

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE**

# **Cahier des Charges d'Habilitation d'une Formation à Recrutement National**

**Master**

**- Génie Logiciel -**

**Université Abdelhamid Mehri – Constantine 2**

الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي والبحث العلمي

دفتـر الشـروط

لتأهيل تكوين ذات تسجيل وطني

ماسـتر

- هندسة البرمجيات -

جامعة عبد الحميد مهري – قسنطينة 2

## **A – Fiche d'identification du Master**

**Etablissement :** Université Abdelhamid Mehri – Constantine 2

**Faculté ou Institut :** Faculté des Nouvelles Technologies de l'Information et de la Communication

**Département :** Technologie du Logiciel et Systèmes d'Information

**Domaine:** MI (Mathématiques et Informatique)

**Filières/spécialités :** Génie Logiciel

**Responsable du Master :**

**Nom :** Bouanaka

**Prénom :** Chafia

**Grade :** Maitre de Conférences Classe B

**Email :** [chafia.bouanaka@univ-constantine2.dz](mailto:chafia.bouanaka@univ-constantine2.dz)

# **Organisation de la formation**

# Semestre 1

Unités d'enseignement	Matières	Crédits	Coefficient	Volume horaire hebdomadaire			VHS (14-16 semaines)	Autre*	Mode d'évaluation	
	Intitulé			Cours	TD	TP			Contrôle Continu	Examen
<b>UE Fondamentale</b> <b>Code : UEF 1.1.1</b> <b>Crédits : 13</b> <b>Coefficients : 8</b>	Programmation orientée composants	7	4	3h		3h	84h		✓	✓
	Programmation Logique et Fonctionnelle	6	4	3h	1h.30	1h30	84h		✓	✓
<b>UE Fondamentale</b> <b>Code : UEF 1.1.2</b> <b>Crédits : 12</b> <b>Coefficients : 7</b>	Sécurité des Réseaux	5	3	1h30	1h30	1h30	6 3h		✓	✓
	Systèmes Distribués	7	4	3h	1h30		63H		✓	✓
<b>UE Methodologique</b> <b>Code : UEM1.1.1</b> <b>Crédits : 5</b> <b>Coefficients : 3</b>	Statistiques et Analyse de données	3	2	1h30	1h30		42h		✓	✓
	Anglais	2	1	1h.30			21h			✓
<b>Total semestre 1</b>		<b>30</b>	<b>18</b>	<b>13h30</b>	<b>6h</b>	<b>6h</b>	<b>357h</b>			

## Semestre 2

Unités d'enseignement	Matières	Crédits	Coefficient	Volume horaire hebdomadaire			VHS (14-16 semaines)	Autre*	Mode d'évaluation	
	Intitulé			Cours	TD	TP			Contrôle Continu	Examen
UE Fondamentale Code : UEF 1.2.1 Crédits : 13 Coefficients : 7	Spécification et vérification des systèmes	7	4	3h	1h30	1h30	84h		✓	✓
	Modélisation et Simulation des Systèmes Informatiques	6	3	1h30	1h30	1h30	63h		✓	✓
UE Fondamentale Code : UEF 1.2.2 Crédits : 11 Coefficients : 6	Exigences et Développement de Logiciels	7	4	3h		3h	84H		✓	✓
	Analyse de la qualité du logiciel	4	2	1h30h	1h30		42H		✓	✓
UE Méthodologique Code : UEM 1.2.1 Crédits : 4 Coefficients : 2	Techniques de Maintenance du Logiciel	4	2	1h30		1h30	42h		✓	
UE Découverte Code : UED 1.2.1 Crédits : 2 Coefficients : 1	Pratique Professionnelle du GL	2	1	1h30			21h		✓	
Total semestre 2		30	16	12h	4h30	7h30h	336h			

## Semestre 3

Unités d'enseignement	Matières	Crédits	Coefficient	Volume horaire hebdomadaire			VHS (14-16 semaines)	Autre*	Mode d'évaluation	
	Intitulé			Cours	TD	TP			Contrôle Continu	Examen
<b>UE Fondamentale</b> <b>Code : UEF 2.1.1</b> <b>Crédits : 10</b> <b>Coefficients : 7</b>	Ingénierie Dirigée par les Modèles	4	4	1h30		1h30	42h		✓	✓
	Architectures Logicielles	6	3	1h30		1h30	42h		✓	✓
<b>UE Fondamentale</b> <b>Code : UEF 2.1.2</b> <b>Crédits : 10</b> <b>Coefficients : 7</b>	SOA et Cloud Computing	6	4	1h30		1h30	42h		✓	✓
	Administration des BDs	4	3	1h30		1h30	42h		✓	✓
<b>UE Méthodologique</b> <b>Code : UEM 2.1.1</b> <b>Crédits : 10</b> <b>Coefficients : 7</b>	Programmation Orientée Agents	6	4	1h30	1h30	1h30	63h		✓	✓
	Méthodes Agiles	4	3	1h30		1h30	42h		✓	✓
<b>Total semestre 3</b>		<b>30</b>	<b>21</b>	<b>9h</b>	<b>1h30</b>	<b>9h</b>	<b>273h</b>			



## Semestre 4

Projet de fin d'étude sanctionné par un mémoire et une soutenance.

