Formal Methods

Course description sheet

Basic information

Field of study Computer Science and Intelligent Systems **Major** Artificial Intelligence and Data Analysis **Organisational unit** Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering **Study level** Second-cycle (engineer) programme **Form of study** Full-time studies **Profile** General academic	**Didactic cycle** 2023/2024 **Course code** EISIAIDAS.Ili1K.c16b27e40c89f8a46358b7431d6e22fd.23 **Lecture languages** english **Mandatoriness** Obligatory **Block** Core Modules **Course related to scientific research** Yes

Course coordinator	Tomasz Szmuc
Lecturer	Tomasz Szmuc, Wojciech Szmuc

Period Semester 1	**Method of verification of the learning outcomes** Exam **Activities and hours** Lectures: 28 Laboratory classes: 28	**Number of ECTS credits** 4

Goals

C1	To educate on the role of formal methods in the rigorous development of software.
C2	To provide knowledge in modelling and analysis using Petri nets formalisms and train the related skils.
C3	To transfer knowledge on modelling and analysis using process algebras and train the related skils.
C4	To provide knowledge about formal modelling systems and their practical use.

Course's learning outcomes

Code	Outcomes in terms of	Learning outcomes prescribed to a field of study	Methods of verification
Knowledge – Student knows and understands:			
W1	Student has deepened knowledge of the main formal modelling languages.	ISI2_W01, ISI2_W02, ISI2_W03	Execution of laboratory classes, Test, Examination, Completion of laboratory classes
W2	Student has a basic knowledge of formal verification systems.	ISI2_W02, ISI2_W04, ISI2_W05	Execution of laboratory classes, Test, Examination, Completion of laboratory classes
W3	Student has a basic knowledge of software tools for modelling and verification of IT systems using formal methods.	ISI2_W05, ISI2_W06, ISI2_W07	Execution of laboratory classes, Test, Examination, Completion of laboratory classes
Skills – Student can:			
U1	Student is able to use formal methods for the analysis and design of IT systems.	ISI2_U01, ISI2_U02, ISI2_U03, ISI2_U04, ISI2_U05	Execution of laboratory classes, Test, Examination, Completion of laboratory classes
U2	Student is able to use software tools supporting formal modelling and verification.	ISI2_U06, ISI2_U07, ISI2_U08, ISI2_U09	Execution of laboratory classes, Test, Examination, Completion of laboratory classes
Social competences – Student is ready to:			
K1	Student understands the need for providing the society with information and opinions on the role of formal methods and their impact on the software quality.	ISI2_K01, ISI2_K02	Execution of laboratory classes, Test, Examination, Completion of laboratory classes

Program content ensuring the achievement of the learning outcomes prescribed to the module

Formal methods in computer science: mathematical modelling of processes, formal verification of properties (correctness). Role of the formal methods in rigorous software development. The lecture starts with formal modelling using Petri nets: place-transition Petri nets, hierarchical and coloured Petri nets, time aspects in Petri nets. Modelling behaviour using process algebras: CCS (Calculus of Communicating Systems), CSP (Communicating Sequential Processes), LOTOS belong to the next group of description formalisms. Modelling examples using the above formalisms are constructed and tested during laboratory classes.

Student workload

Activity form	Average amount of hours* needed to complete each activity form
Lectures	28
Laboratory classes	28

Realization of independently performed tasks	10
Preparation for classes	10
Participation in classes / practical placement	28
Examination or final test/colloquium	2
Student workload	**Hours** 106
Workload involving teacher	**Hours** 56

