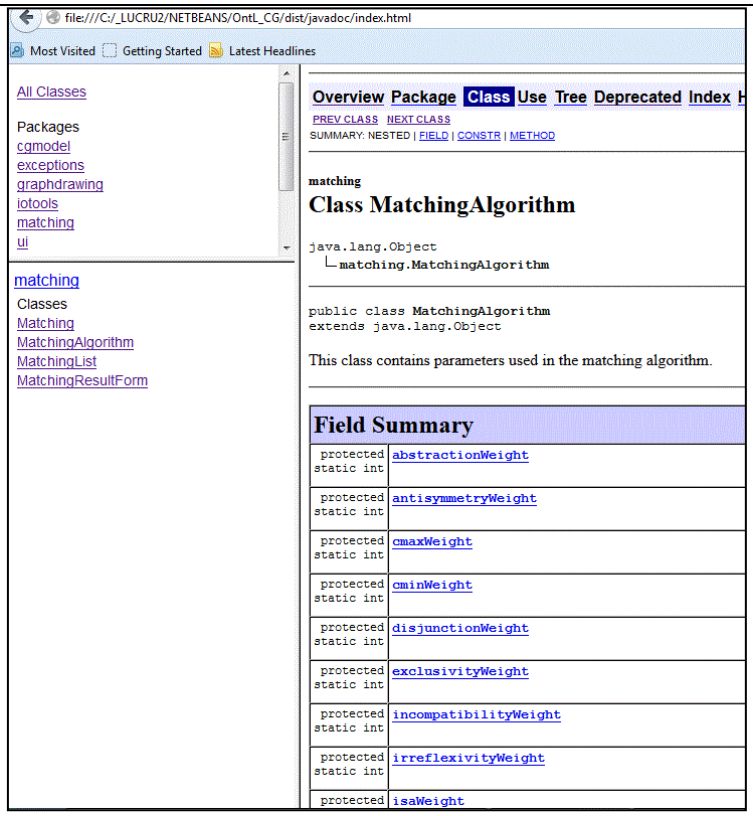


Figura AII - 29 Clasa MatchingAlgorithm din pachetul matching

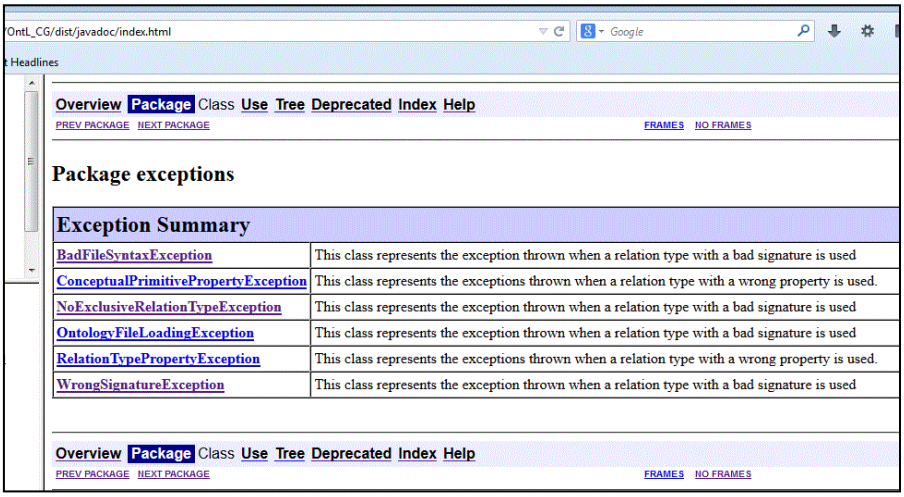
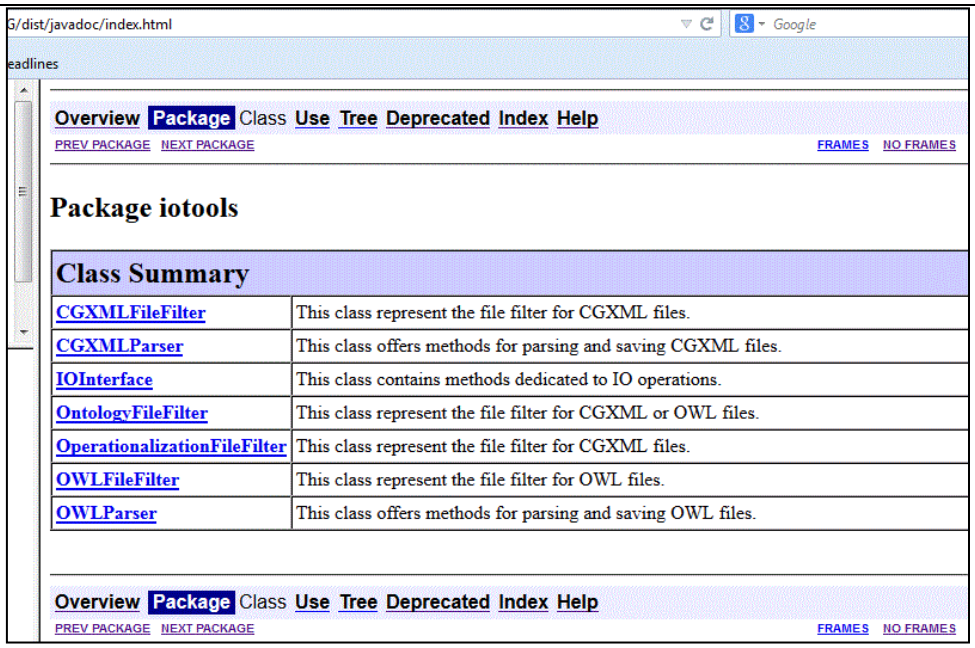


Figura AII - 30 Pachetul exceptions


G/dist/javadoc/index.html

Google

eadlines

Overview Package Class Use Tree Deprecated Index Help

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

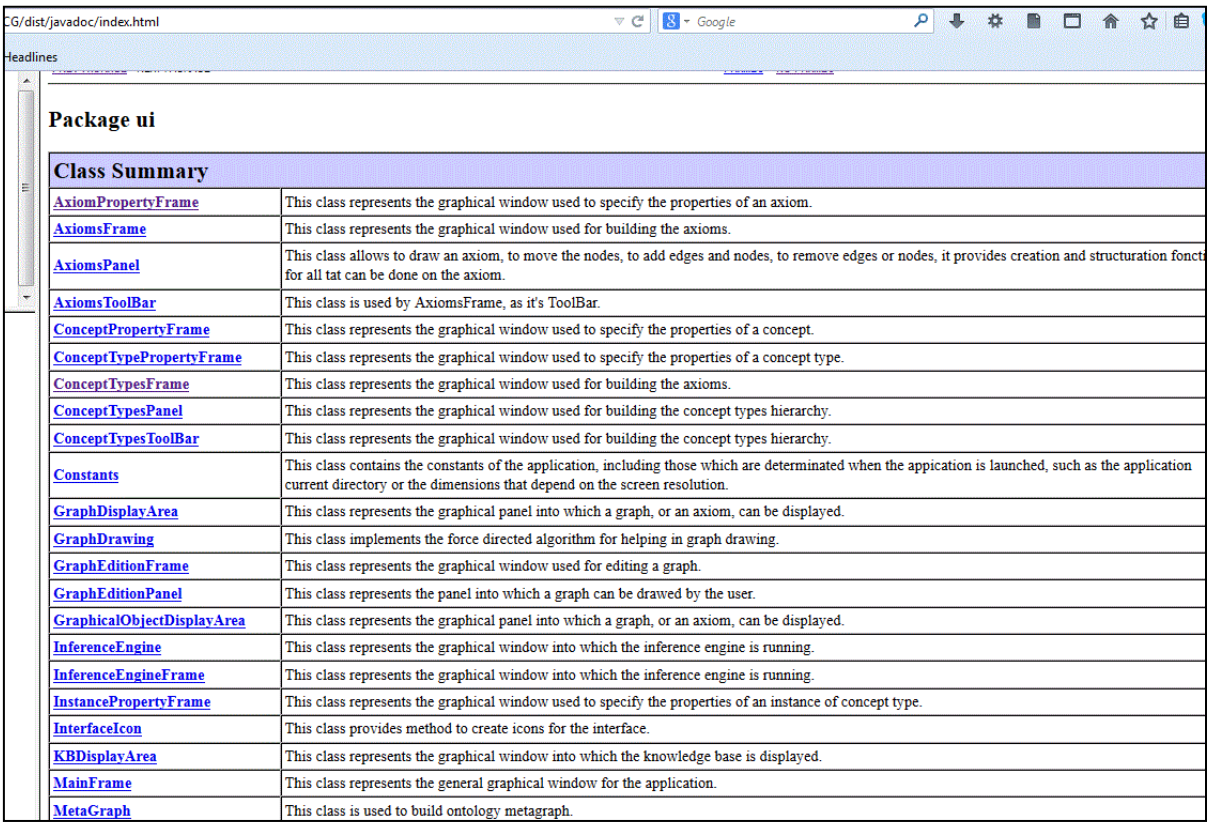
Package iotools

Class Summary	
CGXMLFileFilter	This class represent the file filter for CGXML files.
CGXMLParser	This class offers methods for parsing and saving CGXML files.
IOInterface	This class contains methods dedicated to IO operations.
OntologyFileFilter	This class represent the file filter for CGXML or OWL files.
OperationalizationFileFilter	This class represent the file filter for CGXML files.
OWLFileFilter	This class represent the file filter for OWL files.
OWLParser	This class offers methods for parsing and saving OWL files.