	VHS	Coeff	Crédits
<b>Travail Personnel</b>	14 semaines	9	27
<b>Stage en entreprise</b>			
<b>Séminaires à thèmes ouverts</b>	1h30/semaine	1	3
<b>Autre (préciser)</b>			
<b>Total Semestre 4</b>	14 semaines	10	30

### Récapitulatif global de la formation:

<b>VH \ UE</b>	<b>UEF</b>	<b>UEM</b>	<b>UED</b>	<b>UET</b>	<b>Total</b>
<b>Cours</b>	25h30	7h30	3h		36h
<b>TD</b>	9h	3h	0h		12h
<b>TP</b>	18h	4h30	0h		22h30
<b>Travail personnel</b>	24h	14h	4h		42h
<b>Autre (préciser)</b>					
<b>Total</b>	52h30	15h	3h		70h30
<b>Crédits</b>	95	18	7		<b>120</b>
<b>% en crédits pour chaque UE</b>	79,16%	15%	5,83%	25	100%

# **Programme Détaillé par Matière**

**Semestre : 1**

**UE : UEF 1.1.1**

**Intitulé de matière: Programmation Orientée Composants**

### **Objectifs de l'enseignement**

Ce cours comprend deux parties. La première vise à introduire les applications à base d'objets distribués, leurs structures et leurs constituants de base, en insistant sur la notion d'interface de l'objet distribué, proxy et référence d'objets. Cette partie met en évidence le rôle des intergiciels dans les applications distribuées. La deuxième concerne le développement d'applications à base de composants distribués. Elle présente quelques technologies de composants.

### **Connaissances préalables**

- Concepts de base du GL
- Programmation orientée objet

### **Contenu de la matière :**

#### **Chapitre 1: Introduction aux applications distribuées**

- 1.1 Introduction
- 1.2 Construction d'applications réparties en mode message
- 1.3 Construction d'applications réparties dans un environnement OO
- 1.4 Environnements d'exécution à objets répartis

#### **Chapitre 2 : Mécanisme RPC**

- 2.1 Motivation
- 2.2 Principe de base
- 2.3 Eléments d'un RPC

#### **Chapitre 3 : Introduction à la POC**

- 3.1 Définition de la POC
- 3.2 Rôles et intérêts de la POC
- 3.3 Définition d'un composant
- 3.4 Différences entre la POO et la POC

#### **Chapitre 4: Les composants .NET**

- 4.1 Le cadre .NET
- 4.2 Le modèle composant de .NET
- 4.3 Le modèle de connexion de .NET
- 4.4 Le modèle de déploiement de .NET
- 4.5 Visual Studio .NET
- 4.6 Exemples pratiques

#### **Chapitre 5 : Les composants OSGI**

- 5.1 L'infrastructure OSGI
- 5.2 Le modèle composant des OSGI
- 5.3 Le modèle de connexion OSGI
- 5.4 Le modèle de déploiement OSGI

## 5.5 Exemples pratiques

### **Chapitre 6 : Les composants Fractal**

- 6.1 Introduction à l'infrastructure Fractal
- 6.2 Le modèle composant de Fractal
- 6.3 Le modèle connexion de Fractal
- 6.4 Le modèle de déploiement de Fractal
- 6.5 Exemples pratiques

### **Chapitre 7 : Modèle de composants orienté service (SCA)**

- 7.1 Le modèle de composant SCA
- 7.2 Le modèle de connexion SCA
- 7.3 Outils d'implémentation du modèle SCA
- 7.4 Exemples pratiques

### **Mode d'évaluation : TP, Examen écrit**

### **Références** (*Livres et photocopiés, sites internet, etc*).

1. Intergiciel et Construction d'Applications Réparties, (19 janvier 2007)  
Distribué sous licence Creative Commons :  
<http://creativecommons.org/licenses/by-nc-nd/2.0/fr/deed.fr>
2. COMPONENT-ORIENTED PROGRAMMING, ANDY JU AN WANG KAI QIAN  
Southern Polytechnic State University, Marietta, Georgia  
A JOHN WILEY & SONS, INC., PUBLICATION, 2008.
3. Tuscany SCA in action, Simon Laws, Mark Combellack, Raymond Feng, Halen Mahbod, Simon Nash. Manning Publications, 2011.

**Semestre : 01**

**UE : UEF 1.1.2**

**Intitulé de matière: Programmation Logique et Fonctionnelle**

### **Objectifs de l'enseignement**

En s'appuyant sur la programmation déclarative, l'objectif de ce cours est d'exposer les techniques de base de l'Intelligence Artificielle. Situer l'Intelligence Artificielle en tant que discipline scientifique, Maîtriser les concepts de base de l'intelligence artificielle.

### **Connaissances préalables recommandées**

Les connaissances algorithmiques et éléments de logiques acquises en cours de formation « licence »

### **Contenu de la matière :**

Ce cours comprend deux parties.

**Partie 1 :** cette première partie vise à étudier les paradigmes de programmation fonctionnelle et logique. Pour chacun, on y présente d'abord les concepts de base (fonction, réduction, évaluation, pour la programmation fonctionnelle, logique de Horn, stratégie de sélection, négation) avant d'aborder des sujets plus avancés, comme les fonctions d'ordre supérieur, les types polymorphes, les monades ou encore quelques exemples de programmation non déterministe. Une attention particulière est aussi portée aux techniques de programmation.

**Partie 2 :** la deuxième partie aborde les techniques de base de l'Intelligence Artificielle. Après une introduction à ce domaine ainsi que l'exposé de concepts fondamentaux au travers d'un exemple simple de système expert, cette deuxième partie s'articule autour de trois parties:

- i. étude des fonctionnalités et des composants d'un système expert,
- ii. techniques de représentation des connaissances,
- iii. techniques de résolution de problèmes.

