



Guide pédagogique

Module « Ingénierie dirigée par les modèles »

Option IL – 9.4 (4 crédits ECTS)

Place du module et enjeux

L'ingénierie dirigée par les modèles (IDM) regroupe un ensemble de techniques avancées de développement logiciel qui remplace, autant que possible, l'écriture de code par l'écriture de modèles. En se concentrant sur des productions plus abstraites, l'ingénierie dirigée par les modèles permet de concevoir des logiciels indépendamment de la technologie (langage de programmation, *framework* spécifique) et, ainsi, de concentrer l'activité d'ingénierie sur des artefacts abstraits pérennes. Des techniques génératives permettent ensuite de projeter les modèles dans un espace technologique donné. Comme conséquence, maintenir les logiciels, décliner des gammes de produits en adaptant certaines parties et faire évoluer ces produits devient plus simple et permet d'atteindre une meilleure qualité.

Ce module pose les fondements de l'ingénierie dirigée par les modèles. Y sont présentés la méta-modélisation et la transformation de modèles. Les bonnes pratiques d'ingénierie et de développement centré architectures y sont aussi abordés. Enfin, le module comprend un cours inversé qui étend la culture des étudiants sur les sujets et paradigmes actuels de l'ingénierie du logiciel.

Teaching guide and syllabus

« Model Driven Engineering » module

SE options – 9.4 (4 ECTS credits)

Subject matter importance and associated issues

Model driven engineering (MDE) groups advanced software development techniques that try to replace, as far as possible, the coding activity by a modeling activity. Producing more abstract products helps design software in a technology-independent manner (staying away from specific programming languages or frameworks) thus focusing the engineering activity on long-lasting abstract artifacts. Generative techniques then translate models into a given technological space. As a consequence, maintaining software applications, deriving software products lines by adapting specific parts of the software or managing software evolution becomes easier and leads to better quality.

This module defines the foundation of model driven engineering. Meta modeling and model transformations are studied. Software development best practices and architecture-centered development also are presented. The module also comprises an inverted course to broaden student awareness of the latest subjects and paradigms of software engineering.

Responsable : Christelle Urtado

Téléphone : 04 34 24 62 89

Courriel : christelle.urtado@mines-ales.fr



IMT Mines Alès
École Mines-Télécom

ENSEIGNEMENTS ACADÉMIQUES	Volume horaire	Détail des coefficients	Crédits
Ingénierie dirigée par les modèles	50 h		
○ Méta-modélisation et transformations de modèles	15	1	4
○ Bonnes pratiques et développement centré-architecture	20	1	
○ Sujets et paradigmes actuels en ingénierie du logiciel	15	1	

Matière 1

Titre de la matière : Méta-modélisation et transformation de modèles	
Code : 2IA-il 9.4.1	Titre du module : « Ingénierie dirigée par les modèles »
Semestre : S9	Cursus de rattachement : Département 2IA, option IL

Heures présentiel	Heures total	Cours	TD	TP	Projet	Contrôles	Travail personnel	Coef /module	ECTS
15	25	6			9		10	1	1,3

Résumé	Ce module présente les concepts de méta-modélisation et de méta-programmation (conception de langages de modélisation et de programmation), les techniques de transformation de modèles et l'utilisation de ces concepts pour définir de nouveaux langages et étendre les outils de développement logiciel.
---------------	---

Responsable	Christelle Urtado – LGI2P/IMT Mines Alès
Équipe enseignante	Clémentine Nebut – Univ. Montpellier

Mots-clés	Métamodèle, UML, MOF, transformation de modèles, DSL, QVT, Ecore
Prérequis	Module 2IA 8.4 Ingénierie Logicielle