Overview Package Class Use Tree Deprecated Index Help

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

Figura AII - 31 Pachetul iotools


CG/dist/javadoc/index.html

Google

Headlines

Package ui

Class Summary	
AxiomPropertyFrame	This class represents the graphical window used to specify the properties of an axiom.
AxiomsFrame	This class represents the graphical window used for building the axioms.
AxiomsPanel	This class allows to draw an axiom, to move the nodes, to add edges and nodes, to remove edges or nodes, it provides creation and structuration fonct for all tat can be done on the axiom.
AxiomsToolBar	This class is used by AxiomsFrame, as it's ToolBar.
ConceptPropertyFrame	This class represents the graphical window used to specify the properties of a concept.
ConceptTypePropertyFrame	This class represents the graphical window used to specify the properties of a concept type.
ConceptTypesFrame	This class represents the graphical window used for building the axioms.
ConceptTypesPanel	This class represents the graphical window used for building the concept types hierarchy.
ConceptTypesToolBar	This class represents the graphical window used for building the concept types hierarchy.
Constants	This class contains the constants of the application, including those which are determinated when the appication is launched, such as the application current directory or the dimensions that depend on the screen resolution.
GraphDisplayArea	This class represents the graphical panel into which a graph, or an axiom, can be displayed.
GraphDrawing	This class implements the force directed algorithm for helping in graph drawing.
GraphEditionFrame	This class represents the graphical window used for editing a graph.
GraphEditionPanel	This class represents the panel into which a graph can be drawed by the user.
GraphicalObjectDisplayArea	This class represents the graphical panel into which a graph, or an axiom, can be displayed.
InferenceEngine	This class represents the graphical window into which the inference engine is running.
InferenceEngineFrame	This class represents the graphical window into which the inference engine is running.
InstancePropertyFrame	This class represents the graphical window used to specify the properties of an instance of concept type.
InterfaceIcon	This class provides method to create icons for the interface.
KBDisplayArea	This class represents the graphical window into which the knowledge base is displayed.
MainFrame	This class represents the general graphical window for the application.
MetaGraph	This class is used to build ontology metagraph.

Figura AII - 32 Pachetul ui (1)

MainFrame	This class represents the general graphical window for the application.
MetaGraph	This class is used to build ontology metagraph.
OntL_CGAxiomListCellRenderer	This class represents the list cell renderer used to display the list of axioms in the axiom frame.
OntL_CGMutableTreeNode	This class represents tree nodes used in the TooCoM interface.
OntL_CGProgressBar	This class represents the graphical window used to present and access the different objects of the ontology.
OntL_CGTreeCellRenderer	This class represents the tree cell renderer used to display the cells of the ontology summary.
OntologyPropertyFrame	This class represents the graphical window used to specify the properties of an ontology
OntologySummaryFrame	This class represents the graphical window used to present and access the different objects of the ontology.
OperationalizationBox	This class represents the labeled box used in the form dedicated to the operationalization of axioms.
OperationalizationForm	This class represents the graphical window into which the context of use of axioms are set.
RelationPropertyFrame	This class represents the graphical window used to specify the properties of a relation.
RelationTypePropertyFrame	This class represents the graphical window used to specify the properties of a relation type.
RelationTypesFrame	This class represents the graphical window used for building the relation types hierarchy.
RelationTypesPanel	This class represents the graphical window used for building the relation types hierarchy.
RelationTypesToolBar	This class represents the graphical window used for building the relation types hierarchy.

Overview Package Class Use Tree Deprecated Index Help

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

Figura AII - 33 Pachetul ui(2)

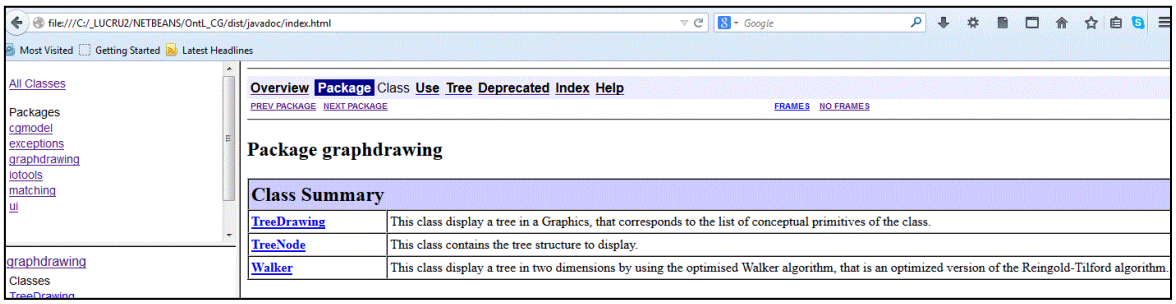


Figura AII - 34 Pachetul graphdrawing

III. ANEXA III – FORMALISMUL GRAFURILOR CONCEPTUALE FOLOSIT ÎN LUCRARE

O *prezentare informală* a grafurilor conceptuale (CG) a fost realizată în ANEXA I, secțiunea 2.3.

Modelul formal - bazat pe teoria structurilor conceptuale - introdus de către Sowa în anii ’80 [219], ca model de reprezentare a cunoștințelor, a cunoscut numeroase extinderi. Extinderile vizează creșterea expresivității modelului de bază, prin: *introducerea negației* (FCG[90]), a *tipurilor conceptuale conjunctive*, a *referenților de tip mulțime*, a *referenților de tip graf conceptual* (cazul grafurilor conceptuale imbricate, NCG[91], care să permită reprezentarea unor structuri complexe sau a contextelor [220], [221]), a *legăturilor de co-referință (de identitate)* [206], *utilizarea regulilor și a constrângerilor* [175], *raționament fuzzy cu CG* sau *actori, demoni sau procese* pentru reprezentarea aspectelor dinamice a cunoașterii. La ora actuală există o multitudine de abordări şi formalizări, unele dintre acestea constituind încă subiecte de cercetare. Dintre cercetătorii de marcă, putem aminti: ***Chein şi Mugnier***, Baget, Salvat, Kerdiles, Leclère, Genest, Dieng, Lukose, Mineau, Kabaj.

Formalismul folosit în lucrare este modelul de bază al CG (SCG[92]), extins cu reguli și constrângeri.

În secțiunile următoare justificăm alegerea formalismului, precizăm sensul termenilor utilizați în lucrare și sistemul formal pe care îl folosim. *Menționez că fundamentele teoretice prezentate aici reprezintă rezultatul unei munci de sinteză pe baza lucrărilor referite în Anexa I* (secțiunea 2.3) și *a lucrărilor* [222], [223], [224], [225], [226], [227], [228], [229]. Definirea termenilor și formalizarea prezentată în secțiunile următoare au la bază lucrările enumerate anterior.

AIII.1. DEFINIREA TERMENILOR FOLOSIȚI ÎN LUCRARE

Definiția AIII-1: Un tip reprezintă o mulţime de concepte sau o mulţime de relaţii conceptuale cu aceleaşi proprietăţi.

Definiția AIII-2: Un *tip conceptual (*pe scurt, *concept)* reprezintă o mulţime de concepte având aceleaşi proprietăţi. Tipurile de concepte sunt organizate într-o ierarhie, pe baza unei relaţii de ordine parţială, AKO (engl., *“a_kind_of”*) (care poate fi sau nu o latice).

[90] FCG – grafuri conceptuale complete (engl., *“Full Conceptual Graphs”*)

[91] NCG – grafuri conceptuale imbricate (engl., *“Nested Conceptual Graphs”*)

[92] SGC – grafuri conceptuale simple (engl., ”*Simple Conceptual Graphs*”)

Definiția AIII-3: Un *tip relaţie (relaţie conceptuală)* reprezintă o mulţime de relaţii care exprimă aceeaşi asociere. Relaţiile de aceeaşi aritate pot fi şi ele organizate într-o latice.

Definiția AIII-4: Un concept corespunde unui obiect (concret sau abstract) al universului de discurs. *Eticheta* unui concept este formată din două părţi: un *tip* care clasifică obiectul reprezentat şi un *referent* care identifică obiectul reprezentat. Referentul trebuie să fie conform cu tipul.

Definiția AIII-5: Un referent este un reprezentant al universului de discurs în baza de cunoştinţe. Un referent poate fi generic sau individual.

Definiția AIII-6: Un *concept generic* este un concept al cărui referent nu este cunoscut (există un obiect al universului de discurs, neindentificat încă); referentul este unul generic.

Definiția AIII-7: Un *concept individual* este un concept al cărui referent este cunoscut şi reprezintă o instanţă.

Observaţie:

În formalismul grafurilor conceptuale, majoritatea cercetătorilor folosesc termenul de *concept generic*, echivalentul noţiunii de clasă din programarea orientată obiect. Pentru evitarea confuziilor care pot apare în legătură cu meta-proprietatea de genericitate a conceptelor (concepte care nu admit extensiuni), consider că termenul de *concept general* ar fi mai potrivit.

Definiția AIII-8: O *legătură de co-referinţă* este o legătură între două concepte considerate identice (aceleaşi).

Definiția AIII-9: Un *arc/muchie etichetată* leagă o un nod relaţie conceptuală de un nod tip de concept într-un graf conceptual.

Definiția AIII-10: Numim *relaţie conceptuală* un nod de asociere între două sau mai multe concepte. Este compusă din *eticheta relaţiei* care clasifică relaţia şi arcele/muchiile etichetate care leagă relaţia de conceptele asociate.

Definiția AIII-11: Un *graf de definiţie* este un graf care defineşte un tip indicând forma pe care trebuie să o aibă graful care implică instanţele acestui tip.