L'accent est, en particulier, mis sur l'utilisation de la logique et de la programmation par contraintes. On y développe enfin les bases de l'apprentissage automatique.

### **Références**

H. Farreny, M. Ghallab. *Eléments d'Intelligence Artificielle*. Hermès, 1985.

J.-P. Haton, N. Bouzid, F. Charpillet, M.-C.Haton, B. et H. Lâasri, P. Marquis, T. Mondot, A. Napoli. *Le Raisonnement en Intelligence Artificielle - Modèles, Techniques et Architectures pour les Systèmes à Bases de Connaissances*. InterEditions, 1991.

J.-L. Laurière. *Intelligence Artificielle : Résolution de Problèmes par l'Homme et la Machine*. Eyrolles, Paris, 1986.

Robert W. Sebesta, " Concepts of Programming Languages", Addison Wesley 7th Edition, 2006.

## **Semestre : 1**

### **Unité d'enseignement : UEF 1.1.2**

### **Matière : Sécurité des Réseaux**

#### **Objectifs de l'enseignement :**

L'étudiant est censé acquérir une maîtrise globale de la sécurisation d'un réseau privé d'entreprise utilisant les technologies Internet/intranet, et de son interconnexion avec des réseaux extérieurs, ainsi cette matière doit offrir à l'étudiant:

- Initiation aux principes de sécurité des réseaux informatiques en termes de méthodes, pratiques et politiques.
- Présentation des dispositifs logiciels et matériels qui servent à assurer la sécurité des ressources en informatique.
- Évaluation des risques d'attaque, et estimation du niveau de vulnérabilité et proposition de solutions de protection du réseau.

#### **Connaissances préalables recommandées :**

- Principes fondamentaux des réseaux de communication TCP/IP,
- Principes fondamentaux de l'administration des réseaux en termes d'architectures de services.

#### **Contenu de la matière :**

##### **1. Introduction à la sécurité des réseaux**

- 1.1. Définition des principaux concepts de sécurité informatique.
- 1.2. Etat des lieux de la sécurité informatique.
- 1.3. Types de menaces et d'attaques Réseau
  - 1.3.1. Attaques "couches basses".
  - 1.3.2. Attaques applicatives.
  - 1.3.3. Déni de service
  - 1.3.4. Vers, virus, trojans, malwares et keyloggers.

##### **2. Sécurisation d'un système ("Hardening")**

- 2.1. Insuffisance des installations par défaut.
- 2.2. Critères d'évaluation
- 2.3. Sécurisation d'Unix-Linux.
- 2.4. Sécurisation de Windows.

##### **3. Sécurisation des Données ou Cryptographie.**

- 3.1. Cryptographie
  - 3.1.1. Introduction à la cryptographie (protocoles et techniques).
  - 3.1.2. Chiffrements symétrique et asymétrique.
  - 3.1.3. Fonctions de hachage.
  - 3.1.4. Services cryptographiques.
- 3.2. Authentification de l'utilisateur.
- 3.3. Protocole SSL/TLS,
- 3.4. Protocole SSH.
- 3.5. Signature électronique(protocole LDAP)

#### **4. Politique et Architecture de sécurité (normes et standards).**

- 4.1. Quelles architectures pour quels besoins ?
  - 4.1.1. Plan d'adressage sécurisé : RFC 1918.
  - 4.1.2. Translation d'adresses
  - 4.1.3. Rôle des zones démilitarisées (DMZ).
- 4.2. Virtualisation.
- 4.3. Firewall
  - 4.3.1. Actions et limites
  - 4.3.2. Evolution technologique des firewalls.
  - 4.3.3. Proxy serveur et relais applicatif,
  - 4.3.4. Relais SMTP.
- 4.4. IPSec
  - 4.4.1. Présentation du protocole.
  - 4.4.2. Modes tunnel et transport.
  - 4.4.3. ESP et AH.

#### **5. Moyens d'Analyse et de Détection des Attaques**

- 5.1. Tests d'intrusion : outils et moyens.
- 5.2. Détection des vulnérabilités (scanners, sondes IDS, etc.).
- 5.3. Outils de détection temps réel IDS-IPS, agent, sonde ou coupure.
- 5.4. Supervision et administration.
- 5.5. Veille technologique.

#### **6. Mécanismes de la sécurité des réseaux *sans fil* (Facultatif)**

- 6.1. Sécurité Wi-Fi.
- 6.2. Risques inhérents aux réseaux sans fil.
- 6.3. Types d'attaques.
- 6.4. Limites du WEP.
- 6.5. Protocole WPA et WPA2.
- 6.6. Attaque Man in the Middle avec le rogue AP.

#### **Travaux pratiques (à titre indicatif)**

- Installation et utilisation de l'analyseur réseau Wireshark ou autre à travers GNS3.
- Exemple de sécurisation d'un système Linux.
- Mise en œuvre d'une attaque type ARP.
- Mise en œuvre d'une attaque applicative.
- Mise en œuvre d'un proxy Cache/Authentification.
- Déploiement d'un relais SMTP et d'un proxy HTTP/FTP Antivirus.
- Mise en œuvre d'IPSec mode transport/PSK.