* hour means 45 minutes

Program content

No.	Program content	Course's learning outcomes	Activities
1.	1. Formal methods in computer science: mathematical modelling of processes, formal verification of properties (correctness). Role of the formal methods in rigorous software development. 2. Formal modelling using Petri nets: place-transition Petri nets, hierarchical and coloured Petri nets, time aspects in Petri nets. 3. Modelling behaviour using process algebras: CCS (Calculus of Communicating Systems), CSP (Communicating Sequential Processes), LOTOS. 4. Time automata for modelling real-time systems.	W1, W3, U1, K1	Lectures
2.	1. An overview of Temporal Logics. 2. Specification of required properties using Temporal Logic.	W2, W3, U2, K1	Lectures
3.	1. Practical classes in modelling processes using Petri nets description and the related tool - CPN Tools. 2. Practical classes in modelling processes using LOTOS/LNT specifications and CADP package.	W1, W3, U1, U2, K1	Laboratory classes
4.	1. Practical classes in using CADP for proving properties of LOTOS/LNT specifications.	W2, W3, U2, K1	Laboratory classes

Extended information/Additional elements

Teaching methods and techniques:

Lectures, Discussion, E-learning, Problem based learning

Activities	Methods of verification	Credit conditions
Lectures	Examination	Final positive grade of the laboratory classes and positive grade received from exam are required to complete the subject.
Lab. classes	Execution of laboratory classes, Test, Completion of laboratory classes	Positive grades of all laboratory tasks.

Conditions and the manner of completing each form of classes, including the rules of making retakes, as well as the conditions for admission to the exam

1. Lecture - a positive result of the exam on the applicable date. The exam consists of a written and oral part. Written parts are exempted for those who received at least a grade of 4.0 from laboratory exercises on the first assessment date.
2. Laboratory exercises: positive evaluation of all exercises and passing of partial colloquia.

Method of determining the final grade

The course grade is the average of laboratory and exam grades. In the case of the correction exams, grades of all terms are taken into account.

Manner and mode of making up for the backlog caused by a student justified absence from classes

The way to compensate for the delays in the classes is independent reading of the textbook indicated by the teacher, as well as the completion and passing of all outstanding work, tests and examines.

Prerequisites and additional requirements

Basic knowledge of:

- discrete mathematics, in particular relations and graphs
- finite automata theory
- propositional logic and first order logic

Rules of participation in given classes, indicating whether student presence at the lecture is obligatory

Lectures: attendance is not obligatory; however, students are encouraged for active participation, asking questions and discussion.
Laboratory classes: students perform tasks according to guidance and tutorials delivered by teaching assistant. Interim verifications of knowledge by colloquiums are applied.

Literature

Obligatory

1. Murata T.: Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE, Vol. 77, pp. 541-580. 1989
2. Jensen K.: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, Vol. I-III, Springer Verlag, 1995/96
3. Turner K.J.: The formal specification language LOTOS. A course for users
4. Behrman G., David A., Larsen K.G.: A tutorial for UPAAL 4.0
5. Jensen K., Kristensen Lars M., Coloured Petri Nets. Modelling and validation of Concurrent Systems, Springer Verlag, 2009

Optional

1. Aceto L., Ingofsdottir A., Larsen K.G., Srba J.: Reactive Systems: Modelling, Specification and Verification. Cambridge University Press, 2007

Scientific research and publications

Research

1. Rigorous Development of Cyber-Physical Systems (CPS) supported by mathematical modelling and formal verification.
2. Analysis and verification of real-time systems using formal methods