AIII.2. JUSTIFICAREA ALEGERII FORMALISMULUI

Aparent simple, grafurile conceptuale furnizează nu numai un mod de reprezentare grafică a cunoaşterii, ci şi un model matematic bazat pe teoria grafurilor şi a logicii. Raţionamentul asupra cunoștințelor reprezentate poate fi realizat (așa cum s-a ilustrat și în Anexa I) prin rezoluție logică sau prin algoritmi asupra grafurilor (proiecție).

Suportul – ontologie a domeniului

Cunoştinţele din domeniul de interes sunt grupate într-o structură numită **suport**, în raport cu care se definesc grafurile conceptuale. Suportul constituie ontologia. Tipurile de relaţii, tipurile de concepte şi referenţii individuali folosiţi în grafurile conceptuale trebuie declarate sau definite în cadrul suportului, cu scopul de a le ordona şi/sau a impune constrângeri asupra acestora.

Tipurile de concepte sunt ordonate într-o structură de latice finită pe baza unei relaţii de ordine parţială, AKO (*subtip_al,* engl. "*a_kind_of*"); relaţia constituie o relaţie de specializare/generalizare (subsumare) între tipurile de concepte. În plus, referenţii individuali ai unui tip de concept trebuie să respecte relaţia de "conformitate" (*is-a*), verificată tot pe baza informaţiilor conţinute în cadrul suportului.

Tipurile de relaţii conceptuale (de aceeaşi aritate) pot fi şi ele ordonate printr-o relaţie de subsumare.

Astfel, subtipurile (specializările) unui tip de concepte sau de relaţii pot fi *directe* (imediate) sau *indirecte* (în cazul existenţei unor tipuri intermediare).

Ierarhia de tipuri definită în cadrul suportului semnifică implicaţia logică. Tocmai aceste relaţii de subsumare a tipurilor determină relaţia de subsumare asupra etichetelor nodurilor grafurilor conceptuale definite pe suportul respectiv şi induc relaţia de subsumare asupra grafurilor, descrisă în termenii algoritmilor grafurilor prin "proiecţie".

Ontologia domeniului poate varia de la *forma cea mai simplă* (cea de ontologie nucleu), care cuprinde doar ierarhia tipurilor de concepte, cea a tipurilor de relaţii şi marcatorii individuali, până la *forma unei ontologii puternic formalizate* (engl., "heavy-weight ontology"), capabilă să exprime proprietăţile şi axiomele din domeniu.

Deasemenea, utilizarea interpretării logice a suportului sau a celei din teoria mulţimilor oferă posibilităţi diverse de a obţine sisteme operaţionale de reprezentare a cunoştinţelor, bazate pe grafuri conceptuale.

Modelul grafurilor conceptuale - bazat pe teoria structurilor conceptuale

Modelul grafurilor conceptuale (modelul iniţial propus de Sowa, dar şi extinderile acestuia) se bazează pe ***teoria structurilor conceptuale*** [175]; este un model simplu, care unifică:

- ideile lui Pierce, care fac distincţie între tipurile de concepte (exprimate prin c-noduri (echivalente claselor din OOP) şi atomi, individuali sau conceptele instanţă (care reprezintă instanţe ale c-nodurilor);
- ideile lui Aristotel, cu privire la moştenirea tipului [175];
- ideile lui Leibniz, care propune existenţa unei latici semantice de descriere a universului.

Posibilitatea structurării cunoaşterii domeniului pe mai multe niveluri

Formalismul grafurilor conceptuale oferă posibilitatea de a structura cunoştinţele exprimate pe următoarele niveluri:

- Nivelul terminologic sau lingvistic [175], folosit pentru definirea vocabularului domeniului, nivel la care semnificaţia termenilor se structurează într-un sistem.
- Nivelul ontologic, folosit pentru specificarea definiţiilor, regulilor şi constrângerilor asupra cunoaşterii domeniului.
- Nivelul aserţional, utilizat în conceperea grafurilor conceptuale, care pot reprezenta fapte, aserţiuni ale domeniului sau interogări. Proiectarea acestor grafuri are la bază vocabularul conceptual.
- Nivelul epistemologic este asigurat de posibilitatea de a identifica în mod clar tipurile de cunoştinţe, prin *diferenţiere*.

- Nivelul logic este asigurat de semantica logică şi asamblistă a cunoştinţelor descrise cu ajutorul modelului cunoaşterii ontologice.

Nivelul terminologic este compus din numele (etichetele) tipurilor de concepte (ale conceptelor generice), numele tipurilor de relaţii conceptuale şi numele marcatorilor individuali. Ontologia terminologică poate fi privită ca un graf conceptual canonic [175] ale cărui concepte sunt explicite şi fixate, deoarece instanţierea relaţiilor sale (AKO şi ISA) este coerentă.

La **nivel conceptual** semnificaţia conceptelor este structurată în ierarhiile de tipuri (de concepte generice şi relaţii conceptuale), la diferite niveluri de granularitate. Relaţiile AKO, între două tipuri de concepte, şi relaţia IS-A, între un tip de concept şi instanţele sale, sunt considerate *relaţii interne*. Deoarece doar aceste relaţii nu pot reflecta bogăţia semantică a domeniului, lor li se pot alătura şi alte relaţii, considerate *relaţii externe.*

Nivelul ontologic este folosit pentru reprezentarea semanticii domeniului considerat şi, în particular, pentru a explicita cunoştinţele implicite. La acest nivel, cunoştinţele sunt exprimate prin *reguli*, *constrângeri*, *definiţii ale tipurilor de concepte şi de relaţii*. Toate aceste primitive se bazează pe noţiunea de ***graf conceptual simplu***.

Un *graf conceptual simplu* (SCG) este un graf bipartit, compus din noduri tipuri de concepte şi noduri relaţii conceptuale. Pe baza relaţiei de subsumare între tipurile de concepte şi tipurile de relaţii, este definită o relaţie de subsumare (specializare/generalizare) între SCG. Proiecţia este operaţia fundamentală care permite calculul relaţiei $\leq$ între două grafuri.

O *regulă de producţie* este o regulă de inferenţă de forma R: $G_1 \rightarrow G_2$, în care G_1 şi G_2 sunt grafuri conceptuale simple, reprezentând ***ipoteza***, respectiv ***concluzia***. Reprezentarea grafică a unei reguli este una bicoloră. O regulă este aplicabilă în graful G dacă există o proiecţie a ipotezei (G_1) în G ($G \leq G_1$); în această situaţie, concluzia, G_2, este adăugată la G.

O *constrângere* defineşte condiţiile pe care un SCG trebuie să le îndeplinească pentru a fi valid. O constrângere este formată dintr-o condiţie şi o parte obligatorie. Constrângerile pot fi pozitive sau negative. O constrângere pozitivă exprimă “dacă A este prezent, atunci şi B trebuie să fie prezent”. Un graf G satisface o constrângere pozitivă dacă, pentru orice proiecţie a părţii condiţionale există o proiecţie a părţii obligatorii în G. O constrângere negativă exprimă “dacă A este prezent, atunci B trebuie să fie absent”. Reprezentarea grafică a unei constrângeri este aceeaşi ca a unei reguli (bicolorare).

O *definiţie completă a unui tip concept* exprimă o *condiţie necesară şi suficie*ntă, introducând o afirmaţie de echivalenţă între tipul conceptului şi definiţia acestuia: $t_C(x) \overset{def}{\Leftrightarrow} D(x)$ (definiţia tipului t_C cu variabila x ca parametru formal. În viziunea aristoteliană, genul noului tip este tipul conceptului x, iar $D(x)$ reprezintă diferenţa dintre t_C şi genul (supertipul său). Orice obiect recunoscut prin descrierea sa trebuie să aparţină tipului şi orice instanţă a tipului are atributele descrierii.

Definirea completă a unui tip de relaţie introduce o afirmaţie de echivalenţă între tipul relaţiei şi o abstracţiune n-ară: $t_r(x_1, x_2, ..., x_n) \overset{def}{\Leftrightarrow} D(x_1, x_2, \ldots, x_n)$, în care tipul relaţiei este t_r, cu variabilele $x_1, x_2, \ldots, x_n$ ca parametri formali.

Nivelul aserţional al formalismului este compus din grafurile conceptuale definite în respect cu canonul. La nivelul aserţional, faptele sunt reprezentate prin grafuri conceptuale, definite tot în raport cu vocabularul de la nivelul terminologic. Interogările asupra bazei de cunoştinţe, care verifică dacă o anumită aserţiune este adevărată într-o situaţie particulară, se exprimă tot printr-un graf conceptual (un graf cerere). Un graf conceptual poate reprezenta un fapt, o interogare, o constrângere sau o componentă (ipoteză sau concluzie) a unei reguli.

La **nivel epistemologic**, grafurile conceptuale oferă posibilitatea organizării cunoştinţelor în unităţi mult mai complexe decât simple noduri şi legături sau predicate şi propoziţii şi a realizării de inferenţe asupra acestora. Prin folosirea descrierilor intensionale ale conceptelor, prin expansiunea şi contractarea tipurilor organizate într-o ierarhie, se pot obţine reprezentări multiple ale aceloraşi cunoştinţe, deoarece formalismul furnizează mecanismele de agregare a structurilor simple în structuri complexe. Fiind bazate pe lambda-expresii, toate aceste operaţii permit reprezentarea şi prelucrarea cunoştinţelor la diferite niveluri de granularitate.