**Mode d'évaluation :** Contrôle de connaissances + contrôle TP + évaluation continue

## Références bibliographiques :

- Solange Ghernaouti-Hélie, « Sécurité Informatique et Réseaux », 2006, Dunod ISBN-2100074482.
- Raymond Panko « Sécurité des Systèmes d'information et des Réseaux » , Pearson Education Stallings, W. Network Security Essentials, 2<sup>ième</sup> edition. Prentice Hall, 2003
- Eric Maiwald, « Network Security A Beginner's Guide », 3<sup>ième</sup> Edition, Publisher: McGraw-Hill Osborne Media. 2012.



**Semestre : 01**

**UE : UEF 1.1.2**

**Intitulé de matière: Systèmes Distribués**

### **Objectifs de l'enseignement**

Acquérir les concepts de base concernant la structure et le fonctionnement d'un système distribué, tous les mécanismes de partage d'information et de communication doivent être étudiés.

### **Connaissances préalables recommandées**

- Systèmes d'exploitation
- Systèmes centralisés
- Programmation Java

### **Contenu de la matière :**

#### **Chapitre I : Introduction aux systèmes distribués**

1. Définition structurelle d'un système distribué
2. Motivation des systèmes distribués
  - 2.1. Avantages des systèmes distribués/systèmes centralisés
  - 2.2. Avantages des systèmes distribués/ordinateurs personnels
  - 2.3. Désavantage des systèmes distribués

#### **Chapitre II : Systèmes Distribués : Définition et caractéristiques**

1. Définition d'un système Distribué
2. Caractéristiques d'un système distribué
  - 2.1. Mécanisme de communication
  - 2.2. Structure globale de nommage
  - 2.3. Image unique
  - 2.4. Partage des organes de stockages
  - 2.5. Temps de réponse
3. Différences des systèmes distribués/ systèmes centralisés
  - 3.1. Recouvrement
  - 3.2. Sécurité
  - 3.3. Temps de réponse
  - 3.4. Croissance incrémentale

#### **Chapitre III : Architecture des systèmes distribués**

1. Notion de client/serveur
2. Familles des systèmes distribués
  - 2.1. Monolithique noyau
    - 2.1.1. Première approche
    - 2.1.2. Deuxième approche
  - 2.2. Micro noyau

#### **Chapitre IV : Système de fichiers dans les systèmes distribués**

1. Systèmes de fichiers distribués
2. Classe des systèmes de fichiers distribués
3. Composantes d'un système de fichiers distribués

- 3.1. Service de répertoire
- 3.2. Communication
- 3.3. Service de fichiers
- 4. Service de répertoires
  - 4.1. Système hiérarchique
  - 4.2. Spécification des fichiers distants
  - 4.3. Binding (liaison)

## **Chapitre V : Mécanismes de communication dans les systèmes distribués**

- 1. Communication dans les systèmes distribués
- 2. Eléments de la communication
  - 2.1. Service Distant
  - 2.2. Caching
- 3. Service Distant
  - 3.1. Pourquoi pas OSI
  - 3.2. Simple protocole
  - 3.3. RPC
- 4. Caching
  - 4.1. Objectif
  - 4.2. Localisation du cache
  - 4.3. Politique de propagation de modifications
  - 4.4. Validation du cache
  - 4.5. Sémantique de partageabilité

## **Chapitre VI : Besoins d'un service de fichiers**

- 1. Service de fichiers
- 2. Caractéristiques désirables d'un service de fichiers
  - 2.1. Disponibilité
  - 2.2. Sécurité
  - 2.3. Croissance incrémentale
- 3. Structures de serveurs
- 4. Types de serveurs & tolérances aux pannes

**Mode d'évaluation :** (Contrôle \* 2 + interrogation)/3

**Références** (*Livres et photocopiés, sites internet, etc*).

- Distributed operating systems, Andrew S. Tanenbaum, Pearson Education, 1995.
- Distributed operating systems: concepts and design, Pradeep K. Sinha, PHI Learning, Pvt, Ltd, 1998.

**Semestre : 01**

**UE : UEM 1.1.1**

**Intitulé de matière:** Statistiques et Analyse de données

### **Objectifs de l'enseignement**

Il est tout à fait courant qu'après un certain traitement ou fonctionnement d'un système on se trouve en face d'une masse importante de données recueillies qu'on a besoin d'analyser. L'analyse des données est un domaine scientifique très large qui comporte des techniques variées principalement empruntées au domaine de la statistique et l'apprentissage automatique. Outre la préparation et le nettoyage des données, elle vise à mettre en exergue les relations qui peuvent exister entre les données, à résumer les données et / ou à établir des prédictions. Cette matière a pour objectif de permettre à l'étudiant de se familiariser avec le domaine de l'analyse de données et d'acquérir une certaine maîtrise des techniques et outils permettant d'analyser les données. L'étudiant aura l'occasion d'apprendre comment appréhender un tableau de données et à choisir et appliquer la technique appropriée pour répondre à un besoin d'analyse précis compte tenu de la nature des données disponibles.

### **Connaissances préalables recommandées**

Ce cours suppose que l'étudiant maîtrise le contenu du module probabilités et statistique vu en L1 licence.