<p>Contexte et objectif général :</p> <p>L'objectif de cette matière est de faire découvrir aux étudiants les notions de méta-modélisation et de transformation de modèles comme une alternative moderne à l'ingénierie classique. Il est aussi de faire entrevoir les possibilités d'extension des langages et outils par la méta-programmation ; de définir et outiller un langage de modélisation spécifique (Domain Specific Language) et de mettre en œuvre des transformations de modèles.</p>
<p>Programme et contenu :</p> <p>Méta-modélisation Exemple du méta-modèle d'UML Transformation de modèles Définition de langages spécifiques Programmation et extension d'outils de développement par utilisation de méta-modèles standard : MOF, Ecore</p>
<p>Méthode et organisation pédagogique :</p> <p>Les enseignements sont prévus pour 30 étudiants. Les manipulations seront réalisées sur les ordinateurs personnels de ces derniers.</p> <p>Le découpage est prévu comme suit :</p> <ul style="list-style-type: none"> - 6h de cours - 9h de projet
<p>Acquis d'apprentissage visés :</p> <p>Comprendre la notion de métamodèle Connaître l'exemple du métamodèle d'UML Comprendre les tenants et aboutissants de la création de nouveaux langages Savoir définir le métamodèle d'un langage simple Comprendre l'utilité des transformations de modèles et savoir les mettre en œuvre</p>
<p>Évaluation : Notation sur projet (coef 1)</p>
<p>Retour sur l'évaluation fait à l'élève : fiche d'évaluation des projets mise à disposition 3 semaines maximum après la dernière séance.</p>
<p>Support pédagogique et références :</p> <p>1 Polycopié composé des supports de présentation</p>

Matière 2

Titre de la matière : Bonnes pratiques et développement centré-architecture	
Code : 2IA-il 9.4.2	Titre du module : « Ingénierie dirigée par les modèles »
Semestre : S9	Cursus de rattachement : Département 2IA, option IL

Heures pré-sentiel	Heures total	Cours	TD	TP	Projet	Contrôles	Travail personnel	Coef /module	ECTS
20	30	8			12		10	1	1,4

Résumé	Ce cours présente dans un premier temps les bonnes pratiques de développement logiciel qui permettent de disposer, en continu, d'un logiciel de meilleure qualité prenant en compte les derniers développements. Dans un deuxième temps, il illustre le développement d'architectures logicielles à base de composants.
---------------	---

Responsable	Christelle Urtado – LGI2P/IMT Mines Alès
Équipe enseignante	Christelle Urtado – LGI2P/IMT Mines Alès Sylvain Vauttier – LGI2P/IMT Mines Alès Doctorants en appui pour l'encadrement du projet

Mots-clés	Qualité des logiciels, tests, intégration continue, composant, architecture, langage de description d'architecture
Prérequis	Module 2IA 8.4 Ingénierie Logicielle

Contexte et objectif général :	<p>Ce cours présente dans un premier temps les bonnes pratiques de développement logiciel avec, notamment, les métriques et qualités du logiciel, l'écriture de tests et la chaîne d'intégration continue. Les méthodes agiles sont aussi évoquées.</p> <p>Dans un deuxième temps, il illustre le développement centré architecture en introduisant les qualités attendues telles que la modularité et le découplage permettant de mettre en oeuvre une ingénierie par réutilisation. Les composants logiciels et les architectures logicielles sont présentés ainsi que l'impact de ces concepts sur le cycle de développement (conteneurs, annuaires).</p>
---------------------------------------	--

Programme et contenu :	<p>Partie 1 : Bonnes pratiques de développement Qualité et test de logiciel Méthodes de développement agiles Modularisation d'applications et bases de l'intégration continue (automatisation de tâches)</p> <p>Partie 2 : Développement centré architecture Patrons d'architectures logicielles : Modularité, découplage et réutilisation Composants logiciels et architecture à base de composants Langages de description d'architectures</p>
-------------------------------	---

Méthode et organisation pédagogique :	<p>Les enseignements sont prévus pour 30 étudiants. Les TP seront réalisés sur les ordinateurs personnels de ces derniers.</p> <p>Le découpage est prévu comme suit :</p> <ul style="list-style-type: none"> - 8h de cours - 12h de projet
--	--

Acquis d'apprentissage visés :	<p>Comprendre les enjeux des bonnes pratiques de développement logiciel</p> <p>Savoir utiliser les outils de mesure de la qualité, écrire des tests, modulariser une application et mettre en place une chaîne d'intégration continue</p> <p>Être sensibilisé aux méthodes de développement agiles</p> <p>Comprendre les enjeux du développement centré architecture</p> <p>Savoir modéliser une architecture de logiciel à base de composants en respectant des patrons d'architecture</p>
---------------------------------------	---

