SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study / Qualification	Computer science / Engineer
1.7 Form of education	Full time
1.8 Subject code	54.10

2. Data about the subject

2.1 Subject name	***Knowledge-Based Systems***				
2.2 Course responsible / lecturer	Prof. dr. eng. Adrian Petru Groza – Adrian.Groza@cs.utcluj.ro				
2.3 Teachers in charge of seminars / laboratory / project	Assoc. prof. dr. eng. Anca Mărginean Anca.Marginean@cs.utcluj.ro				
2.4 Year of study	IV	2.5 Semester	8	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

3. Estimated total time

3.1 Number of hours per week	5	of which:	Course	2	Seminars	1	Laboratory	2	Project	-
3.2 Number of hours per semester	70	of which:	Course	28	Seminars	14	Laboratory	28	Project	-
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										25
(b) Supplementary study in the library, online and in the field										26
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										25
(d) Tutoring										
(e) Exams and tests										4
(f) Other activities:										

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	80
3.5 Total hours per semester (3.2+3.4)	150
3.6 Number of credit points	6

4. Pre-requisites (where appropriate)

4.1 Curriculum	Artificial Intelligence, Intelligent Systems
4.2 Competence	Important material that you should have learned: first order logic, algorithm design, big-O complexity analysis, heuristic search, logic programming, machine learning. Useful skills that you should have: Linux, Latex, Python, Java programming languages. Functional and Logic programming is a plus.

5. Requirements (where appropriate)

5.1. For the course	Each student is required to enrol on the Moodle platform. By enrolling in this course, each student assumes the responsibility of an active participant in lecture and applications.
5.2. For the applications	

6. Specific competence

6.1 Professional competences	**C3** - Problems solving using specific Computer Science and Computer Engineering tools (1 credit) **C3.1** Identifying classes of problems and solving methods that are specific to computing systems **C3.2** Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results **C3.3** Applying solution patterns using specific engineering tools and mehods **C3.4** Comparatively and experimentaly evaluation of the alternative solutions for performance optimization **C3.5** Developing and implementing informatic solutions for concrete problems **C5 -**Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems (1 credit) **C5.1** Specifying the relevant criteria regarding the lifetime cycle, quality, security and computing system's interaction with the environment and human operator **C5.2** Using interdisciplinary knowledge for adapting the computing system to the specific requirements of the application field **C5.3** Using fundamental principles and methods for security, reliability and usability assurance of computing systems **C5.4** Adequate utilization of quality, safety and security standards in information processing **C5.5** Creating a project including the problem's identification and analysis, its design and development, also proving an understanding of the basic quality requirements **C6 -** Designing intelligent systems (2 credits) **C6.1** Describing the components of intelligent systems **C6.2** Using domain-specific tools for explaining and understanding the functioning of intelligent systems **C6.3** Applying the fundamental methods and principles for specifying solutions for typical problems using intelligent **C6.4** Choosing the criteria and evaluation methods for the quality, performances and limitations of intelligent systems **C6.5** Developing and implementing professional projects for intelligent systems
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	Understanding conceptual instrumentation for knowledge representation and reasoning
7.2 Specific objectives	Applying various knowledge-based techniques aiming to increase the quality of software systems

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
1.Introduction: application case analysis, representative scenarios from different domains, first order logic. Knowledge graphs	2	Slides, Warm-up examples, Quick individual work, Open discussions, Assignments, Round-up quizes Student engagement techniques, Kahoot quizzes	
2. Description logics: concepts, roles, instances, expressivity.	2		
3. Reasoning in description logics. Tableaux-based algorithms	2		
4. Ontology engineering: ontology design and evaluation	2		
5. Description Logic Programs. Natural Language Processing for Description logics	2		
6. Machine Learning for Description Logics	2		
7. Agents for Semantic Web. Jason programming language	2		