### **Contenu de la matière**

#### **Chapitre 1 : Rappels des concepts de base de probabilités et statistiques**

- 1.1 Introduction : tableau de données, attributs, type d'attributs, les données d'un point de vue algébrique et géométrique.
- 1.2 Statistique descriptive
- 1.3 Probabilité et variables aléatoires
- 1.4 Distributions de probabilités usuelles
- 1.5 Tests d'hypothèses
- 1.6 Outils de visualisation

#### **Chapitre 2 : Modèles descriptifs pour l'analyse des données : clustering**

- 2.1 La classification non supervisée (clustering) des données
- 2.2 L'algorithme k-means
- 2.3 Le clustering hiérarchique
- 2.4 Le clustering basé densité : DBSCAN
- 2.5 Outils pour le clustering

#### **Chapitre 3 : Modèles descriptifs pour l'analyse des données : méthode de réduction de dimensionnalité**

- 3.1 Introduction
- 3.2 Analyse en composantes principales
- 3.3 Analyse factorielle des correspondances

**3.3** Analyse factorielle des correspondances multiples

**3.4** L'analyse discriminante

## **Chapitre 4 : Modèles prédictifs pour l'analyse des données : régression**

**4.1** Introduction

**4.2** Modèle de régression linéaire simple

**4.3** Modèle de régression non linéaire simple

**4.4** Modèle de régression multiple

**4.5** Outils pour la régression

## **Chapitre 5 : Modèles prédictifs pour l'analyse des données : classification supervisée**

**5.1** Introduction à la classification supervisée

**5.2** Classification Bayésienne

**5.3** Réseaux de neurones

**5.4** Classification par machines à vecteurs de support

**5.5** Outils pour la classification supervisée

**Mode d'évaluation :  $\text{Contrôle} \times 2 + (\text{Interrogation} + \text{TP})/2$**

## **Références bibliographiques :**

1. Ludovic Lebart, Marie Piron et Alain Morineau, Statistique Exploratoire multidimensionnelle, Paris, Dunod, (7-049616-10-2-978 NBSI) ,2006 .
2. Stéphane Tufféry, Data Mining et statistique décisionnelle, Paris, éditions Technip, (3-0946-7108-2-978 NBSI) ,2010
3. Härdle, W. Simar L. (2007) Applied Multivariate Statistical Analysis, 2nd edition, Springer.
4. Bach, F.R., Jordan, M.I., (2006). A probabilistic interpretation of Canonical Correlation Analysis. Univ. California, Berkeley.
5. G. Saporta, (2005). Probabilités, analyses des données et statistiques.

**Semestre : 2**

**Unité d'enseignement : UEF 1.2.1**

**Matière : Spécification et vérification des systèmes**

## **Objectifs de l'enseignement**

L'objectif de ce cours est de doter l'étudiant de techniques de spécification et de vérification formelles utilisables dans le contexte des systèmes répartis caractérisés par des composants distincts, chacun d'eux s'exécutant potentiellement sur un processeur différent.

## **Connaissances préalables recommandées**

Génie logiciel (Licence), Systèmes Distribués (S1)

## **Contenu de la matière :**

### **Chapitre1- Généralités sur les méthodes formelles**

1. Utilité des méthodes formelles en GL
2. Concepts de base (ensemble, prédicat, fonction, foncteur, équivalence, abstraction...)

### **Chapitre2- Spécification des systèmes**

1. Notion de spécification
2. Types de spécification (axiomatique, opérationnelle, ....)
3. Réseaux de Petri (simples, étendus, de haut niveau)
4. Algèbre de processus (CCS,  $\pi$ -calcul)
5. Systèmes de transitions étiquetés (LTS)

### **Chapitre 3- Propriétés à Vérifier**

1. Types de Propriétés  
Atteignabilité, Sureté, Vivacité, Équité, Absence de Blocage, Terminaison, ...
2. Spécification des propriétés
  - Logique des prédicats
  - Logique Temporelle

### **Chapitre4- Vérification des systèmes**

1. Notion de Vérification/validation
2. Types de Vérification
  - Formelle : Basées sur les preuves de théorèmes, Basées sur les équivalences, Basées sur le model-checking
  - Par Simulation
  - Par émulation

**Mode d'évaluation : (Contrôle \* 2 + Interrogation + TP)/4**

## Références bibliographiques:

- Michael Huth, Mark Ryan, “Logic in Computer Science Modeling and Reasoning about Systems” (2nd Edition), 2004.
- Joost-Pieter Katoen, “Principles of model-checking”, Lecture Notes, 2004.
- Henri Habrias and Marc Frappier” *Software Specification Methods*”, ISTE Ltd, 2006, ISBN 1-905209-34-7.
- Jean-François Monin “*Understanding Formal Methods*”, Springer, 2003, ISBN 1-85233-247-6.

**Semestre : 02**

**UE : UEF 1.2.1**

**Intitulé de matière:** Modélisation et Evaluation des Systèmes Informatiques

## **Objectifs de l'enseignement**

L'objectif de ce module est de sensibiliser les étudiants aux problèmes de modélisation et d'évaluation des performances des systèmes réels. A l'issue de ce module, l'étudiant sera capable de poser un modèle représentant le système étudié et de sélectionner, parmi les méthodes analytiques et de simulation, la plus appropriée.

## **Connaissances préalables recommandées**

Ce cours suppose que l'étudiant maîtrise le contenu du module Statistiques et Analyse des données de M1.