Évaluation :	Notation sur projet (coef 1)
---------------------	------------------------------

Retour sur l'évaluation fait à l'élève :	fiche d'évaluation des projets mise à disposition 3 semaines maximum après la date de rendu des projets
---	---

Support pédagogique et références :	1 Polycopié composé des supports de présentation
--	--

Matière 3

Titre de la matière : Sujets et paradigmes actuels en ingénierie du logiciel	
Code : 2IA-il 9.4.3	Titre du module : « Ingénierie dirigée par les modèles »
Semestre : S9	Cursus de rattachement : Département 2IA, option IL

Heures pré-sentiel	Heures total	Cours	TD	TP	Projet	Contrôles	Travail personnel	Coef /module	ECTS
15	25	15					10	1	1,3

Résumé	<p>Ce cours a pour objectif d'amener les étudiants à élargir leur culture sur différents sujets et paradigmes actuels en ingénierie du logiciel. Des thèmes d'étude sont proposés, chacun illustré par un ou deux documents (articles choisis). Les étudiants se répartissent les thèmes d'étude et en proposent des mini-cours à destination des autres étudiants. Chaque mini-cours donne lieu à une discussion (de type atelier). Les étudiants sont évalués tant sur leur production que sur leurs réactions aux cours de leurs collègues.</p>
---------------	--

Responsable	Christelle Urtado – LGI2P/IMT Mines Alès
Équipe enseignante	Christelle Urtado – LGI2P/IMT Mines Alès

Mots-clés	Cours inversé, sujets et paradigmes actuels : agilité, extreme programming, programmation orientée aspects, génie logiciel empirique, évolution du logiciel, découplage et inversion de contrôle, ...
Prérequis	Module 2IA 8.4 Ingénierie Logicielle

<p>Contexte et objectif général :</p> <p>Ce cours a pour objectif d'amener les étudiants à élargir leur culture sur différents sujets et paradigmes actuels en ingénierie du logiciel. Des thèmes d'étude sont proposés, chacun illustré par un ou deux documents (articles choisis). Les étudiants se répartissent les thèmes d'étude et en proposent des mini-cours à destination des autres étudiants. Chaque mini-cours donne lieu à une discussion (de type atelier). Les étudiants sont évalués tant sur leur production que sur leurs réactions aux cours de leurs collègues.</p>
<p>Programme et contenu :</p> <p>Cours inversé, sujets et paradigmes actuels : fouille de dépôts de code et génie logiciel empirique, Apports de l'intelligence artificielle au génie logiciel, architectures logicielles vertes, conteneurisation d'applications, ingénierie des systèmes d'intelligence artificielle, architectures à micro-services, ingénierie d'applications pour l'internet des objets ou les systèmes cyber-physiques, déploiement de logiciels, ...</p>
<p>Méthode et organisation pédagogique :</p> <p>Les enseignements sont prévus pour 30 étudiants. Les mini-cours seront préparés par les étudiants en travail personnel. Ils seront dispensés en utilisant les ordinateurs personnels des étudiants.</p> <p>Le découpage est prévu comme suit :</p> <ul style="list-style-type: none"> - 2h de cours : présentation de l'exercice, des thèmes, des attendus. Les étudiants se répartissent ensuite les thèmes et constituent des groupes en autonomie - 13h de mini-cours inversés : les étudiants présentent selon un planning qui leur est communiqué à l'avance.
<p>Acquis d'apprentissage visés :</p> <p>Connaître différents sujets et paradigmes actuels d'ingénierie du logiciel Maîtriser la préparation puis la présentation d'un mini-cours sur un sujet technique Montrer sa curiosité scientifique et savoir participer et animer une discussion</p>
<p>Évaluation : évaluation des mini-cours et de la participation aux discussions et à l'animation. Une évaluation par les pairs sera intégrée.</p>
<p>Retour sur l'évaluation fait à l'élève : temps de bilan oral à l'issue des présentations, évaluation de l'enseignement et suggestions faites par les étudiants</p>
<p>Support pédagogique et références : documents (articles choisis) fournis en début de cours</p>

Méthode et organisation pédagogique

A l'exception du cours en pédagogie inversée, il s'agit d'un enseignement relativement classique avec une partie réalisée en cours magistral et une partie appliquée au travers de TP et Projets.

