

1 1.1 Title of module (GER / ENG) Model Driven Development	1.2 Short description (optional) MDD	1.3 Module code (from HIS-POS)			
2 2.1 Cycle of module: <input checked="" type="checkbox"/> each summer semester, <input type="checkbox"/> each winter semester other cycle, namely:	2.2 Duration of module <input checked="" type="checkbox"/> 1 semester <input type="checkbox"/> 2 semesters				
3 3.1 Module offered in the following study programme(s): Master Business Informatics full time part time- summer part time-winter	3.2 Compulsory (Pf), compulsory elective (WPf), elective (W) Pf	3.3 Recommended semester: 2 3 4			
4 Workload					
Contact hours (e.g. lecture, seminar, practical course, practical phase/internship, group work, project work, case study, simulation game, credited tutorial (additional lines possible))	Teaching methods Seminar Practical course	Weekly teaching hours ("Semesterwochenstunde") per teaching method 2 1	Hours in semester per teaching method 1 weekly teaching hour per semester can be indicated as 15 hours, i.e. 1 weekly teaching hour = 1 hour x 15 semester weeks 30 15	Workload in total Workload in hours sum contact hours and self-study in hrs. ECTS (credit points) generally 30 hrs. = 1 credit point; only full numbers allowed 150 5	
	Sums	Sum contact hours in weekly teaching hours ("Semesterwochenstunden")	Sum contact hours in hrs.		
Self-study (e.g. tutorial, preparation, follow-up work, preparation for assignments and homeworks, research etc.)	Preparation/follow-up work		75		
	Preparation for assignments		30		
	Sums		Sum self-study in hrs		
5 5.1 Intended learning outcomes (What should students be able to do after having accomplished the module? Does the module provide the opportunity to acquire soft skills in addition to professional knowledge? For which other modules and prospective tasks in the labour market are the acquired knowledge and skills relevant?)					
After successful participation you are able to...					
<ul style="list-style-type: none"> • specify systems and processes using formal method, in particular Petri Nets, • analyze or simulate your models in order to prove certain properties, • design your custom modeling language, • specify executable model transformation rules, • generate code from object-oriented domain models, • choose an appropriate modeling technique. 					
The course will improve your methodological expertise in...					
<ul style="list-style-type: none"> • analyzing problems and finding appropriate solutions • applying formal methods and languages for the specification of business logic and processes • presenting your findings to others, e.g., with the help of audiovisual media 					
Attendees are also trained in soft skills such as...					
<ul style="list-style-type: none"> • academic English language skills • time management • self-organization 					

- self-assessment along regular exercises

5.2 Course content

Part I - Models for Reasoning:

- Basic Concepts of Petri Nets
- Application of Petri Nets
- Extending Petri Nets with Color
- Extending Petri Nets with Time
- CPN Language and Tools
- Hierarchical Petri Nets
- Analysis of Petri Nets
- Simulation of Petri Nets

Part II - Models for Code Generation:

- Modeling Languages
- String Grammars
- Language Development with Xtext
- Unidirectional Model Transformations
- Bidirectional Model Transformations

→ details can be found in course syllabus, recommended study plan etc.

Module Description

5 5.3 Short information about module (This paragraph [max. 250 characters] will be published on the website of FH Münster to support persons interested in studying at FH Münster to choose the appropriate study programme. Please focus on the main intended learning outcomes and course content, ideally also comprising information about the relevance of the module for the further course of study and the labour market. Please formulate whole sentences, address your (prospective) students directly and avoid technical terms.)

You will learn how to specify processes and software with the help of formal methods in such a way that your models are applicable to automated analysis, simulation, or code generation.

6 6.1 Prerequisites (*formal*: examination of module XY has to be passed or similar *content-wise*: *module XY should have been attended, the following knowledge and skills should have been acquired: ...*)

6.2 Requirements for awarding credit points (e.g. passing final examination, successful accomplishment of assignments in the course of study, regular active participation)

Passing final examination

6.3 Type and extent of examination (e.g. written exam, oral exam, term paper, presentation, portfolio, duration of examination in minutes)

paper / presentation (audiovisual tutorial)

6.4 Requirements for admission to examination



6.5 Weighing of module grade when calculating final grade

see examination regulations for aforementioned study programmes (line 3).*

*You will find the examination regulations of all study programmes in the official announcements of the FH Münster: https://www.fh-muenster.de/hochschule/aktuelles/amtliche_bekanntmachungen/index.php?p=2,7.

7 7.1 Languages used in the module:

German English others, namely:

7.2 Contact person for module:

Prof. Dr. Sebastian Thöne

7.3 Professors (optional)

Prof. Dr. Sebastian Thöne

7.4 Maximum number of participants (optional)

7.5 Further information (optional) (e.g. literature recommendations, other persons involved, etc.)