Publications

1. Szmuc T., Szpyrka M.: Formal Methods - Support or Scientific Decoration in Software Development. Proceedings of the 22nd International Conference "Mixed Design of Integrated Circuits and Systems". June 27, 2015, Toru, Poland. pp. 24-32 (Invited Paper)
2. Szmuc W., Szmuc T.: Modelling UML Object Event Handling with Petri Nets. Towards improvement of embedded systems analysis and design. Proceedings of the 23rd International Conference "Mixed Design of Integrated Circuits and Systems". June 23-25, 2016 Łódź. pp.454-457
3. Szmuc W., Szmuc T.: Towards Embedded Systems Formal Verification. Translation from SysML into Petri Nets. Proceedings of the 25th International Conference "Mixed Design of Integrated Circuits and Systems". June 21-23, 2018. Gdynia. pp. 420-423
4. Szmuc T., Szmuc W.: Rigorous Development of Embedded Systems Supported by Formal Tools. Proceedings of the 27th International Conference "Mixed Design of Integrated Circuits and Systems". Łódź, June 25-27, 2020. pp. 272-276
5. Mrówka R., Szmuc T.: UML Statecharts compositional semantics in LOTOS. ISPDC 2008 Proceedings of the 7th International Symposium on Parallel and Distributed Computing : 1–5 July 2008, Krakow, Poland / eds. Marek Tudruj. — Los Alamitos, California; Washington; Tokyo : CPS Conference Publishing Services, IEEE Computer Society, cop. 2008. ISBN 978-0-7695-3472-5. pp. 459–463

Learning outcomes prescribed to a field of study

Code	Content
ISI2_K01	Absolwent jest gotowy do krytycznej oceny odbieranych treści, potrafi myśleć i działać w sposób kreatywny i przedsiębiorczy, rozumie w odniesieniu do tego specyfikę systemów inteligentnych, rozumie potrzebę i zna możliwości podnoszenia kompetencji swoich i innych osób, potrafi współdziałać i pracować w grupie, przyjmując w niej różne role, ma świadomość odpowiedzialności za pracę własną i za wspólnie realizowane zadania
ISI2_K02	Absolwent jest przygotowany do pełnienia różnych ról zawodowych, ma świadomość roli społecznej absolwenta uczelni technicznej, rozumie potrzebę formułowania i przekazywania społeczeństwu informacji i opinii dotyczących osiągnięć informatyki, w szczególności systemów innowacyjnych i inteligentnych, wagi profesjonalnego zachowania i przestrzegania zasad etyki zawodowej, prawidłowo identyfikuje i rozstrzyga dylematy związane z wykonywaniem zawodu
ISI2_U01	Absolwent potrafi pozyskiwać informacje z literatury i innych źródeł, korzystać z uzyskanych informacji, uwzględniając nieprzewidywalne i nietypowe warunki, szczególnie dla systemów inteligentnych, oraz dokonywać ich interpretacji i oceny, wyciągać wnioski i formułować opinie, a także określać kierunki dalszego uczenia się
ISI2_U02	Absolwent posługuje się językiem obcym w stopniu wystarczającym do porozumiewania się, czytania ze zrozumieniem literatury fachowej, a także wygłoszenia krótkiej prezentacji na temat realizacji zadania projektowego lub badawczego, umie poprowadzić debatę, potrafi komunikować się na tematy specjalistyczne z różnym kręgiem odbiorców
ISI2_U03	Absolwent potrafi formułować i weryfikować hipotezy, analizować nieprzewidywalne warunki związane z problemami inżynierskimi i prostymi zagadnieniami badawczymi, szczególnie z uwzględnieniem specyfiki systemów inteligentnych, potrafi opracować specyfikację projektową złożonego oprogramowania systemu inteligentnego, z uwzględnieniem aspektów prawnych oraz innych aspektów pozatechnicznych, z uwzględnieniem norm i standardów, zaprojektować oprogramowanie adekwatnie do specyfikacji wymagań, opracować szczegółową dokumentację wyników, a także przygotować i i przedstawić prezentację oraz przeprowadzić dyskusję wyników
ISI2_U04	Absolwent potrafi pracować zarówno indywidualnie i kolektywnie, opracować i zrealizować harmonogram prac, szczególnie systemów innowacyjnych i obarczonych dużym ryzykiem, przeprowadzać symulacje oraz prace walidacyjne oraz kierować małym zespołem w sposób zapewniający realizację zadania w założonym terminie
ISI2_U05	Absolwent potrafi wykorzystać poznane metody, algorytmy i modele do tworzenia różnego rodzaju programów o charakterze użytkowym i naukowym, z uwzględnieniem specyfiki specjalności oraz systemów inteligentnych
ISI2_U06	Absolwent potrafi wykorzystać znane metody i struktury danych w budowie systemu inteligentnego, a także formalizować metody z wykorzystaniem zaawansowanych technik algorytmicznych oraz analizować i optymalizować ich własności w tym złożoność
ISI2_U07	Absolwent potrafi dokonać identyfikacji i analizy wymagań oraz analizy ryzyka związanych z budową systemu inteligentnego, projektować oprogramowanie zgodnie z wybraną metodyką, dobierać modele i procesy wytwarzania i walidacji oprogramowania, a także skonfigurować system inteligentny, w szczególności w zakresie funkcji i narzędzi związanych ze specjalnością
ISI2_U08	Absolwent posługuje się technikami i językami programowania, szczególnie z uwzględnieniem specyfiki systemów inteligentnych, potrafi ocenić przydatność różnych paradygmatów i związanych z nimi środowisk programistycznych do rozwiązywania różnego typu problemów; potrafi czytać ze zrozumieniem, pisać, uruchamiać i weryfikować programy zapisane z użyciem różnych paradygmatów programowania, z uwzględnieniem specyfiki specjalności
ISI2_U09	Absolwent potrafi dogłębnie ocenić przydatność i korzystać z dostępnych bibliotek i pakietów dla systemów inteligentnych oraz komponentów oprogramowania oraz narzędzi z uwzględnieniem specyfiki specjalności, ze szczególnym uwzględnieniem specyfiki systemów inteligentnych, a także porównać istniejące rozwiązania ze względu na zadane, kryteria użytkowe i ekonomiczne oraz wskazać możliwości ich ulepszenia, w szczególności potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć w zakresie informatyki systemów inteligentnych
ISI2_W01	Absolwent ma poszerzoną i ugruntowaną wiedzę i potrafi formułować i rozwiązywać zarówno typowe, jak i nietypowe, w tym złożone, problemy w sposób innowacyjny w oparciu o znajomość w zakresie przedmiotów ścisłych, w szczególności zadania z zakresu informatyki systemów inteligentnych