Modalité d'évaluation

Le niveau d'acquisition des compétences sera évalué selon les exigences suivantes :

N° indicateur	Indicateur
1	connaître les savoirs formels et pratiques du socle des fondamentaux
2	Exploiter les savoirs théoriques et pratiques
3	Analyser, interpréter, modéliser, émettre des hypothèses, et résoudre

Répartition

Matière	Contrôle	Coefficients	Type de notation	Indicateurs évalués	Chapitres
Méta-modélisation et transformation de modèles	Projet	1	En binôme	1, 2 et 3	Tous
Bonnes pratiques et développement centré architectures	Projet	1	En binôme	2,3	Tous
Sujets et paradigmes actuels en ingénierie du logiciel	Évaluation des mini-cours et participation	1	Travail préparé en binôme / évaluation individuelle	3	Tous

Dans chacune des matières du module, une évaluation non prévue à l'emploi du temps (contrôles surprise) peut advenir.

Engagement de l'étudiant, éthique et professionnalisme

La démarche éthique est définie dans le règlement intérieur de l'établissement. Chaque étudiant s'engage à en prendre connaissance et à la respecter.

Obligation des cours : Présence obligatoire pour tous à chaque séance.

Engagement de conduite éthique : Aucune triche ou plagiat ne sera toléré, toute tentative avérée mènera à la note de zéro pour tout le groupe d'étudiants sans distinction.

Nombre d'heures estimées de travail personnel : pour acquérir les compétences demandées, il est nécessaire que l'étudiant consacre minimum 45 min de travail personnel de compréhension et d'approfondissement par séance de cours.

30h de travail personnel pour définir le périmètre des projets

Nombre d'heures estimées de préparation aux travaux dirigés (TD) : 1 à 2h

Pénalité pour retard

Tout travail remis en retard sans motif valable peut être pénalisé de 1 point par jour de retard.

Équipe enseignante

Nom	Domaine d'expertise	Courriel/Téléphone
Christelle URTADO	Génie Logiciel, Composants et architectures logicielles	christelle.urtado@mines-ales.fr 04 34 24 62 89
Clémentine NEBUT	Génie Logiciel, Ingénierie Dirigée par les Modèles	nebut@lirmm.fr
Sylvain VAUTTIER	Génie Logiciel, Composants et architectures logicielles	sylvain.vauttier@mines-ales.fr 04 34 24 62 85

ACADEMIC TEACHING	Teaching hours	Coefficients	Credits
Model Driven Engineering	50 h		
○ Meta-modeling and Model transformation	15	1	4
○ Best development practices and architecture centered development	20	1	
○ Modern software engineering themes and paradigms	15	1	

Class 1

Class title: Meta-modeling and Model transformation	
Code: 2IA-il 9.4.1	Module title: « Model Driven Engineering »
Semester: S9	Classification: CSAI department, SE option

Hours of presence	Total hours	Lectures	Workshop	Labs	Project	Testing	Personal work	Coef /module	ECTS
15	25	6			9		10	1	1.3

Summary	This module presents the concepts of meta-modeling and meta-programming (design of modeling and programming languages), model transformation techniques and application of these notions to define new languages and extend software development tools.
----------------	---

Head	Christelle Urtado – LGI2P/IMT Mines Alès
Teaching team	Clémentine Nebut – Univ. Montpellier

Key words	Métamodèle, UML, MOF, transformation de modèles, DSL, QVT, Ecore
Prerequisites	2IA 8.4 Module : Software Engineering