## **Contenu de la matière**

### **Chapitre 1 : Introduction à la modélisation et à l'évaluation de performances des systèmes**

- 1.1 Introduction
- 1.2 La modélisation des réseaux
- 1.3 Problème de la caractérisation et prévision de la charge
- 1.4 La qualité de service dans la conception des réseaux
- 1.5 Les solutions possibles pour dimensionner un réseau multiservice

### **Chapitre 2 : Le formalisme Files d'attentes**

- 2.1 La station de service de base
- 2.2 Processus des arrivées
- 2.3 Processus des services
- 2.4 Les chaînes de Markov
- 2.5 La file M/M/1 et la file M/M/C
- 2.6 Les réseaux de files d'attente

### **Chapitre 3 : La différenciation de services dans les réseaux**

- 3.1 Introduction
- 3.2 Le modèle DiffServ
- 3.3 Le traitement différencié de paquet dans le routeur DiffServ
- 3.4 Intégration MPLS/Diffserv

### **Chapitre 4 : Evaluation de performance de DiffServ**

- 4.1 Utilité des techniques de simulation
- 4.2 La simulation par événements
- 4.3 Estimation de la précision
- 4.4 Le simulateur NS
- 4.5 Implémentation de DiffServ dans NS
- 4.6 Résultats et interprétations : Equité de la bande passante et taux de perte

### **Chapitre 5 : Le simulateur NS (Travaux Pratiques)**

### **5.1** Présentation générale du simulateur NS2

### **5.2** Téléchargement et installation de NS2

### **5.3** Exemples de simulation sous NS2 :

- Exemple 1 : Simulation d'un réseau à deux nœuds
- Exemple 2 : Simulation de la fonction de routage et visualisation de la file d'attente
- Exemple 3 : Simulation d'un réseau ad hoc pour l'évaluation des performances d'un protocole de routage

**Mode d'évaluation : (Contrôle \* 2 + Interrogation + TP)/4**

### **Références bibliographiques :**

- Frikha, M., (2007). Planification et simulation des réseaux. Hermès Science Publication.
- Melin, J.-L., (2002). Qualité de services sur IP. Eyrolles.
- Baynat, B., (2000). Théorie des files d'attente. Hermès Science Publication.



**Semestre : 02**

**UE : UEF 1.2.2**

**Intitulé de matière:** Exigences et Développement de Logiciel

### **Objectifs de l'enseignement**

L'ingénierie des exigences est la partie du génie logiciel permettant de déterminer quel système sera développé. A l'issue de ce module, l'étudiant pourra mettre en œuvre les différents processus d'ingénierie des besoins (exigences) nécessaires à la confection d'un cahier des charges cohérent. Ces processus seront appliqués à un projet logiciel

### **Connaissances préalables recommandées**

- Connaissances sur UML et le processus unifié
- Bases de la gestion de projet

### **Contenu de la matière :**

#### **Introduction**

Gestion de projet et exigences logicielles  
Crise du logiciel et génie logiciel  
Enjeux de l'ingénierie des exigences

#### **Processus d'ingénierie des exigences**

Analyse du domaine  
Elucidation des exigences et évaluation des exigences  
Documentation des exigences  
Consolidation des exigences

#### **Classification des exigences**

Exigences fonctionnelles  
Exigences non fonctionnelles

#### **Techniques de l'ingénierie des exigences**

Etude de documents, interview, questionnaire  
Scénarios d'interaction  
Prototypage  
Revue des exigences  
Validation des modèles  
Tests d'acceptation  
Gestion du changement

#### **Modélisation des exigences**

Modélisation des objectifs, des objets, des agents  
Modélisation des opérations et des comportements

**Mode d'évaluation :** Contrôle + TP

**Semestre : 02**

**UE : UEF 1.2.2**

**Intitulé de matière:** Analyse de la qualité du logiciel

### **Objectifs de l'enseignement**

La qualité en logiciel est une notion complexe et globale qui regroupe beaucoup de facteurs. Le rôle et l'importance de chacun de ces facteurs sont parfois difficiles à discerner et à isoler. De plus, la qualité est une notion souvent suggestive qui est perçue différemment selon qu'on la regarde dans la perspective du client-utilisateur ou de l'ingénieur logiciel.

Ce module permettra aux étudiants de comprendre et de pouvoir utiliser les concepts et les méthodes d'ingénierie de la qualité du logiciel ou encore de prendre un part active au sein d'une équipe d'assurance qualité.

### **Contenu de la matière**

#### **Chapitre 1: Concepts de base**

- Exigences qualité
- Modèles de qualité
- Mesures de qualité
- Évaluation de qualité

#### **Chapitre 2: Modèle de la structure de la qualité du produit logiciel**

- Le chemin de décomposition des exigences haut niveau aux catégories des mesures qualité
- Analyse de la traçabilité
- Impact des exigences fonctionnelles sur les exigences qualité

#### **Chapitre 3: Formalisation des exigences qualité**

#### **Chapitre 4: Modèle d'implantation de qualité**

#### **Chapitre 5: Processus d'ingénierie de qualité**

**Mode d'évaluation :** contrôle \*2 + Interrogation

### **Références**

- Suryn W. "Maturing Usability: Quality in Software, Interaction and Value. Part 1: Software Quality Engineering – the leverage for gaining maturity". Springer Publisher, December 14<sup>th</sup>, 2007
- Lauesen S., "Software Requirements". Addison-Wesley, 2002
- Vliet, H., "Software Engineering, Principles and Practice", Second Edition. John Wiley & Sons, 2002
- McGarry, J et al, "Practical Software Measurement". Addison-Wesley, 2002
- Kan S., H., "Metrics and Models in Software Quality Engineering", Second Edition, Addison-Wesley 2003

**Semestre : 02**

**UE : UED 2.1.1**

**Intitulé de matière:** Techniques de maintenance du logiciel

### **Objectifs de l'enseignement**

Ce module vise à présenter des techniques de maintenance du logiciel : Compréhension de programmes, Reengineering, Reverse engineering, Migration, Retirement.