Nivelul logic permite realizarea de inferenţe asupra cunoştinţelor. Deşi primitivele formalismului sunt extrem de simple (tipuri de concepte şi tipuri de relaţii conceptuale), raţionamentul poate fi realizat – în funcţie de abordare – prin: reguli de transformare şi proiecţii (în cazul abordării alese în lucrare), prin regulile de deducţie (ale lui Peirce) bazate pe mecanismele contextului, prin mecanismele logicii (prin maparea CG-urilor în formulele logicii de ordinul I) sau prin mecanismele specifice teoriei mulţimilor.

AIII.1. SISTEMUL FORMAL AL GRAFURILOR CONCEPTUALE

Modelul grafurilor conceptuale se compune din ***partea terminologică*** și ***partea aserțională***. Pornind de la noţiunile prezentate implicit sau explicit în cartea lui Sowa [219], cercetătorii Chein şi Mugnier definesc sistemul formal al grafurilor conceptuale simple. Un prim model – cel al S-grafurilor – este propus în 1992 [224]. În 1996, aceeaşi cercetători extind modelul de bază, păstrându-i, însă, proprietăţile fundamentale

AIII.3.1. Definirea formală a suportului și a grafurilor conceptuale

Informaţiile specifice domeniului de aplicaţie (informaţii despre conceptele domeniului şi relaţiile dintre acestea - structurate sau nu în ierarhii care pot fi în ambele situaţii latici - constrângeri sintactice, vocabularul şi ierarhia termenilor) sunt grupate într-o structură – numită **suport**. De aceea, considerăm că suportul furnizează **ontologia** domeniului.

Cu ajutorul suportului se reprezintă cunoştinţele generale ale domeniului, iar propoziţiile legate de acest context sunt reprezentate separat, cu ajutorul **S-grafurilor** (**echivalente grafurilor conceptuale simple** – grafuri neimbricate, care nu conţin negaţii); **S** – de la **S**owa, **s**implu şi **s**uport.

Definiția AIII-12: Ontologia domeniului (suportul) este tuplul $\mathcal{S} = (\mathcal{T}_C, \mathcal{T}_R, \mathcal{M}, conf, \mathcal{B})$, unde:

- $\mathcal{T}_C$ - reprezintă mulţimea tipurilor de concepte, organizate într-o latice finită – pe baza relaţiei AKO (ordine parţială) – având tipul universal (τ sau 1) ca supremum şi tipul absurd ca infimum (⊥ sau 0) ca limite inferioare şi superioare.
- $\mathcal{T}_{\mathcal{R}}$ - mulţimea finită a tipurilor de relaţii conceptuale, fiecare relaţie conceptuală fiind un tuplu de elemente din $\mathcal{T}_C$, iar $\mathcal{T}_C \cap \mathcal{T}_{\mathcal{R}} = \varnothing$.

  Observaţie: În unele modele şi relaţiile sunt organizate într-o latice.
- $\mathcal{B}$ – mulţimea grafurilor stea (numită şi bază pentru suportul S), construită ca bijecţie a mulţimii $\mathcal{T}_{\mathcal{R}}$: $\mathcal{B} = \{\mathcal{B}_{r_i}, r_i \in \mathcal{T}_{\mathcal{R}}\}$.

  Fiecare graf stea reprezintă **semnătura** unei relaţii r_i din $\mathcal{T}_{\mathcal{R}}$. Un astfel de graf stea este format dintr-un r-nod şi o mulţime nevidă şi ordonată de c-noduri generice (fiecare c-nod generic fiind etichetat printr-un element din $\mathcal{T}_C$).
- $\mathcal{M}$ – mulţimea numărabilă a marcatorilor pentru c-noduri: $\mathcal{M} = I \cup \{\star, 0\}$, în care I reprezintă mulţimea marcatorilor individuali pentru concepte, $\star$ este marcatorul generic, iar 0 este marcatorul absurd. Elementele mulţimii $\mathcal{M}$ sunt şi ele structurate într-o latice, pe baza relaţiei de ordine <: două elemente din mulţimea $\mathcal{M}$ sunt incomparabile şi $\forall m \in I$, $0<m<\star$.
- $conf$ - relaţia de conformitate între mulţimea conceptelor şi mulţimea marcatorilor, deci un predicat definit pe $\mathcal{T}_C$ x $\mathcal{M}$, care satisface următoarele axiome: $\forall m \in \mathcal{M}$ şi $\forall t, t' \in \mathcal{T}_C$
  1) $conf(1, m) \wedge \neg conf(0, m) \wedge \neg conf(t, m)$
  2) $t \leq t' \wedge conf(t', m) \rightarrow conf(t, m)$
  3) Dacă $conf(t', m) \wedge conf(t, m) \rightarrow conf(t' \wedge t, m)$, atunci $t' \wedge t > 0$
  4) $\forall t \in \mathcal{T}_c - \{0\}$, avem $conf(t, *) \wedge \neg conf(0, *)$.

Observaţii:

1. Din axioma 2 rezultă că dacă un marcator este conform cu tipul t', este deasemenea conform cu oricare dintre tipurile mai specifice (t) ale lui t'.
2. Din axioma 3 rezultă că dacă un marcator este conform cu tipurile t şi t', este deasemenea conform cu $t' \wedge t$.
3. Argumentele fiecărei relaţii $r_i \in \mathcal{T}_{\mathcal{R}}$ reflectă **constrângerile de tip** impuse prin fiecare dintre grafurile stea $\mathcal{B}_{r_i}$ - deci constrângerile de tip asupra c-nodurilor adiacente (argumentelor) r-nodului (relaţiei) r_i .

Definiția AIII-13: Un S-graf este un multi-graf $G = (C, \mathcal{R}, \mathcal{A}, etich)$, unde:

- G este un graf finit, bipartit, **conex**;
- C - mulţimea c-nodurilor, $C \neq \emptyset$;
- $\mathcal{R}$- mulţimea r-nodurilor;
- $\mathcal{A}$ – mulţimea arcelor
- $etich$ – funcţie care etichetează c-nodurile şi r-nodurile astfel:

- Pentru un r-nod, $etich$ (r) este tipul relaţiei, deci un element din $\mathcal{T}_{\mathcal{R}}$.
- Pentru un c-nod, $etich$ (c) este un tuplu din $\mathcal{T}_C$ x $\mathcal{M}$, care verifică $conf$.

  $etich$(c) = (tip(c), ref (c)), unde tip(c) ∈ $\mathcal{T}_C$, iar ref (c) ∈ (I ∪ ***)**

  $\forall$ c∈C, $conf$ (tip(c), ref (c))

  Observaţii: Dacă ref (c) ∈ I, c este numit **concept individual**

  Dacă ref (c) = *, c este numit **concept generic**
- Gradul în G al unui r-nod de tip r este egal cu gradul lui r_i din graful stea asociat, $\mathcal{B}_{r_i}$ pentru r-nodul r_i şi tipul vecinului de indice i este ≤ cu tipul vecinului de indice i pentru r_i din $\mathcal{B}_{r_i}$.
- Graful este parţial ordonat prin R. Pentru fiecare r∈R, mulţimea arcelor adiacente lui r este total ordonată, iar aceste arce sunt etichetate cu i, $i = \overline{1, grad(r)}$. Notăm cu G(r) mulţimea vecinilor r-nodului r: pentru fiecare c-nod c, considerăm c=G_i (r) dacă există un arc rc etichetat i, iar c este vecinul de indice i al lui r.

Observaţii:

1. Un c-nod izolat este un S-graf.
2. Notând cu E_q mulţimea tuturor etichetelor posibile ale c-nodurilor - deci mulţimea tuplelor de forma (t, m) - aceasta este o latice:
   - Pentru etichetele e = (t, m) şi e' = (t', m'), e ≤ e^l dacă t şi m ≤ m^l.
   - Pentru orice mulţime de S-grafuri ale suportului $\mathcal{S}$, marginea inferioară t a tuturor tipurilor asociate c-nodurilor cu acelaşi marcator individual m, îndeplineşte: t > 0 şi $conf$ (t, m). Altfel spus, dacă m este un t_1 si t_2, atunci m este $t_1 \wedge t_2$ (diferit de ⊥).

Definiţia AIII-14: Un SCG G este **bine format în raport cu suportul S** dacă este conform cu constrângerile definite în **baza suportului S** (mulţimea grafurilor stea).

Observaţie: Un graf bine format conferă argumentelor relaţiilor tipuri cel puţin la fel de specifice ca cele indicate în graful stea corespunzător, iar marcatorii c-nodurilor sunt în relaţie de conformitate cu tipul lor.

Acest prim model impune o serie de restricţii:

- S-grafurile sunt conexe;
- Mulţimea tipurilor de concepte este structurată într-o latice;
- Proiecţia (aşa cum este privită de Sowa), este o **proiecţie injectivă**, care restrânge drastic comparabilitatea a două grafuri (Chein, 1995).

O altă formalizare pentru suport şi grafurile conceptuale simple, dată tot de Chein şi Mugnier – aduce **modelului anterior următoarele extinderi**:

- SCG pot *să nu fie conexe*;
- Ierarhia tipurilor de concepte *este ordonată* (posedă un cel mai mare şi un cel mai mic element), însă *nu este neapărat o latice*;
- Mulţimea tipurilor de *relaţii este ordonată*.
- Relaţia de conformitate (din definiția AIII-12) este înlocuită printr-o relaţie “IS-A” între conceptele individuale şi tipurile de concepte (τ din definiția AIII-15).

Deşi păstrează proprietăţile fundamentale ale modelului S-grafurilor, extinderea modelului de bază influenţează operaţiile elementare de specializare-generalizare şi proiecţia.