<p>Context and general objective:</p> <p>The objective of this course is to present meta-modeling and model transformation as a modern alternative to more classical software engineering techniques. It also aims to show how languages and tools can be extended by using meta-programming; define and tool a domain-specific language (DSL) and implement model transformations.</p>
<p>Program and contents:</p> <p>Metamodeling Example of the UML meta-model Model transformation Domain-specific language definition Programming et extending development tools using standard meta-models such as MOF or Ecore</p>
<p>Method and pedagogic organisation:</p> <p>The course is tailored to groups of 30 students. For labs and projects, students will use their own laptops. The course decomposes into :</p> <ul style="list-style-type: none"> - 6h lecture - 9h project
<p>Targeted skills or knowledge:</p> <p>Understand the notion of meta-model Know the example of the UML meta-model Understand the issues of creating new specific languages Know how to define the meta-model of a simple language Understand the usefulness of model transformations and know how to implement some</p>
<p>Evaluation : Notation based on a project (coef 1)</p>
<p>Feedback made to the student: project evaluation form after a maximum of three weeks after the project have taken place.</p>
<p>Teaching material and references:</p> <p>Lecture slides</p>

Class 2

Class title: Best development practices and architecture centered development	
Code: 2IA-il 9.4.2	Module title: « Model Driven Engineering »
Semester: S9	Classification: CSAI department, SE option

Hours of presence	Total hours	Lectures	Workshop	Labs	Project	Testing	Personal work	Coef /module	ECTS
20	30	8			12		10	1	1.4

Summary	This course presents best development practices that make it possible to continuously release the latest version of the developed software while raising its overall quality. It also illustrates how software development can be driven by component-based architecture modeling.
----------------	--

Head	Christelle Urtado – LGI2P/IMT Mines Alès
Teaching team	Christelle Urtado – LGI2P/IMT Mines Alès Sylvain Vauttier – LGI2P/IMT Mines Alès PhD student to help tutor the project

Key words	Software quality, tests, continuous integration, software component, architecture, architecture description language
Prerequisites	2IA 8.4 Module : Software Engineering

<p>Context and general objective:</p> <p>This course first presents good software development practices, including software metrics and qualities, test writing and the continuous integration chain. Agile methods are also mentioned.</p> <p>In a second step, it illustrates architecture-centred development by introducing expected qualities such as modularity and decoupling to enable engineering by reuse. Software components and software architectures are presented as well as the impact of these concepts on the development cycle (containers, directories).</p>
<p>Program and contents:</p> <p>Part 1 : Best development practices Software quality and test Agile development methods Continuous integration</p> <p>Part 2 : Architecture-centered development Modularity, decoupling and reuse Definition et modeling of a software component Definition et modeling of a component-based software architecture Runtime frameworks: containers, service directories, component repositories</p>
<p>Method and pedagogic organisation:</p> <p>The course is tailored to groups of 30 students. For labs and projects, students will use their own laptops. The course is decomposed into:</p> <ul style="list-style-type: none"> - 8h lecture - 12h project
<p>Targeted skills or knowledge:</p> <p>Understand issues of best software development practices Know how to use quality measure tools, write tests and set a continuous integration Know principles of agile development methods Understand issues of architecture-centered development Know how to model a component-based software architecture Know how to set an architecture-centered development process</p>
<p>Evaluation : Notation based on a project (coef 1)</p>
<p>Feedback made to the student: project evaluation form after a maximum of three weeks after the project have taken place.</p>
<p>Teaching material and references:</p> <p>Lecture slides</p>

Class 3

Class title: Modern software engineering themes and paradigms	
Code: 2IA-il 9.4.3	Module title: « Model Driven Engineering »
Semester: S9	Classification: CSAI department, SE option

Hours of presence	Total hours	Lectures	Workshop	Labs	Project	Testing	Personal work	Coef /module	ECTS
15	25	15			0		10	1	1.3

Summary	This course aims to broaden student knowledge on various modern software engineering issues and paradigms. Various themes are proposed, each illustrated by one or two chosen articles. Students choose among the proposed themes so that all themes are covered and develop mini-lectures that are given in front of the other students. Each mini-lecture is followed by a Q&A session (workshop like). Evaluation is based on both the lecture and oral participation.
----------------	---

Head	Christelle Urtado – LGI2P/IMT Mines Alès
Teaching team	Christelle Urtado – LGI2P/IMT Mines Alès

Key words	Active pedagogy, modern subjects and paradigms: agility, extreme programming, aspect-oriented programming, empirical software engineering, software evolution, decoupling and inversion of control, ...
Prerequisites	2IA 8.4 Module : Software Engineering

