National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

АПЕПС Department of Design Automation for Energy Processes and Systems

Model-Driven Design of Software Systems
Work program of the discipline (Syllabus)

Course Requisites

Educational level	*Third (PhD)*
Field of knowledge	*12 Information technology*
Specialty	*121 Software Engineering*
Educational program	*Software Engineering*
Course status	*Normative*
Form of study	*Part-time*
Year of study, semester	*2 year, autumn semester*
Number of ECTS credits	*4 ECTS credits (120 hours)*
End-of-semester control / control measures	*Exam, modular test, blitz poll, calendar control*
Timetable	*http://rozklad.kpi.ua/*
Language of study	*English*
Information about course leader / teachers	Lecturer: Ph.D., Associate Professor Koval Oleksandr Vasyliovych, o.koval@kpi.ua, tel. 067-249-82-46 Practical training: Ph.D., Associate Professor Koval Oleksandr Vasyliovych, o.koval@kpi.ua, tel. 067-249-82-46
Course placement	http://campus.kpi.ua/

Curriculum

1. Description of the discipline, its purpose, subject of study and learning outcomes

The study of the discipline "Model-oriented design of software systems" allows students to form competencies necessary to solve complex problems of professional activities related to the design of software systems using standards and approaches of model-oriented design, in particular on the example of semantic model development subject area of the software system.

The subject of the discipline "Model-oriented design of software systems" are elements of technology for designing a semantic model of the subject area of the software system and its use to build scenarios for the user of the software system.

The purpose of the discipline is to form higher competencies in higher education students:

- Ability to develop new models and scientific methods of designing, developing and researching software effectiveness.

- Ability to make strategic decisions that anticipate and formulate future directions for the development of model-oriented processes, new business products and services.

- Ability to perform pre-processing of data using common means of data extraction; to find new useful data and their relationships.

- Ability to develop new and improve existing models, methods, tools, processes in the field of software engineering, which provide development or provide new opportunities for technology development and use of software.

- Ability to critically rethink existing software engineering technologies and track trends.

- Ability to ensure continuous self-development and self-improvement, responsibility for the development of others in the professional field, adhering to pedagogical ethics, the rules of academic integrity in scientific and pedagogical activities.

- Ability to use adequate methods of effective interaction with representatives of different groups (social, cultural and professional).

- Ability to work in a team, form positive relationships with colleagues, communicate with the general scientific community and the public in the field of software engineering.

- Ability to expand the boundaries of knowledge using the results of original research.

As a result of studying the discipline, applicants for higher education will acquire the following program learning outcomes:

- Be able to develop new and improve existing models, methods, tools in the field of software engineering, which ensure the development of technologies for software development and use.

- Be able to use the modeling language to visualize, specify, design and document artifacts of software systems.

- Be able to apply, develop and improve software verification methods.

- Know the methods of analysis of large amounts of data.

- Be able to develop and improve methods of model-oriented design of information systems to solve theoretical and applied problems in the creation of object-oriented, scenario models.

- Be able to study the operating parameters of software life cycle processes, as well as to analyze the selected methods and tools to support these processes and be able to justify their choice.

- Understand the theoretical foundations underlying research methods of information systems and software, research methodologies and computational experiments.

- Deeply understand the general principles and methods of software engineering, as well as research methodology, apply them in their own research in the field of software engineering and in teaching practice.

- Have advanced conceptual and methodological knowledge of software engineering and cross-border subject areas, as well as research skills sufficient for

conducting scientific and applied research at the level of modern world achievements in the relevant field, gaining new knowledge and / or implementing innovations.

2. Prerequisites and postrequisites of the discipline (place in the structural and logical scheme of education according to the relevant educational program)

Successful study of the discipline "Model-oriented design of software systems" is preceded by the study of disciplines "Formal methods of software engineering" curriculum for doctors of philosophy in the specialty 121 Software Engineering.

The discipline "Model-Oriented Software Systems Design" precedes the discipline "Software Reengineering Methods". The theoretical knowledge and practical skills obtained as a result of mastering the discipline "Model-oriented design of software systems" can be useful for conducting research on the topic of the dissertation.

3. The content of the discipline

The discipline "Model-oriented design of software systems" involves the study of the following topics:

Section 1. Introduction to model-oriented design of software systems.

Topic 1.1. Basic approaches to software modeling.

Topic 1.2. Simulation software using simulation language.

Section 2. Model-oriented approach to building a model of the subject area of the aircraft.