Definiția AIII-15: Suportul este tuplul $S=(\mathcal{T}_C, \mathcal{T}_{\mathcal{R}}, \sigma, I, \tau)$, unde:

- $\mathcal{T}_C$ - reprezintă mulţimea tipurilor de concepte, mulţime parţial ordonată, având tipul universal ($\top$ sau 1) ca cel mai mare element şi tipul absurd ($\bot$ sau 0) ca cel mai mic element.
- $\mathcal{T}_{\mathcal{R}}$ - mulţimea tipurilor de relaţii conceptuale, partiţionată în mulţimi de relaţii cu aceeaşi aritate: $\mathcal{T}_{\mathcal{R}}= \mathcal{T}_{R_{i_1}} \cup \mathcal{T}_{R_{i_2}} \cup \ldots \mathcal{T}_{R_{i_j}}$. Mulţimea $\mathcal{T}_{R_{i_j}}$ reprezintă mulţimea relaţiilor de aritate i_j, $i \neq 0$. Orice $\mathcal{T}_{R_{i_j}}$ posedă un cel mai mare şi un cel mai mic element.
- σ asociază oricărui tip de relaţie tipul maximal al fiecăruia dintre argumentele sale; mai precis, o aplicaţie care pentru $\forall\, t_r \in \mathcal{T}_{R_{i_j}}$ asociază $\sigma(t_r) \in (\mathcal{T}_C)^{i_j}$ şi satisface condiţia: Pentru $\forall\, tr_1, tr_2$ din $\mathcal{T}_{R_{i_j}}$, dacă $tr_1 < tr_2$, atunci $\sigma(tr_1) < \sigma(tr_2)$. Altfel spus, tipul asociat argumentului k al t_{r_1} este $\leq$ ca tipul asociat argumentului k al tr_2. Vom nota argumentul de indice i pentru $\sigma(t_r)$ cu $\sigma_i(t_r)$.
- I reprezintă mulţimea marcatorilor individuali. Mulţimea tuturor marcatorilor, $\mathcal{M}$, este şi ea ordonată: marcatorul generic * este mai mare ca orice marcator individual, iar marcatorii individuali nu sunt comparabili.
- τ este o aplicaţie definită pe I cu valori în $\mathcal{T}_C - \{\bot\}$, care asociază un concept tip t fiecărui marcator individual m.

Observaţie: În definiția AIII-15 nu se consideră decât tipurile (de concepte sau relaţii) atomice, nedefinite. Există și mecanisme de definire a unor noi tipuri şi de integrare a acestora în cadrul suportului, plecând de la cele existente.

Definiția AIII-16: Un graf conceptual simplu (SCG) este un multi-graf neorientat, bipartit, nu neapărat conex, $G = (C, \mathcal{R}, \mathcal{A}, etich)$, unde:

- C - mulţimea c-nodurilor, $C \neq \emptyset$;
- $\mathcal{R}$- mulţimea r-nodurilor;
- $\mathcal{A}$ – mulţimea arcelor; arcele adiacente oricărui r-nod sunt total ordonate (prin numerotare 1, …., grad(r)). Se notează G_i (r) vecinul de indice i al lui r în G.
- $etich$ – funcţie care etichetează c-nodurile şi r-nodurile astfel:
  - Pentru un r-nod, $etich$ (r) este tipul relaţiei, deci un element din $\mathcal{T}_{\mathcal{R}}$.
    Pentru un c-nod, $etich$ (c) este un tuplu din $(\mathcal{T}_C - \{\bot\}) \times \mathcal{M}$.
    Observaţie: Dacă ref (c) $\in I$, c este numit **concept individual**
    Dacă ref (c) = *, c este numit **concept generic**.
  - $etich$ satisface constrângerile impuse de σ şi τ.
    - $\forall$ r $\in \mathcal{R}$, tip(Gi (r)) $\leq$ σ_i (tip(r));

- $\forall$ c $\in C$, dacă ref(c) $\in I$, atunci (tip(c)) = τ (ref(c)).

Observaţie: O restricţie mai slabă poate fi (tip(c)) τ (ref(c)). Oricărei etichete e = (t, m), cu m $\in I$ i se asociază un tip superficial ("de suprafaţă"), t, şi unul profund, τ (m), utilizat în operaţii.

Deoarece un SCG poate reprezenta cunoştinţe de natură diferită – fapte, scopuri, definiţii de tip (aşa cum se va arăta în paragrafele următoare), reguli, constrângeri, etc., modelul grafurilor conceptuale poate fi considerat un model omogen.

Mulţimea S-grafurilor definite pe suportul $\mathcal{S}$ este mulţimea S-grafurilor care pot fi obţinute (sau **derivate**) pornind de la baza $\mathcal{B}$ (mulţimea grafurilor stea care impune o serie de constrângeri de care este legată "bine-formarea"), printr-o secvenţă de reguli elementare de specializare. Sowa introduce noţiunea de "bază canonică", care generalizează mulţimea grafurilor stea.

Canonul (ontologia) conţine toate informaţiile necesare pentru obţinerea de noi grafuri canonice: ierarhia tipurilor de concepte şi a tipurilor de relaţii ($\mathcal{T}$), mulţimea marcatorilor individuali (I), relaţia de conformitate (notată **::**) şi **baza canonică** (o mulţime de grafuri conceptuale, $\mathcal{B}$, cu etichete în $\mathcal{T}$ X ($I \cup$ {*})).

Definiția AIII-17: O bază canonică $\mathcal{B}$ asociată unui suport $\mathcal{S}$ este o mulţime finită de CG conexe pentru $\mathcal{S}$. Operaţiile elementare de specializare se mai numesc reguli canonice de formare. Un nou graf conceptual (numit - de exemplu - G) poate fi obţinut din altele prin *compunerea regulilor canonice de formare*. Mulţimea grafurilor canonice care pot fi obţinute reprezintă închiderea lui $\mathcal{B}$ sub regulile de formare canonică (operaţiile elementare de specializare).

Introducerea bazei canonice are ca urmare faptul că la aplicarea regulilor de formare trebuie să se ia în considerare constrângerile impuse prin bază. Astfel, baza canonică este folosită ca o **mulţime generatoare pentru grafurile conceptuale derivate**. Unui suport $\mathcal{S}$ i se pot asocia mai multe baze canonice (figura AIII- 1).

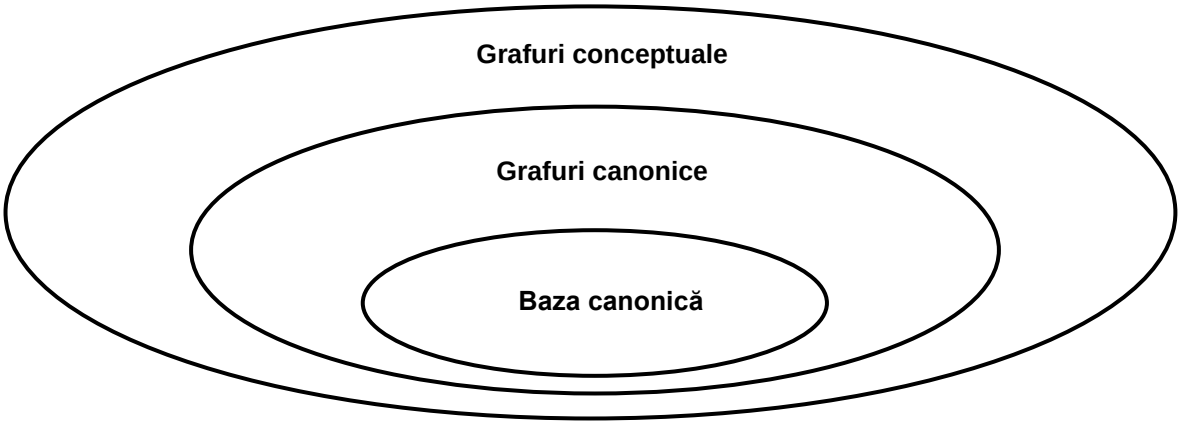


Figura AIII- 1 Bază canonică, grafuri canonice și grafuriconceptuale

AIII.3.2. Raționamentul în limbajul de reprezentare a grafurilor conceptuale

Fundamentul raţionamentului îl constituie operaţia de **specializare-generalizare** (echivalentă subsumării logice) asupra mulţimii grafurilor conceptuale.

Conform lui Sowa, relaţia de generalizare/specializare ($\leq$) poate fi definită în următorii termeni:

- Din *punct de vedere a grafurilor*:
  - Prin transformări elementare (numite **reguli de formare**), fiecare dintre acestea verificând constrângerile legate de bine-formare; intuitiv, dându-se două CG, G şi H, G este o specializare a lui H dacă G poate fi obţinut din H şi eventual alte grafuri printr-o compunere a aplicării regulilor de formare.
  - Printr-un **morfism al grafurilor, proiecţia** (secvenţă a *specializărilor elementare*, numite şi *reguli de formare canonică*).

- Din *punct de vedere logic*:
  - Subsumarea logică (implicaţia logică, aplicând raţionamentul logic asupra formulelor asociate grafurilor).

Operaţiile pe grafuri asigură aceeaşi putere a raţionamentului ca şi ***deducţiile logice*** (dacă G este o specializare a lui H, atunci interpretarea lui G implică interpretarea lui H). În [224] se demonstrează că *regulile de generalizare constituie o mulţime completă şi consistentă de reguli de inferenţă asupra formulelor logice asociate CG*.

AIII.3.2.1 Raționament prin algoritmi ai grafurilor

AIII.3.2.1.1 Operații de specializare/generalizare

OPERAȚII ELEMENTARE

Operaţiile ***elementare de specializare*** definite de Sowa (numite reguli de formare canonică) sunt:

- **Ştergerea unui r-nod "geamăn" – simplificarea** (operaţie unară)

  Definiția AIII- 18: Două r-noduri ale aceluiaşi SCG sunt numite noduri gemene dacă ele au aceeaşi etichetă şi aceeaşi vecini ($\forall$i, vecinii de indice i ai celor două r-noduri sunt identici).

