



الجامعة العربية المفتوحة
Arab Open University

Module specification

1. Factual information			
Module title	TM295 : System Modelling	Level	2
Module tutor	TBA	Credit value	10
Module type	Taught	Notional learning hours	3

2. Rationale for the module and its links with other modules

System modelling is used as an essential part of the software development process; it also referred to as software modelling in this context. Models are built and analysed prior to the implementation of the system, and are used to direct the subsequent implementation. Modelling can be defined as considering the system from different views (or perspectives) in order to provide a better understanding of it. These views include (among others) requirements models, static models, and dynamic models of the software system. These different views can be further developed, understood, and communicated through the use of graphical modelling languages such as the Unified Modelling Language (UML).

The module focuses on how to use adequate models to express software at all levels of development; from the initial specification to implementation, with a special attention paid to UML.

3. Aims of the module

This module aims to introduce students to the software development process in general with emphasis on the software modelling and analysis phase. The unified modelling language is used throughout the module to illustrate the different models.

4. Pre-requisite modules or specified entry requirements

MT132

5. Intended learning outcomes	
A. Knowledge and understanding	Learning and teaching strategy
<p>Upon completing this module, students will be able to:</p> <p>A1. Describe the software engineering lifecycle and in particular the role of analysis and design phase in the lifecycle</p> <p>A2. Discuss the different software lifecycle models including (e.g., waterfall, agile)</p> <p>A3. Define the requirements elicitation and structuring</p> <p>A4. Explain how to use the UML models to develop and document software analysis and design artefacts.</p>	<ul style="list-style-type: none"> • The different Learning outcomes are covered in the study materials which are covered during the face-to-face meetings. • The LOs will be assessed through a number of formative and summative assessments <ul style="list-style-type: none"> ○ Formative assessments include activities within module components; Activities are designed to enable students to apply the concepts that have been taught, or to explore issues that extend students' knowledge and skills. These frequently involve reading parts of the module book or other documents. Some of the activities are done during the module meetings and others are done at students' own time and discussed with each other's and the module tutor during the module meetings. Formative assessments are also in the form of feedback on the tutor marked assignment (TMA) provided by the module tutor. ○ Summative assessments in the form of continuous assessment provided by the midterm assessment (MTA) and the tutor marked assessment (TMA) and the final exam
B. Cognitive skills	Learning and teaching strategy
<p>Upon completing this module, students will be able to:</p> <p>B1. Specify, analyse and organise requirements for a software product</p> <p>B2. Model, analyse and validate software requirements using UML and set-theoretic notations</p>	<ul style="list-style-type: none"> • The module include a number of activities that provides students with the mentioned cognitive skills and at the same time constitute a formative assessment of these skills. In addition, the TMA assesses and validates the cognitive skills related learning outcomes (LOs). The other assessments

B. Cognitive skills	Learning and teaching strategy
<p>B3.Apply appropriate UML design patterns and notations to the design of components of a product</p>	<p>(MTA and Final exam) are also means for validation of the Los.</p>

C. Practical and professional skills	Learning and teaching strategy
<p>Upon completing this module, students will be able to:</p> <p>D1. Apply an appropriate software engineering process and tools to the task of structuring, modelling and validating requirements for a software product</p> <p>D2. Work independently, planning, monitoring, reflecting on and improving your own learning and working practices</p> <p>D3. work in a group, communicating computing ideas effectively in speech and in writing</p> <p>D4. Find, assess and apply information from a variety of sources, using information technology where necessary, in a number of assignments, including at least one significant piece of work</p>	<ul style="list-style-type: none"> • The module material exposes students to various case studies, which are analyzed and discussed as part of the module activities during the face-to-face meetings (some of them are done at student's own time but discussed during the tutorial sessions). Some activities are performed in groups to improve the team work skills. • The practical and professional skills are mainly assessed in the TMA.

D Key transferable skills	Learning and teaching strategy
<p>Upon completing this module, students will be able to:</p> <p>D1. Design software systems</p> <p>D2. Use modern software tools, both within and outside your workplace</p> <p>D3. Communicate effectively about software modelling and design</p> <p>D4. Be able to learn independently from third-party materials, in order to keep up to date in software engineering in general and software modelling in particular</p>	<ul style="list-style-type: none"> • Some of the module activities require students to do scientific search to collect, assess and synthesize information from various reliable resources. These activities are mainly done at student's own time but under the guidance of the module tutor. • The transferrable skills are mainly assessed in the TMA (formative and summative).

6. Indicative content.
<ul style="list-style-type: none"> • Software development lifecycle: from problems (requirements) to solutions (programs) • Software process models, e.g. waterfall, agile • Requirements elicitation: structuring narrative requirements • Introduction to Software Modelling • Introduction to OO-Design and Analysis • Introduction to UML • UML Common Notation Conventions • Use Cases • Class Diagrams • Collaboration Diagrams • Interaction Sequence Diagrams • Activity Diagrams • State Diagrams • Implementation Diagrams

6. Indicative content.

- Code Generation from UML Models
- Model Validation and Testing
- Components and Frameworks with UML 3 h

7. Assessment strategy, assessment methods and their relative weightings

MTA 30%
TMA 20%
Final exam 50%

8. Mapping of assessment tasks to learning outcomes

Assessment tasks	Learning outcomes														
	A1	A2	A3	A4	B1	B2	B3	C1	C2	C3	C4	D1	D2	D3	D4
TMA					X	X	X	X	X	X	X	X	X	X	X
MTA	X	X	X	X			X							X	
End of Semester Exam	X	X	X	X	X	X	X							X	

9. Teaching staff associated with the module**Name and contact details**

TBA

10. Key reading list				
Author	Year	Title	Publisher	Location
Ian Sommerville	2016	Software Engineering – 10th edition	Pearson	
Simon Bennett, Ray Farmer and Steve Mcrobb	2010	Object-Oriented Systems Analysis and Design Using UML – 4 th edition	McGraw-Hill	

11. Other indicative text (e.g. websites)
References Software Modelling and design module specifications, Southampton university