Topic 2.1. Construction of the model of the subject area of the aircraft.

Topic 2.2. Metadata as an element of the description of the subject area of the aircraft.

Section 3. Modeling of the subject area of the software system using semantic modulation of knowledge management.

Topic 3.1. Methodology of semantic modeling of knowledge management system construction.

Topic 3.2. Construction of a conceptual model of the ontology of the subject area.

Topic 3.3. Modeling the relationship between the concepts of the subject area and their properties.

Topic 3.4. Languages of logical inference.

Topic 3.5. Evaluation of the quality of the ontology model.

Topic 3.6. Construction of queries to the description of the ontological model of PS.

Topic 3.7. Methods and technologies of integration of ontological model with relational data ..

Section 4. Modeling of PS user activity on the basis of ontological model of software.

Topic 4.1. Scenario-target approach to the activity of the user of the aircraft and construction of scenarios of the user of the aircraft.

Topic 4.2. Construction of scenarios of user activity and user interface of PS on the basis of ontological model of software.

4. Training materials and resources

Basic literature:

John D. Poole. Model-Driven Architecture: Vision, Standards And Emerging Technologies. – Режим доступу: URL: https://www.omg.org/mda/mda_files/Model-Driven_Architecture.pdf

2. Alberto Rodrigues da Silva. Model-driven engineering: A survey supported by the unified conceptual model. Computer Languages, Systems & Structures, Volume 43, October 2015, Pages 139-155

3. Додонов О. Г., Коваль О. В., Глоба Л. С., Бойко Ю. Д. Комп'ютерне моделювання інформаційно-аналітичних систем: монографія. Київ: ІПРІ НАН України, 2017. 239 с.

4. А.Г. Додонов, В.Р. Сенченко, А.В. Коваль Аналитика и знания в компьютерных системах. Монография. Институт проблем регистрации информации НАН Украины. Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», Киев, 2020, 315 с.

5. Web Protege User Guide. URL: https://protegewiki.stanford.edu/wiki/Main_Page

Additional literature:

3. Marco Brambilla, Jordi Cabot, Manuel Wimmer. Model-Driven Software Engineering in Practice: Second Edition. Morgan & Claypool, 2017. 168 p. – Режим доступу: URL: (https://www.slideshare.net/mbrambil/modeldriven-software-engineering-in-practice-chapter- 1-introduction?from_action=save)

4. Volter M. et al. Model-Driven Software Development: Technology, Engineering, Management. John Wiley & Sons, 2006

5. Alberto Rodrigues da Silva. Model-driven engineering: A survey supported by the unified conceptual model. Computer Languages, Systems & Structures, Volume 43, October 2015, Pages 139-155.

6. Петрик М.Р. Моделювання програмного забезпечення : науково-методичний посібник / М.Р. Петрик, О.Ю. Петрик – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с.

7. Scenario-Based Task Analysis Practice / Kentaro Go, John M. Carrol // Yamanashi University and Virginia Techhttps: // [Електронний ресурс]. – Режим доступу: URL: www.researchgate.net/publication/228690465_Scenario-based_task_analysis.

8. Marvin Minsky. A Framework for Representing Knowledge. URL: http://web.media.mit.edu/~minsky/papers/Frames/frames.html

9. Novogrudska R.L., Globa L.S., Koval O.V., Senchenko V.R. Ontology model of intelligent modeling system for marine facilities identification. Proceedings of Interna-tional Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), 2017. DOI:10.1109/UkrMiCo.2017.8095426. IEEE Digital Library, 8095426 (Scopus).

10. Novogrudska R.L., Globa L.S., Koval O.V., Senchenko V.R. Ontology for Applications Development. Chapter 2 «Ontology in Information Science»; Book edited by Ciza Thomas. Print ISBN 978-953-51- 3887-7. Published: March 8, 2018. P. 29–53. URL: http://dx.doi.org/10.5772/intechopen.74042

9. Novogrudska R.L., Globa L.S., Koval O.V., Senchenko V.R. Examples of Ontology Model Usage in Engineering Fields. Chapter 3 «Ontology in Information Science»; Book edited by Ciza Thomas. Print ISBN 978-953-51-3887-7. Published: March 8, 2018. P. 54–81. URL: http://dx.doi.org/10.5772/intechopen.74369

10. Senchenko V.R., Koval A.V. The technology of semantic modeling for knowledge management system in environment Protege. «Информационные технологии и безопасность». Материалы XVII международной научно-практической конференции. Київ, 2017. Вип. 17. С. 211–234.