### **Contenu de la matière :**

Ce cours présente les différentes techniques de maintenance du logiciel :

- Compréhension de programmes
- Reengineering
- Reverse engineering
- Migration
- Retirement

**Mode d'évaluation :** *Contrôle \*2 + Interrogation + TP.*

**Semestre : 03**

**UE : UEM 2.1.1**

**Intitulé de matière: Pratique Professionnelle du GL**

**Objectifs de l'enseignement :**

Le développement de logiciel nécessite la coopération d'un ensemble de partenaire (l'utilisateur final, le chef du projet et les membres de l'équipe de développement), l'objectif du module est d'initier l'étudiant au travail de groupe et la technique de communication et la rédaction de rapports.

**Connaissances préalables recommandées :**

**Contenu de la matière :**

- Problématique méthodologique de la recherche scientifique
- Etapes du processus de la recherche
- Méthodes et validité de la recherche
- Recherche de l'information scientifique
- Différentes étapes d'un projet de recherche
- Rédaction scientifique et techniques d'exposés
- Déontologie de la recherche

**Mode d'évaluation :** *examen écrit et/ou exposés*

**Références**

**Sites**

- [aeris.11vm-serv.net](http://aeris.11vm-serv.net)
- [csidoc.insa-lyon.fr/sapristi/](http://csidoc.insa-lyon.fr/sapristi/)
- etc....

**Semestre : 03**

**UE :UEF 2.1.1**

**Intitulé de matière:** Architectures Logicielles

### **Objectifs de l'enseignement**

Ce cours a pour objet de présenter la notion d'architecture logicielle ainsi que son rôle tout au long du cycle de vie du logiciel. Les concepts introduits seront illustrés à travers de nombreux exemples de langages de description d'architecture. Par ailleurs, après avoir étudié les principaux concepts manipulés par l'architecte, un ensemble de pratiques permettant de spécifier, concevoir et évaluer les architectures est proposé. Une étude de cas suivie sera appliquée aux architectures étudiées, et les outils méthodologiques permettront d'en évaluer les possibilités, avantages et inconvénients.

### **Connaissances préalables recommandées**

- *UML et Programmation orientée objet*
- *Programmation Orientée Composants*

### **Contenu de la matière :**

1. Fondements de la notion d'architecture logicielle en génie logiciel
  - Les points de vue architecturaux
  - Styles architecturaux
  - Architectures basées sur les composants
2. Introduction aux langages de description d'architecture logicielle
3. Comparaison de quelques ADL
4. UML et la description des architectures logicielles
5. Le standard AADL

**Mode d'évaluation :** (Moyenne notes TP + 2\*Note-Examen-écrit)/3

### **Références :**

- **Zheng Qin, Jiankuan Xing et Xiang Zheng**, «Software Architecture», Springer, **2008**.
- **Medvidovic N. et Taylor, R. N.**, « A Classification and Comparison Framework for Software Architecture Description Languages », IEEE TSE: 26, pages 70–93, **2000**.
- **Medvidovic N., Rosenblum David S., Redmiles David F., Robbins Jason E.**, «Modeling Software Architectures in the Unified Modeling Language», ACM Trans. Softw. Eng. Methodol., 11(1): pages 2-57, **2002**.
- **Peter H. Feiler, David P. Gluch, John J. Hudak**, «The Architecture Analysis & Design Language (AADL): An Introduction», Software Engineering Institute Carnegie Mellon University, Pittsburgh, PA 15213, February **2006**.
- « Architecture Description Languages » livre IFIP The International Federation for Information Processing, Volume 176 2005, **Editeurs: Pierre Dissaux, Mamoun Filali-Amine, Pierre Michel, François Vernadat**

### **Sites web :**

- [http://fr.wikipedia.org/wiki/Architecture\\_logicielle](http://fr.wikipedia.org/wiki/Architecture_logicielle)



**Semestre : 03**

**UE : UEF 2.1.1**

**Intitulé de matière:** Ingénierie Dirigée par les Modèles

### **Objectifs de l'enseignement**

Ce cours présente les principes de l'ingénierie logicielle dirigée par les modèles. En particulier, le cours aborde le processus de développement logiciel par transformations de modèles et les concepts de modèles indépendants des plateformes et modèles spécifiques aux plateformes. Ce cours couvre aussi les principes et les standards de modélisation et de méta-modélisation, les langages spécifiques aux domaines et l'architecture dirigée par les modèles MDA (Model-Driven Architecture) de l'OMG (Object management group).

### **Connaissances préalables recommandées**

Notation UML, programmation orientée objets.