  Dacă un S-graf G are două r-noduri gemene, prin ştergerea (suprimarea) unuia dintre noduri se obţine S-graful G'.
- **Diminuarea etichetei unui c-nod – restricţionarea** (operaţie unară)
  Dacă c este un c-nod al S-grafului G, prin restricţionarea lui c se obţine S-graful G'. Restricţionarea lui c se realizează prin înlocuirea etichetei acestuia (e) cu una mai mică (e'), e'$\leq$e, care satisface condiţia de conformitate, $conf$(e') . Copierea poate fi privită ca un caz particular al restricţionării.
- **Joncţiunea elementară a două c-noduri cu aceeaşi etichetă**
  Fie G şi G', două S-grafuri (nu neapărat distincte) cu c-nodul c, respectiv c-nodul c', cu aceeaşi etichetă. Joncţiunea elementară a G şi G' în rapot cu c şi c' este S-graful G" obţinut prin identificarea şi joncţiunea (unificarea) c-nodurilor c şi c'.

Observaţii:

1. Dacă G şi G' sunt identice, operaţia se numeşte **joncţiune internă** (operaţie unară).
2. Dacă G şi G' sunt distincte, operaţia se numeşte **joncţiune externă** (operaţie binară).

În cazul **modelului extins**, în care *grafurile pot să nu fie conexe*, iar *relaţiile sunt şi ele ordonate*, apar următoarele **operaţii de specializare**:

- Simplificarea unui r-nod;
- Restricţionarea etichetei unui c-nod;
- *Restricţionarea etichetei unui r-nod* (operaţie care trebuie să păstreze constrângerile impuse prin σ);
- Joncţiunea internă (nodurile fuzionate pot aparţine aceleaşi componente conexe sau a două componente conexe distincte);
- *Joncţiunea externă*, care *nu mai este o operaţie elementară de specializare*, fiind o compunere a sumei disjuncte şi a unei joncţiuni interne.

Definiţia AIII- 19: Graful *G este o specializare a lui H* $(G \leq H)$ dacă există o secvenţă de operaţii elementare de specializare care se termină cu G. Vom nota relaţia de specializare cu $\leq$. Pentru mulţimea de S-grafuri E, notăm Σ(E) ***închiderea lui E sub operaţiile de specializare***. Un S-graf G din Σ(E) este numit **derivat** din E.

Operaţiile ***elementare de generalizare*** sunt inverse operaţiilor elementare de specializare:

- **Adăugarea (duplicarea) unui r-nod geamăn**.
- **Mărirea etichetei unui c-nod**: înlocuirea etichetei unui c-nod cu o etichetă mai mare, astfel încât noul tip să satisfacă constrângerile impuse prin baza B (pentru fiecare arc rc cu eticheta i, tip(c)< tipul vecinului de indice i al r-nodului din graful stea corespunzător).
- **Scindarea (expansiunea)**: duplicarea unui c-nod, notat c, în două c-noduri c_1 şi c_2, cu etichete identice, iar mulţimea arcelor adiacente acestora constituie o bipartiţie a mulţimii arcelor adiacente lui c.

Observaţie: Operaţiile de restricţionare şi mărirea etichetei unui c-nod trebuie să respecte constrângerile impuse prin suport (ontologie): creşterea tipului unui c-nod este constrînsă prin bază, iar micşorarea sa este constrânsă prin relaţia de conformitate.

În cazul **modelului extins**, în care *grafurile pot să nu fie conexe*, iar *relaţiile sunt şi ele ordonate*, apar următoarele **operaţii de generalizare**:

- Duplicarea unui r-nod;
- Mărirea etichetei unui c-nod (trebuie să respecte constrângerile impuse prin σ);
- *Mărirea etichetei unui r-nod*;
- Scindarea (expansiunea);
- *Descompunerea (suprimarea anumitor componente conexe).*

Definiţia AIII-20: Graful H este o generalizare a lui G (H ≥ G) dacă H poate fi obţinut din G printr-o o secvenţă de operaţii elementare de generalizare. Vom nota relaţia de generalizare cu ≥. Pentru mulţimea de S-grafuri E, notăm Γ(E) ***închiderea lui E sub operaţiile de generalizare***.

Observaţie: Dacă o operaţie elementară de specializare (respectiv generalizare) se realizează de la G la H, atunci există o operaţie elementară inversă, de generalizare (respectiv specializare) de la H la G. Ca urmare, are loc următoarea proprietate:

Proprietatea AII- 1: G este specializare a lui H dacă şi numai dacă H este o generalizare a lui G.

Observaţie: Mulţimea S-grafurilor care pot fi definite pentru suportul S cu baza B este Σ(B $\cup${[1, *]}), unde [1, *] este S-graful redus la un singur c-nod generic, de tip universal. În particular, pentru orice mulţime E de S-grafuri, Γ(E) şi Σ(E) sunt incluse în Σ(B $\cup$ {[1, *]}).

Pentru **modelul extins**, în ceea ce priveşte relaţia de specializare-generalizare, rămân valabile:

- Echivalenţa între $\leq$ şi proiecţie;
- Existenţa unui CG neredundant unic în orice clasă de echivalenţă modulo $\leq$;
- În modelul extins mulţimea grafurilor neredundante are o structură de latice.

OPERAȚII COMPLEXE

Pe lângă operaţiile elementare de specializare – generalizare, pot fi definite pentru SCG operaţii complexe, cum ar fi: *specializarea comună maximală*, *generalizarea comună minimală*, *distanţa minimală* între grafurile G și H.

Definiția AIII- 21: Fie grafurile conceptuale G, H şi K. Dacă G este o specializare a lui H (G $\leq$ H) şi G este o specializare a lui K (G $\leq$ K), atunci G este numit **generalizarea comună** a grafurilor H şi K.

Definiția AIII- 22: Fie grafurile conceptuale G, H şi K. Dacă H este o specializare a lui G (H$\leq$G) şi K este o specializare a lui G (K$\leq$G) atunci G este numit **specializarea comună** a grafurilor H şi K.

Aceste operaţii complexe se bazează pe noţiunea de proiecţii compatibile.

AIII.3.2.1.2 Operații de derivare

Prin aplicarea succesivă a operaţiilor elementare de formare se obţine un nou graf conceptual (notat, de exmplu, cu G). Secvenţa operaţiilor elementare de specializare se numeşte **derivare**. G este o specializare a fiecărui CG H din care este derivat, notaţia fiind G$\leq$H.

Sowa (1984) defineşte o secvenţă de derivare în termenii unui graf de derivare. Dându-se un graf de derivare D al lui G, se consideră orice sortare topologică $x_0...x_n$ din D (orice ordonare $x_0...x_n$ a mulţimii vârfurilor, astfel încât pentru orice arc $\langle ab\rangle$, $a = x_i, b = x_j$ implică $i < j$). Secvenţa de etichete $(G_0 = etich(x_0)...G_n = etich(x_n))$ este o secvenţă de derivare a lui G.

Observaţii:

1. G este o specializare a lui H (G $\leq$ H) dacă G poate fi derivat din H prin operaţii de specializare. Deoarece ca operaţie elementară de specializare există şi o operaţie binară, derivarea lui H din G poate avea şi alte surse în afară de H.
2. G este o generalizare a lui H (G ≥ H) dacă G poate fi derivat din H printr-o secvenţă de operaţii de generalizare. Deoarece acestea sunt doar operaţii unare, G poate fi obţinut doar din H, iar *derivarea de la G la H este o cale ("path").*

AIII.3.2.1.3 Proiecţii şi morfisme în SCG

Proiecţia (morfism de grafuri) corespunde unei secvenţe de derivare. Este operaţia fundamentală asupra grafurilor conceptuale, deoarece toate celelalte operaţii pot fi exprimate în termenii proiecţiei.

Definiţia AIII-23: O **proiecţie** Π, a grafurilor conceptuale simple (S-grafurilor), de la SCG G = (C, R, A, etich) la SCG H = (C', R', A', etich') este perechea ordonată de aplicaţii Π = (f, g), f: R → R', g: C → C', care satisface condiţiile:

(1) $\forall r \in R$, $\forall$ i ∈ {1, …, grad(r)}, G_i (r) = c implică g(c) = H_i (f(r))

(pentru orice arc $\langle rc \rangle$ etichetat cu i în A, $\langle f(r)g(c) \rangle$ este un arc i în A').

(2) ∀r ∈ R, etich'(f(r)) = etich(r)

(3) ∀c ∈ C, etich'(g(c)) ≤ etich(c)

În modelul extins [199] – în care şi relaţiile sunt ordonate, operaţia de proiecţie permite şi micşorarea etichetelor relaţiilor, deci condiţia (2) din proiecţie (morfism de grafuri) corespunde unei secvenţe de derivare. Este operaţia fundamentală asupra grafurilor conceptuale, deoarece toate celelalte operaţii pot fi exprimate în termenii proiecţiei.

Definiţia AIII-23: O **proiecţie** Π, a grafurilor conceptuale simple (S-grafurilor), de la SCG G = (C, R, A, etich) la SCG H = (C', R', A', etich') este perechea ordonată de aplicaţii Π = (f, g), f: R → R', g: C → C', care satisface condiţiile: este înlocuită cu:

(2') ∀r∈ R, etich'(f(r)) ≤ etich(r).