11. An introduction to the owl web ontology language. URL:

http://www.cse.lehigh.edu/~heflin/IntroToOWL.pdf

12. Лавріщева К.М. Програмна інженерія: підручник. Київ: НАН України, 2008. 319 с.

Educational content

5. Methods of mastering the discipline (educational component)

№	***Type of educational activity***	***Description***
Section 1. Introduction to model-oriented design of software systems.		
1	***Lecture 1.*** *Basic approaches to software modeling*	*An overview of software design methods. Basic approaches and concepts of software modeling. Principles of modeling. Simulation objects. Model-oriented approach. The concept of model-oriented approach.*

2	***Lecture 2.*** *Simulation software using simulation language.*	*Software modeling language. UML modeling language model. UML modelling Artifacts Conceptual Rules of Language*
3	***Computer Workshop 1.***	*Objectives: Describe the relevance, object and subject of research, purpose and objectives of the dissertation. Carry out functional and object decomposition of the selected subject area of the aircraft, which is developed in the dissertation. Build a precedent chart and define the characteristics of the precedent chart elements. Build diagrams of classes and objects, processes. Give the definition of a static class. Give a description of the conceptual scheme of the database. Build diagrams of components, locations. Build and describe the architecture of the aircraft.*
Section 2. Model-oriented approach to building a model of the subject area of the aircraft.		
4	***Lecture 3.*** *Building a model of the subject area of the aircraft using a model-oriented approach*	*The use of the semantic model to build a model of the subject area (AP) of the aircraft. Means of building a model of O. The main directions of modeling O. Metadata as an element of the description of the ontological model of OA.*
Section 3. Modeling of the subject area of the software system using semantic modulation of knowledge management.		
5	***Lecture 4.*** *Semantic modeling of knowledge management system construction.*	*Methodology of semantic modeling of the subject area. Determining the depth and scale of the subject area model. Construction of a conceptual model of the ontology of the subject area.*
6	***Lecture 5.*** *Modeling the relationship between the concepts of the subject area and their properties.*	*The mechanism of linking concepts. Formation of relations between instances of classes. Formation of axioms through linking classes. Graphical representation of the ontology model*
7	***Lecture 6.*** *Languages of logical inference.*	*Descriptive logic. Expressiveness of ontology languages. Evaluation of the quality of the ontology model.*
8	***Computer Workshop 2***	*Task: To build an ontological model of PS, which is being developed in the dissertation.*
9	***Lecture 7.*** *Construction of queries to the description of the ontological model of PS.*	*SPARQL query language basics. Mechanisms of formation of SPARQL-queries. Methods and technologies of integration of ontological model with relational data.*

10	***Computer Workshop 3***	*Task: To determine and construct queries to the description of the ontological model of PS, which is developed in the dissertation.*
Section 4. Modeling of PS user activity on the basis of ontological model of software.		
11	***Lecture 8.*** *Scenario-target approach to the activity of the user of the aircraft and construction of scenarios of the user of the aircraft.*	*Scenario-targeted approach to the activities of the user of the aircraft. PS user activity scenario. Object model of the user activity scenario. The model of functioning of the knowledge portal based on the scenario approach. Creating scenarios for user activities in the web environment.*
12	***Lecture 9****. Construction of the user interface of the aircraft on the basis of user activity scenarios and the ontological model of software.*	*PS user interface when creating user activity scenarios. User model of an integrated platform for building user activity scenarios. Using the ontological model of the PS to build the user interface*
13	***Computer Workshop 4***	*Task: To build a fragment of the scenario of user activity and user interface of the aircraft, which is being developed in* *dissertation work, based on the ontological model of Pro.*
Thematic control work		

6. Self-study

The discipline "Model-oriented design of software systems" is based on independent preparation for classroom classes on theoretical and practical topics.