<p>Context and general objective:</p> <p>This course aims to broaden student knowledge on various modern software engineering issues and paradigms. Various themes are proposed, each illustrated by one or two chosen articles. Students choose among the proposed themes so that all themes are covered and develop mini-lectures that are given in front of the other students. Each mini-lecture is followed by a Q&A session (workshop like). Evaluation is based on both the lecture and oral participation.</p>
<p>Program and contents:</p> <p>Active pedagogy, modern subjects and paradigms: agility, extreme programming, aspect-oriented programming, empirical software engineering, software evolution, decoupling and inversion of control, ...</p>
<p>Method and pedagogic organisation:</p> <p>The course is tailored to groups of 30 students. Students will prepare mini-lectures on their personal work time and use their own laptops to give their mini-lectures.</p> <p>The course is organized as follows:</p> <ul style="list-style-type: none"> - 2h lecture : presentation of the themes, constraints and objectives. Students then choose their study-theme and constitute groups. - 13h mini-lectures by students: students give their mini-lectures following a schedule provided in advance.
<p>Targeted skills or knowledge:</p> <p>Be aware of various modern subjects and paradigms of software engineering Master preparation and lecture of a « mini-lecture » on a technical subject Show scientific openness and ability to participate in a Q&A session</p>
<p>Evaluation : evaluation of the mini-lectures and of oral participation / animation. Peer evaluation will also be integrated. (coef 1)</p>
<p>Feedback made to the student: oral feedback at the end of mini-lecture sessions, evaluation of the course and improvement suggestions by students</p>
<p>Teaching material and references:</p> <p>Chosen articles provided at the beginning of the course.</p>

Method and teaching organisation

This will be a classical course with lectures and applications on TD and Project. One class will be driven under an active pedagogy organization.

Testing procedures

The student's level of knowledge acquisition will be evaluated according to the following points:

N° Indicator	Indicator
1	To know the formal and practical knowledge constituting the foundation of a given field
2	Exploit theoretical and practical knowledge
3	Analyse, interpret, model, hypothesize and solve problems

Grading scheme

Class	Exam	Coefficients	Administration mode	Evaluated Indicators	Chapters
Meta-modeling and model transformation	Project	1	In pairs	1,2,3	All
Best development practices and architecture-centered development	Project	1	In pairs	2,3	All
Sujets et paradigmes actuels en ingénierie du logiciel	Evaluation of mini-lectures and oral participation	1	Work done in pairs / Individual evaluation	3	All

In each course of this module, an unscheduled assessment may occur.

Student commitments, ethics and professionalism

Expectations concerning ethics are defined in the establishment's code of conduct. Each student is expected to know and respect the code of conduct.

Obligatory presence in classes: Presence is required all times in lectures and practical sessions.

Commitment to ethical conduct: No cheating or plagiarism will be tolerated; any proven attempt will lead to a score of zero for the entire group of students without distinction.

Estimated hours of personal study

In order to acquire the required learning level, the student is expected (must) spend a minimum of 45min of personal study time per hour spent in class.

30h of additional personal work is required on the projects.

Estimated hours of preparation required for labs/Work Shop: 1 to 2 hours

Late penalties : All late assignments without any valid cause may be penalized by the withdrawal of up to one point per day after the due date.

Teaching team

Name	Field of expertise	Email/Phone
Christelle URTADO	Software Engineering, Software components and architectures	christelle.urtado@mines-ales.fr 04 34 24 62 89
Clémentine NEBUT	Software Engineering, Model driven engineering	nebut@lirmm.fr
Sylvain VAUTTIER	Software Engineering, Software components and architectures	sylvain.vauttier@mines-ales.fr 04 34 24 62 85

Approbation

Ce guide pédagogique entré en vigueur à compter du 7 janvier 2019 a été mis à jour en novembre 2020. Il est porté à la connaissance des étudiants par une publication sur site Internet de l'école

Rédaction	Vérification	Validation
L'enseignant responsable du module : Christelle URTADO	Le responsable d'UE / de département : Sylvie RANWEZ	Le directeur de l'école, Pour le directeur et par délégation, Le directeur de la DFA / de la DE : Michel FERLUT