Definiţia AIII-24: Un **morfism** al S-grafurilor (grafuri bipartite etichetate), de la G = (C, R, A, etich) la H = (C', R', A', etich') este perechea ordonată de aplicaţii Π = (f, g), care satisface condiţiile (1) şi (2) din definiţia proiecţiei, iar condiţia (3) din definiţia anterioară este înlocuită cu (3'):

(3') ∀ c∈ C, etich'(g(c)) = etich(c)

Observaţii:

1. Morfismul este un caz particular al proiecţiei.
2. Proiecţia este un morfism care păstrează etichetele r-nodurilor şi restricţionează etichetele c-nodurilor. "Micşorarea" ("diminuarea") etichetei unui c-nod se realizează prin înlocuirea tipului c-nodului cu un tip "mai mic" (mai specific); dacă c-nodul este generic – înlocuirea marcatorului generic cu unul individual (care satisface relaţia de conformitate).
3. Definiţia proiecţiei SCG este obţinută pornind de la cea de morfism al SCG, prin înlocuirea relaţiei (3) cu:
   - ∀ c∈ C, etich'(g(c)) ≤ etich(c) (pentru inf-proiecţie)
   - ∀ c∈ C, etich'(g(c)) ≥ etich(c) (pentru sup-proiecţie)
4. În terminologia lui Sowa, proiecţia desemnează, de fapt, o inf-proiecţie.

AIII.3.2.2 Interpretarea logică

AIII.3.2.2.1 Semantica propusă de Sowa (Sowa, 1984)- Operatorul Φ

Interpretarea logică a unui CG

Unui graf conceptual G i se poate asocia o formulă bine formată, Φ(G), în limbajul logicii de ordinul întâi, astfel:

- Fiecărui c-nod generic c_i i se asociază variabila x_i (există o bijecţie între c-nodurile generice şi variabilele asociate).
- Fiecărui marcator individual, m, i se asociază o constantă, notată tot m (există o bijecţie între marcatorii individuali şi constante, dar nu neapărat o bijecţie într c-nodurile individuale şi constante).
- Fiecărui element din $\mathcal{T}_C$ i se asociază un predicat unar care specifică apartenenţa la un anumit tip de concept, $t_{c_i}(id_{c_i})$, unde t_{c_i} este predicatul asociat tipului de concept c_i, iar id_{c_i} este termenul asociat .
- Fiecărui element din $\mathcal{T}_{\mathcal{R}}$ i se asociază un predicat n-ar, conform semnăturii relaţiei (grafului stea asociat), $t_r(id_{c_1},...,id_{c_n})$.

Dacă p(G) este formula formată din *conjuncţia tuturor atomilor asociaţi nodurilor* lui G, Φ(G) este *închiderea existenţială* a lui p(G).

Definiţia 3.31: Dându-se un suport S numim S-formulă bine-formată formula de forma Φ(G), unde G este un S-graf al lui S. Mulţimea S-formulelor este $\Phi(\Sigma(B \cup \{[1,*]\})$.

Interpretarea logică a suportului

Suportului S (definit în AIII-15) i se poate asocia o mulţime de formule Φ(S) ca urmare a interpretării logice a legăturilor "AKO" dintre tipurile de concepte şi/sau tipurile de relaţii.

1. $\forall t_{C_1}, t_{C_2} \in \mathcal{T}_C$, spunem că t_{C_1} acoperă t_{C_2} ($t_{C_2} < t_{C_1}$ şi nu există t_{C_3}, astfel că $t_{C_2} < t_{C_3} < t_{C_1}$) în $\mathcal{T}_C$ dacă $\forall x(t_{C_2}(x) \subset t_{C_1}(x))$ sau $\forall x(t_{C_2}(x) \rightarrow t_{C_1}(x))$
2. $\forall t_{r_1}, t_{r_2} \in \mathcal{T}_{R_n}$ (n – aritatea relaţiei) t_{r_1} acoperă t_{r_2} în $\mathcal{T}_R$: $\forall x_1 ... \forall x_p (t_{r_2}(x_1 ... x_p) \rightarrow t_{r_1}(x_1 ... x_p))$
3. $\forall x \Top(x)$, interpretarea tipului universal
4. $\forall x \perp(x)$, interpretarea tipului absurd
5. $\forall t_r \in T_R$, fie $\sigma(t_r) = (t_{C_1},...,t_{C_p})$, $\forall x_1 ... \forall x_p (t_r(x_1 ... x_p) \rightarrow t_1(x_1) \wedge ... \wedge t_p(x_p))$, interpretarea semnăturilor relaţiilor.
6. $\forall a \in I$, $\tau(a)(a)$.

Consistența [219]: Dacă Φ(S) este mulțimea formulelor asociate suportului, atunci pentru orice două SCG G și H definite pe suport, dacă G ≤ H, atunci Φ (S), Φ (G) → Φ (H) (sau $\Phi(S), \Phi(G) \vDash \Phi(H)$, unde Φ(H) este consecinţa logică pentru Φ(S) şi Φ(G). Altfel spus, dacă graful G este specializarea lui H G ≤ H şi G reprezintă o situaţie adevărată, atunci H va reprezinta tot o situaţie adevărată (interpretarea lui G implică logic interpretarea lui H).

Completitudinea [199]: Dacă $\Phi(S), \Phi(G) \vDash \Phi(H)$ atunci H ≥ G are loc doar dacă G este în **formă normală** (nu există în graf două noduri cu acelaşi marcator individual).

AIII.3.2.2.2 Semantica propusă de Chien & Mugnier (Sowa, 1984)- Operatorul Ψ

Chein propune operatorul Ψ, pentru care proiecția ***este consistentă și completă în raport cu FOL, fără nicio restricție***.

Dacă $\mathcal{S} = (\mathcal{T}_c, \mathcal{T}_\mathcal{R}, I, \star)$ este suportul, atunci fiecărui marcator individual i se asociază o constantă din I, fiecărui concept din $\mathcal{T}_c$ – un predicat binar, iar fiecărei relații de aritate n din T_R^n – un predicat n-ar.

***Observație*:**

Pentru simplificare, numele constantei și a predicatului este același cu numele elementului corespunzător din suport. Astfel, dacă t ∈ $\mathcal{T}_C$ atunci există predicatul binar t(y, x) (conceptul reprezentat prin nodul x este y și are tipul t).

Exemplificarea utilizării operatorilor se găsește în Anexa I, secțiunea 2.3.

Interpretarea logică a suportului

Suportului S i se asociază o mulțime de formule Ψ(S), care corespund interpretării ordinii parțiale a tipurilor conceptelor și relațiilor ($\mathcal{T}_c$ și $\mathcal{T}_\mathcal{R}$):

1. ∀ t_1, t_2 ∈ $\mathcal{T}_C$, și $t_1 \geq t_2$, formula Ψ (t_1, t_2) = ∀x∀y(t_2(x, y) → t_2(x, y)) este adăugată la Ψ(S).
2. ∀ r_1, r_2 ∈T_R^n, și $r_1 \geq r_2$, formula Ψ (r_1, r_2) = ∀x_1….∀x_n(r_2(x_1, .., x_n) → r_2(x_1,….. x_n)) este adăugată la Ψ(S).

Ψ(S) poate fi privită ca și conjuncție a tuturor formulelor corespunzătoare adăugate.

Interpretarea logică a unui SCG

Un graf conceptual G bazat pe suportul S este translatat prin operatorul **Ψ** în formula **Ψ**(S), construită astfel:

AIII.1. Fiecărui concept c, i se asociază o variabilă x_c.

AIII.2. Pentru fiecare relații cu tip(r) = t_r , aritate n și conceptul c_i pe poziția i ($N_G^i(r) = c_i$, i = 1,...., n) se construiește atomul **Ψ**(r) = t_r ($x_{C_1}, \ldots\ldots\ldots, x_{C_n}$).

AIII.3. Atomul **Ψ**(c) i se asociază conceptului c cu tipul tip(c) = t_c : x_c este variabila asociată nodului c și y_c este termenul construit astfel: dacă c are marcator individual, atunci y_c este constanta asociată acestuia, dacă c are marcator generic, atunci y_c este o nouă variabilă.

Astfel, pentru conceptele c_1, c_2, …., c_p cu m marcatori generici, formula asociată este

Ψ(G) = $\exists x_{c_1} \ldots . \exists x_{c_p} \exists y_1 \ldots\ldots . \exists y_m (\bigwedge_{i=1}^{p} \Psi(c_i \wedge \bigwedge_r \Psi(r))$

AIII.4. FAMILIA SCG

Modelul SCG a fost extins prin adăugarea de reguli și constrângeri (modelate ca λ-abstracțiuni).

AIII.4.1 λ-abstracțiuni

Suportul permite reprezentarea ontologiei domeniului, descriind tipurile de concepte şi tipurile de relaţii dintre acestea. Un graf conceptual simplu relativ la un suport S poate reprezenta fapte, scopuri sau ipoteze privind cunoştinţele generale ale ontologiei.

Lambda-abstracţiunile permit exprimarea unor tipuri diverse de cunoştinţe (*explicite sau implicite*), cum ar fi: definirea unor noi tipuri de concepte, definirea unor noi tipuri de relaţii, exprimarea unor constrângeri, definirea unor scheme proptipale ale tipurilor de concepte, sau exprimarea unor reguli de producţie.

Ontologia domeniului trebuie să poată fi îmbogăţită prin adăugarea unor noi tipuri de concepte sau a unor noi tipuri de relaţii. O problemă importantă este legată de locul unde vor fi inserate aceste noi tipuri în cadrul ierarhiilor existente. Utilizarea lambda-expresiilor oferă posibilitatea de a defini tipuri diferite de cunoştinţe.