№	***Topic for self-study***	***Hours***	***Reference***
Section 1			
1	*Preparation to the lecture 1*	*1*	*1, p. 4–13; 2, p. 61–66*
2	*Preparation to the lecture 2.*	*2*	*4, p. 18–46,107–146.*
3	*Preparation to the lecture 3.*	*2*	*2, p. 36–53, 59–72*
4	*Preparation to the lecture 4.*	*2*	*3, p. 128–145; 5.*
5	*Preparation to the lecture 5.*	*2*	*3, p. 146–154*
6	*Preparation to the lecture 6.*	*2*	*3, p. 154–160, 177–182*

7	*Preparation to the lecture 7.*	*2*	*3, p. 160–176, 183–193*
8	*Independent elaboration of lecture materials 8*	*6*	*2, p. 12–17, 78-124* *3, p. 226–235.*
9	*Independent elaboration of lecture materials 9*	*6*	*2, p. 126–129.* *3, p. 215–230*
10	*Preparing for a computer workshop 1*	*7*	
11	*Preparing for a computer workshop 2*	*12*	
12	*Preparing for a computer workshop 3*	*12*	
13	*Preparing for a computer workshop 4*	*12*	
14	*Preparation for thematic control work*	*4*	
15	*Exam preparation*	*9*	

Policy and control

7. Policy of academic discipline (educational component)

Attendance at computer workshops can be sporadic and when you need to defend a computer workshop.

Rules of conduct in the classroom: activity, respect for those present, turning off the phones.

Adherence to the policy of academic integrity.

Rules for protecting the work of a computer workshop: the work must be done

according to the task and topic of the dissertation student's dissertation, which must be determined with a piece of software (software system), which is planned to be developed within the dissertation and accordingly which will be the tasks of the computer workshop.

The rules for assigning incentive and penalty points are as follows.

Incentive points are awarded for:

- accurate and complete answers during the surveys based on lecture materials. During the semester there is a blitz poll on the topics of past lectures. Maximum number of points for the blitz poll: 1 ba x 7l. = 7 points.

- Creative approach in the implementation of computer workshops. Maximum number of points for all works: 4 points x 3 computers. practice. + 11 points x 1 computer. practice. = 23 points.

Penalty points are awarded for:

- plagiarism (program code does not correspond to the task variant, identity of the program code among different works) in the works of the computer workshop: -5 points for each attempt.

8. Types of control and rating system of assessment of learning outcomes (RSO)

During the semester, graduate students perform 4 computer workshops. Maximum number of points for each computer workshop: 15 points.

Points are awarded for:

- quality of computer workshop: 0-5 points;

- answer during the defense of the computer workshop: 0-5 points; - timely submission of work to the defense: 0-5 points.

Criteria for evaluating the quality of performance:

5 points - the work is done qualitatively, in full;

3 points - the work is done in full, but contains minor errors; 1 point - the work is not performed in full, or contains significant errors.

Response evaluation criteria:

5 points - the answer is complete, well-argued;

3 points - in general the answer is correct, but has shortcomings or minor errors; 0 points - no answer or the answer is incorrect.

Criteria for assessing the timeliness of submission of work to the defense:

5 points - the work is submitted for defense no later than the specified period; 1 point - the work is submitted for defense later than the specified period.

Maximum number of points for the implementation and defense of computer workshops: 15 points × 4 computers. practice. = 60 points.

During the semester, lectures are conducted on the topic of the current lesson. Maximum number of points for the survey that can be obtained during the semester: 2 points.

The task for the thematic test consists of 2 theoretical questions. The answer to each theoretical question is evaluated by 5 points.

Criteria for evaluating each theoretical question of the test: 5 points - the answer is correct, complete, well-argued;

4 points - the answer is correct, detailed, but not very well reasoned; 3 points - in general the answer is correct, but has shortcomings;

2 points - there are minor errors in the answer;

1 point - there are significant errors in the answer;

0 points - no answer or the answer is incorrect.

Maximum number of points for the module test: 5 points × 2 theoretical questions = 10 points.

The rating scale for the discipline is equal to:

Rs = Rkom.prakt + Ropituv + Rtv. approach + RTKR = 60 points + 7 points + 23 points + 10 points = 100 points.

Calendar control: conducted twice a semester to monitor the current state of compliance with the requirements of the syllabus.

At the first attestation (8th week) the student receives "credited" if his current rating is not less than 50% of the maximum number of points that a student can receive before the first attestation.

At the second attestation (14th week) the student receives "credited" if his current rating is not less than 50% of the maximum number of points that a student can receive before the second attestation.

Semester control: exam.

Table of correspondence of rating points to grades on the university scale:

Number of pints	*Grades*
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactorily
64-60	Sufficiently
Less 60	Unsatisfactory
Admission conditions are not met	Not allowed

Work program of the discipline (Syllabus):

Developed by Ph.D., Associate Professor Koval Oleksandr Vasyliovych

Approved *by department __________ (protocol № ___ from ____________)*

Resolved *by Methodical commission of the faculty (protocol № __ from _______)*