Code	Content
ISI2_W02	Absolwent ma pogłębioną wiedzę w zakresie specyficznych metod i złożonych struktur danych związanych ze specjalnością, w szczególności w odniesieniu do nietypowych i innowacyjnych problemów systemów inteligentnych, a także wiedzę w odniesieniu do zastosowań metod obliczeniowych oraz wybranych zagadnień systemów inteligentnych
ISI2_W03	Absolwent ma poszerzoną wiedzę w zakresie wybranych paradygmatów języków i technik programowania z uwzględnieniem specyfiki specjalności oraz systemów inteligentnych
ISI2_W04	Absolwent ma pogłębioną wiedzę w zakresie cyklów życia obiektów i systemów informatycznych oraz w zakresie procesów wytwórczych inżynierii oprogramowania, z uwzględnieniem specyfiki specjalności, w szczególności w zakresie stosowania narzędzi i systemów informatycznych, etapów i metod projektowania z zakresu specjalność i systemów inteligentnych
ISI2_W05	Absolwent ma szczegółową wiedzę w zakresie współczesnych narzędzi implementacyjnych, środowisk programistycznych, technik integracji systemów inteligentnych związanych ze specjalnością
ISI2_W06	Absolwent ma wiedzę o tendencjach rozwojowych informatyki systemów inteligentnych oraz ma wiedzę niezbędną do rozumienia pozatechnicznych uwarunkowań działalności inżynierskiej zorientowanej na systemy inteligentne
ISI2_W07	Absolwent ma wiedzę w zakresie prowadzenia działalności gospodarczej, ochrony i zarządzania własnością intelektualną oraz prawa patentowego, w szczególności w odniesieniu do systemów cechujących się innowacyjnością