8. Knowledge graphs	2		
9. Epistemic logics, dynamic epistemic logic, epistemic puzzles, applications	2		
10. Model checking. Computational Tree Logic	2		
11. Fuzzy systems: fuzzy sets, fuzzy inference, fuzzy expert systems. Fuzzy description Logic	2		
12. Rule-based systems: representation, reasoning methods. Probabilistic rules. Cognitive biases	2		
13. Non-monotonic reasoning.	2		
14. Explainable AI. Regulating AI. AI ethics. AI responsable	2		

Bibliography

1. Hogan, Aidan, et al. "Knowledge graphs." *ACM Computing Surveys (CSUR)* 54.4 (2021): 1-37.
2. F. Baader, W. Nutt, Basic Description Logics, Handbook of Description Logics, Cambridge University Press, May 20, 2010
3. Grigoris Antoniou and Frank van Harmelen, A Semantic Web Primer, second edition, MIT Press, 2008
4. Van Eijck and Verbrugge (eds.), Discourses on Social Software, Amsterdam University Press, 2009
5. Brachman, Ronald J., and Hector J. Levesque. "Knowledge representation and reasoning" *Morgan Kaufmann Publishers,* 2004

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
1. Knowledge graphs. Examples of ontologies	2	Examples, Assignments	
2. Semantic Web. Reusing ontologies. Ontology repositories	2		
3. Defining concepts. Reasoning on concepts	2		
4. Defining roles. Reasoning on roles.	2		
5. Populating ontologies.	2		
6. Rules on top of ontologies. Semantic Web Rule Language	2		
7. Ontology design patterns. Natural language processing for ontologies. LLMs for ontology engineering	2		
8. Querying ontologies. SPARQL	2		
9. Integrating ontologies with other applications. AgentSpeak programming language. JASON tool	2		
10. Ontology enrichment with Machine Learning. DLLearner tool	2		
11. Fuzzy knowledge. Fuzzy Description Logic. FuzzyDL tool	2		
12. Debugging ontologies. Consistency checking. Ontology evaluation	2		
13. Documenting ontologies in Latex.	2		
14. Ontology building competition. Student presentations	2		

Bibliography

1. Groza A. - Ontology Engineering with RACER - an activity based approach, UTPress, 2014
2. Groza, A. Modelling puzzles în First Order Logic, Springer Cham, 2021

Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

The presented scenarios are practical and interactive. The course bridges the gap between abstract formalisms of reasoning and representation and the technologies used by companies (knowledge graphs, RDF, formal verification). In support of the business objectives of companies to develop robust software products and minimize errors, the course includes the presentation of engineering methodologies related to formalizing business rules or ontology engineering. Additionally, through CTL, students train with a formal method of verifying and identifying errors in software packages. In line with XAI (Explainable AI), transparent machine learning algorithms are introduced. The content of the discipline is in line with similar courses at other universities.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the
			final grade
Course	The ability to identify and formulate problems from the real world; The ability to build models for specific problems; The ability of critical analysis	Midterm assessment, Exam	60%
Seminar	The ability to argue and support technical opinions; The ability to choose the appropriate technical tools for a specific problem.	Technical and scientific presentation of a paper	10%
Laboratory	The ability to represent and query knowledge; The ability to identify advantages and disadvantages of the proposed solution; The ability to work in a team."	Lab project assessment	30%
Project	-	-	-

Minimum standard of performance:
Understanding description logics, Meeting deadlines. Engineering a decent ontology.
Grade calculus: 0.2 * midterm + 0.3 * lab + 0.5 * exam
Conditions for participating in the final exam: Lab ≥ 5
Conditions for promotion: Grade ≥ 5

Date of filling in: 07.06.2024	**Teachers**	**Title First name Last name**	**Signature**
	Course	Prof. dr. eng. Adrian Groza	
	Applications	Assoc. prof. dr. eng. Anca Mărginean	

Date of approval in the department 20.02.2024	Head of department, Prof.dr.eng. Rodica Potolea
Date of approval in the Faculty Council 22.02.2024	Dean, Prof.dr.eng. Mihaela Dînșoreanu