Definiția AIII- 25: Un λ-graf este o **λ-abstracţiune n-ară** de forma $\lambda x_1 ... x_n G$, unde G este un SCG numit corp, iar x_i reprezintă variabilele asociate c-nodurilor generice ($\forall i = \overline{1,n}$, pentru fiecare c-nod generic se asociază variabila x_i).

- $x_1,...,x_n$ sunt numiţi parametri formali.
- Tuplul $(t_1,...,t_n)$, unde t_i reprezintă tipul conceptului generic de indice i este numită *semnătura lui G*.

AIII.4.2 Reguli și constrângeri

Regulile și constrângerile sunt definite ca și cuple de λ-abstracțiuni (H, C), în care H este graful ipoteză, iar C este graful concluzie.

Reguli

O *regulă* exprimă cunoștințe ca: "*Dacă H este prezent, atunci C trebuie adăugat*":

- $H = \lambda x_1,\ldots, x_n G_1$ este ipoteza
- $C = \lambda x_1,\ldots, x_n G_2$ este concluzia
- $\lambda x_1,\ldots, x_n$ sunt punctele de conexiune (nu pot fi cencepte generice)

Forma unei reguli R: $G_1 \rightarrow G_2$, în care G_1 şi G_2 sunt grafuri conceptuale simple, reprezentând ***ipoteza***, respectiv ***concluzia***. O regulă este aplicabilă în graful G dacă există o proiecţie a ipotezei (G_1) în G ($G \leq G_1$); în această situaţie, concluzia, G_2, este adăugată la G.

O regulă poate fi reprezentată sub forma unui **graf bicolor** (ipoteza – o culoare, concluzia – altă culoare și nu se reprezintă frontierea (punctele de conexiune). Reprezentarea sub această formă este mult mai ușor de înțeles față de reprezentarea sub formă de λ-abstracțiune.

Definiția AIII- 26: Un graf bicolor (G, I) este un graf ale cărui noduri se colorează cu 2 culori (de exemplu, 0 și 1).

Definiția AIII- 27: Regula ca graf bicolor este definită astfel:

- R = (G, I) , care îndeplinește condițiile: subgraful H indus prin nodurile de culoare 0 este un subgraf al lui G (dacă un nod relație are culoarea 0, atunci toți vecinii săi trebuie să aibă culoarea 0).
- H este ipoteza regulii
- Nodurile concepte de culoare 0 cu cel puțin un vecin care nu face parte din ipoteză sunt noduri de frontieră
- Nodurile de culoare 1 împreună cu nodurile de frontieră induc un subgraf C numit concluzia regulii.

Constrângeri

O *constrângere* defineşte condiţiile pe care un SCG trebuie să le îndeplinească pentru a fi valid. O constrângere este formată dintr-o *condiţie* (similară ipotezei unei reguli) şi o *parte obligatorie* (similară concluziei unei reguli). O constrângere poate fi privită tot ca o regulă.

Constrângerile pot fi ***pozitive*** sau ***negative***.

O ***constrângere pozitivă*** exprimă *"Dacă A este prezent, atunci şi B trebuie să fie prezent"*. Un graf G verifică o constrângere pozitivă dacă pentru orice proiecţie a lui A în graf există o proiecţie a lui B în G (G verifică constrângerea dacă fiecare homomorfism de la A la G poate fi extins la un homomorfism de la B la G).

O ***constrângere negativă*** exprimă "*Dacă A este prezent, atunci B trebuie să fie absent*" și modelează cunoștințe care nu trebuie să apară într-un graf. Graful G verifică o constrângere negativă dacă nicio proiecție a ipotezei constrângerii în graf nu poate fi întâlnită în concluzia constrângerii.

Reprezentarea grafică a unei constrângeri este aceeaşi ca a unei reguli (bicolorare).

Semantica regulilor se apropie de cea a constrângerilor: regulile permit o deducție, constrângerile pozitive impun existența unei piese de cunoaștere.

Constrângerile permit validarea bazei de fapte dintr-o bază de cunoștințe[93].

Ca în orice sistem de producţie, raţionamentul poate fi:

- Cu înlănţuire înainte ("forward chaining", "data-driven"), când pornind de la faptele existente (reprezentate prin SCG), prin aplicarea regulilor de producţie se obţin fapte noi.
- Cu înlănţuire înapoi ("backward chaining", "goal-driven"), când pornind de la scop-concluzie, se selectează regula care conduce la scop, iar antecedentul regulii furnizează noi scopuri.
- Mixt.

[93] Baza de cunoștințe este consistentă dacă toate constrângerile sunt satisfăcute.

AIII.4.4 Complexitate

Baget și Mugnier au structurat familia SCG pe 6 modele pentru a studia complexitatea raționamentului.

Forma generală a unei baze de cunoștințe este $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{E}, \mathcal{C})$, unde $\mathcal{F}$ este mulțimea faptelor, $\mathcal{R}$ este mulțimea regulilor de inferență, $\mathcal{E}$ este mulțimea regulilor de evoluție, iar $\mathcal{C}$ este mulțimea constrângerilor (pozitive, $\mathcal{C}^+$ și negative, $\mathcal{C}^-$). Dacă $\mathcal{C}$ este vidă, atunci $\mathcal{R}$ și $\mathcal{E}$ sunt identice.

Ierarhia modelelor și complexitatea sunt ilustrate în figura AIII- 2.

Tabloul complet al rezultatelor privind complexitatea și decidabilitatea este (tabelul AIII- 1), unde *fes* este mulțimea regulilor de expansiune finită, *rr* – mulțimea regulilor, $\mathcal{C}^-$ mulțimea constrângerilor negative, $\mathcal{C}^d$ mulțimea constrângerilor disconecte, iar semnul / indică un caz particular în care problema considerată nu este aplicabilă.

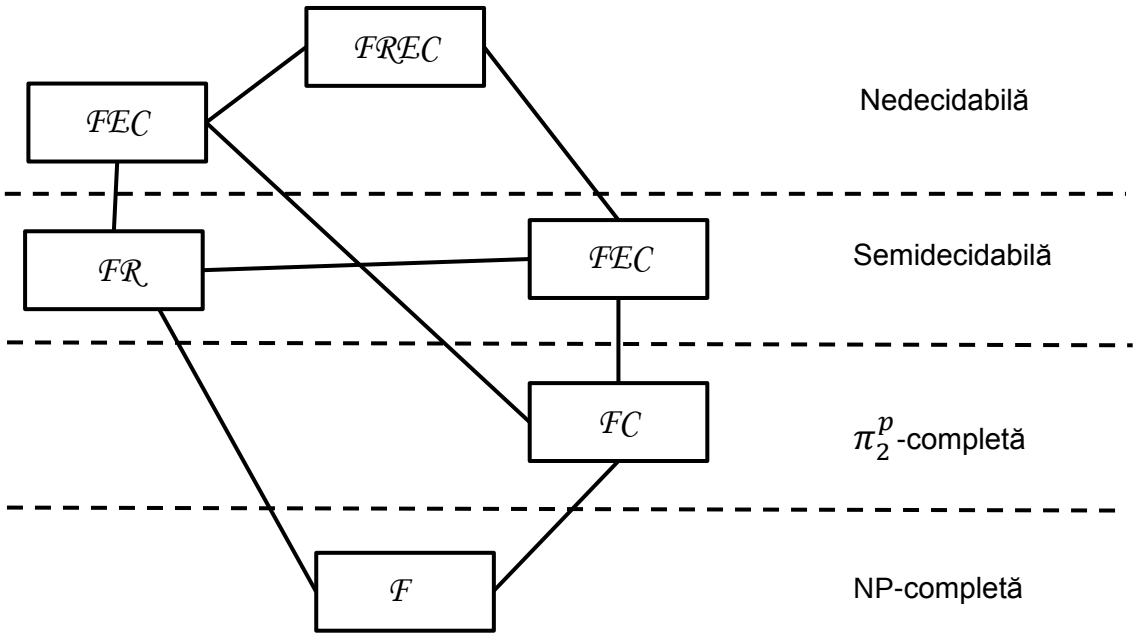


Figura AIII- 2 Familia SGC și complexitatea ([175])

Tabelul AIII- 1 Complexitatea și decidabilitatea ([175])

	Caz General	$\mathcal{R}$fes	ε fes	$\mathcal{R} \cup \varepsilon$ fes	$\mathcal{R}\,\&\,\varepsilon$ rr	C^-	$\mathcal{R}\,\&\,\varepsilon$ rr, C^-	$\mathcal{R}\,\&\,\varepsilon$ rr, C^d
$\mathcal{F}$-DED.	NP-C	/	/	/	/	/	/	/
$\mathcal{FC}$-CONS	π_2^p-C	/	/	/	/	co-NP-C	co-NP-C	co-NP-C
$\mathcal{FR}$-DED.	π_2^p-C	/	/	/	/	DP-C	DP-C	?
$\mathcal{FRC}$-DED.	semi-dec.	dec.	/	dec.	NP-C	/	NP-C	NP-C
$\mathcal{FRC}$-CONS	nedec.	Dec.	/	dec.	π_2^p-C	co-semi-dec	co-NP-C	co-NP-C
$\mathcal{FRC}$-DED.	nedec.	Dec.	/	dec.	π_2^p-C	nedec.	DP-C	?
$\mathcal{FRC}$-DED.	semi-dec.	/	dec.	dec.	$\sum_2^p$-C	semi-dec	$\sum_2^p$-C	$\sum_2^p$-C
$\mathcal{FREC}$-DED.	nedec.	semi-dec	nedec.	dec.	$\sum_2^p$-C	nedec.	$\sum_2^p$-C	$\sum_2^p$-C