

Pourquoi cette UE ?

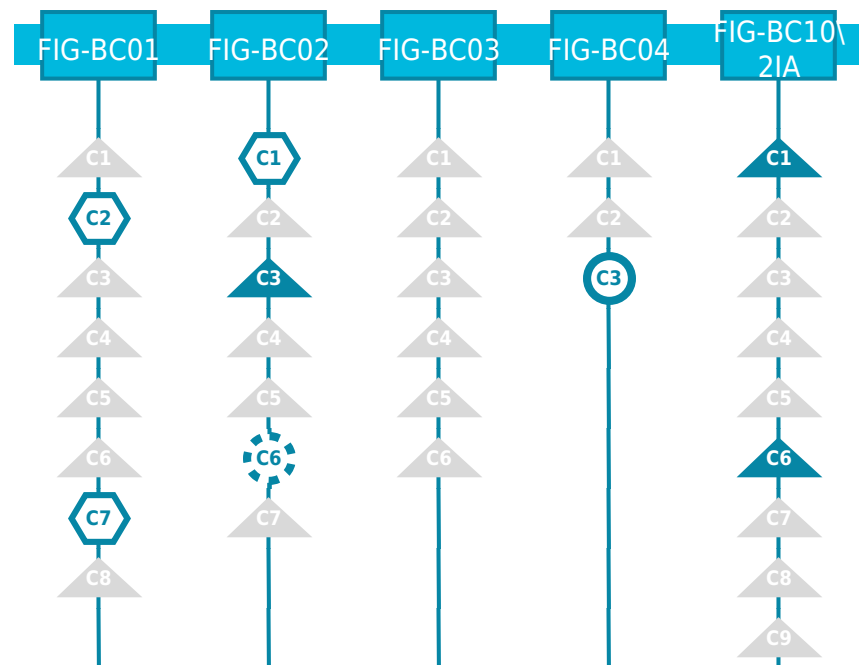
L'ingénierie dirigée par les modèles (IDM) regroupe un ensemble de techniques avancées de développement logiciel qui remplace, autant que possible, l'écriture de code par l'écriture de modèles. Ce module pose les fondements de l'ingénierie dirigée par les modèles. Y sont présentés la métamodélisation 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.

Éléments constitutifs de l'UE

		coefficient
2IAiail_9_4-1 Méta-modélisation et transformation de modèles		1
2IAiail_9_4-2 Bonnes pratiques et développement centré-architecture		1
2IAiail_9_4-3 Sujets et paradigmes actuels en ingénierie du logiciel		1
Volume d'heures d'enseignement encadré	Volume d'heures de travail personnel	Nombre d'ECTS
50	30	4

Alignement curriculaire

Parmi les compétences visées par la formation, lesquelles sont développées dans cette UE ?



- BC1 L'UE ne contribue pas à ce bloc de compétences
- BC1 L'UE contribue à ce bloc de compétences
- C1 Compétence non adressée dans cette UE
- C1 Compétence mise en œuvre dans cette UE
- C1 Compétence enseignée dans cette UE
- C1 Compétence évaluée dans cette UE
- C1 Compétence enseignée et évaluée dans cette UE

2IAiail_9_4 Ingénierie dirigée par les modèles et qualité logicielle	FIG
2IAiail_9_4-1 Méta-modélisation et transformation de modèles	S9

Contexte et enjeux de l'enseignement

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.

Prise en compte des dimensions socio-environnementales

Prérequis

Module 2IA 8.4 Ingénierie Logicielle

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	6
Cours intégré (cours + TD)	
TD	
TP	
Projets	9
Travail en autonomie encadré	
Contrôles et soutenances	
Travail personnel	10

Objectifs pédagogiques	Activités	Évaluations et retours faits aux élèves
(à la fin de cet enseignement, l'étudiant sera capable de ...)	(CM, TD, TP, projet, sortie terrain, etc.)	(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)
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	Les enseignements sont prévus pour 30 étudiants. Les manipulations seront réalisées sur les ordinateurs personnels de ces derniers. Le découpage est prévu comme suit : - 6h de cours - 9h de projet	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 dernière séance.

2IAiail_9_4 Ingénierie dirigée par les modèles et qualité logicielle	FIG
2IAiail_9_4-1 Méta-modélisation et transformation de modèles	S9

Plan de cours

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
--

Ressources et références

1 Polycopié composé des supports de présentation
--

2IAiail_9_4 Ingénierie dirigée par les modèles et qualité logicielle	FIG
2IAiail_9_4-2 Bonnes pratiques et développement centré-architecture	S9

Contexte et enjeux de l'enseignement	Prise en compte des dimensions socio-environnementales	Modalités d'enseignement et d'évaluation	
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. 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 dev			Nb d'heures
		Cours	8
		Cours intégré (cours + TD)	
		TD	
		TP	
		Projets	12
		Travail en autonomie encadré	
		Contrôles et soutenances	
		Travail personnel	10

Objectifs pédagogiques	Activités	Évaluations et retours faits aux élèves	
la fin de cet enseignement, l'étudiant sera capable de ...)	(CM, TD, TP, projet, sortie terrain, etc.)	(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)	
Comprendre les enjeux des bonnes pratiques de développement logiciel Savoir utiliser les outils de mesure de la qualité, écrire des tests, modulariser une application et mettre en place une chaîne d'intégration continue Être sensibilisé aux méthodes de développement agiles Comprendre les enjeux du développement centré architecture Savoir modéliser une architecture de logiciel à base de composants en respectant des patrons d'architecture	Les enseignements sont prévus pour 30 étudiants. Les TP seront réalisés sur les ordinateurs personnels de ces derniers. Le découpage est prévu comme suit : - 8h de cours - 12h de projet	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	

2IAiail_9_4 Ingénierie dirigée par les modèles et qualité logicielle	FIG
2IAiail_9_4-2 Bonnes pratiques et développement centré-architecture	S9

Plan de cours

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) 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
--

Ressources et références

1 Polycopié composé des supports de présentation
--

Contexte et enjeux de l'enseignement

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.

Prise en compte des dimensions socio-environnementales**Prérequis**

1 Polycopié composé des supports de présentation

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	15
Cours intégré (cours + TD)	
TD	
TP	
Projets	
Travail en autonomie encadré	
Contrôles et soutenances	
Travail personnel	10

Objectifs pédagogiques

(à la fin de cet enseignement, l'étudiant sera capable de ...)

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

Activités

(CM, TD, TP, projet, sortie terrain, etc.)

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.
Le découpage est prévu comme suit :
- 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.

Évaluations et retours faits aux élèves

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

Evaluation des mini-cours et de la participation aux discussions et à l'animation. Une évaluation par les pairs sera intégrée.

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

Plan de cours

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, ...

Ressources et références

Documents (articles choisis) fournis en début de cours