### **Contenu de la matière**

- Introduction à l'ingénierie logicielle dirigée par les modèles
- Architecture MDA (Model driven architecture)
- Les modèles dans MDA (modèle indépendant de la plateforme vs modèle spécifique à une plateforme)
- Modélisation et méta-modélisation
- Les standards supportant MDA (e.g., MOF, UML, OCL, QVT, BPMN)
- Syntaxe et sémantique d'un modèle
- Spécification des contraintes sur les modèles
- Langage spécifique au domaine
- Spécification des transformations de modèles
- Outils supportant l'ingénierie dirigée par les modèles

**Mode d'évaluation :** TP + examen écrit

### **Références**

- Jean-Marc Jézéquel, Benoît Combemale et Didier Vojtisek. Ingénierie Dirigée par les Modèles : des concepts à la pratique. Ellipses Marketing, 2012.
- Scott W. Ambler. The Object Primer: Agile Model-Driven Development with UML 2.0. 3rd Edition, Cambridge University Press, 2004.
- Stephen J. Mellor, Kendall Scott, Axel Uhl and Dirk Weise. MDA Distilled. Addison-Wesley Professional, ISBN-10: 0201788918
- AnnekeKleppe, Jos Warmer and WimBast. MDA Explained: The Model Driven Architecture--Practice and Promise. Addison-Wesley, 2003.

- Jim Arlow and IlaNeustadt. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design. Addison-Wesley, 2nd Edition.
- Allistair Cockburn. Writing Effective Use Cases. Addison-Wesley, 2000.
- Stephen J.Mellor and Marc J. Balcer. Executable UML: A Foundation for Model Driven Architecture. Addison-Wesley. ISBN-10: 0201748045



**Semestre : 03**

**UE : UEF 2.1.2**

**Intitulé de matière:** SOA et Cloud Computing

### **Objectifs de l'enseignement**

L'essor des Services Web accompagne naturellement la montée en puissance de SOA, qui fait l'objet aujourd'hui d'un réel engouement. En effet, les applications reposant sur des architectures SOA, qui requièrent des Web Services, assurent flexibilité et réactivité du système d'information. Les étudiants comprendront ce que sont les Web Services et leur fonctionnement ainsi que les technologies mises en œuvre dans le cadre des Services Web. Ils étudieront aussi une mise en œuvre pratique du SOA à travers le cloudcomputing.

### **Connaissances préalables recommandées**

- Concepts de base du GL
- Programmation Orientée Composants

### **Contenu de la matière :**

**Chapitre 1 : Introduction aux Web Services**

**Chapitre 2 : Principes d'architecture orientée services (SOA)**

**Chapitre 3 : Créer un Web Service**

**3-1** Classification des services Web

. Les services de communication et de transport (SOAP)

. Les services techniques

. Les services métiers

**3-2** Le développement des services Web

**3-3** Le déploiement des services Web

**Chapitre 4 : Accéder et utiliser un Web Service**

4-1 Concepts de base : fournisseur, demandeur, annuaire

4-2 Les fonctions de découverte, description et échanges

4-3 UDDI

4-4 WSDL

4-5 SOAP et extension SOAP

**Chapitre 5 : Concepts et définition du cloud computing**

5-1 Définition du cloud computing

5-2 Caractéristiques du cloud computing

5-3 Architecture du cloud computing

- Infrastructure système
- Infrastructure réseau
- Les applications
- Virtualisation des accès

#### 5-4 Modèles de services cloud

- SaaS
- PaaS
- IaaS

#### **Chapitre 6 :** Introduction SaaS et Cloud Computing

- Évolution de l'informatique vers les solutions SaaS et Cloud et comparaison de ces solutions.
- Présentation du lien avec les technologies de virtualisation

**Mode d'évaluation :** *Contrôle + TP*

#### **Références**

- 1- « Services Web avec SOAP, WSDL, UDDI, ebXML... » Editions EYROLLES 2002
- 2- « WEB SERVICES ET INTEROPERABILITE DES SI » collection DUNOD 2004

**Semestre : 03**

**UE : UEF 2.1.2**

**Intitulé de matière:** Administration dans les bases de données

**Objectifs de l'enseignement :**

l'étudiant devra apprendre :

- une méthodologie de conception de base de données centralisée ou répartie.
- des connaissances sur l'architecture d'un SGBD-R et les fonctions fondamentales d'administration des bases de données.
- des connaissances sur la sécurité des BDs.

**Connaissances préalables recommandées**

- Savoir concevoir et manipuler des bases de données relationnelles
- Connaissances sur l'architecture du SGBD-R, SQL et évaluation des requêtes.

**Contenu de la matière :**

**1) Introduction et Rappels**

- Définitions et approches générales aux bases de données et aux systèmes de gestion de bases de données.
- Rappel : La modélisation conceptuelle de données.
- Rappel : Les concepts de base du modèle relationnel et les règles de passage du modèle conceptuel au modèle relationnel.
- La théorie de normalisation du modèle relationnel, 4ème et 5ème formes normales

**2) Architecture d'un SGBDR**

- Architecture globale d'un SGBD-R (modules fonctionnels)
- Traitement de requêtes : analyse, optimisation et évaluation de requête.
- organisation de données (gestionnaire de Buffer, gestion de fichiers, ...)
- La gestion des accès concurrents : définition et propriétés, gestion des transaction, Exécution concurrente de transactions (ordonnancement, sérialisation, ...)
- Contrôle de concurrence et Verrouillage des données .
- La gestion de reprise après incident : approches, natures de pannes, stratégies.

**3) Administration et optimisation de bases de données**

- Administration de données et administration de base de données.
- Architecture du SGBD ORACLE (instance, BD, organisation de stockage, ...)
- Fonctions d'administration: Création de BDs, La création de l'instance Oracle, Création d'utilisateurs (droit, privilèges, gestion des droits, rôles, ...), Gestion de schémas et de données, Créer le schéma de la base utilisateur, ...
- surveillance du bon fonctionnement de la base, amélioration des performances, ...

- Gestion de la performance: Règles de base pour garantir une bonne performance de la BD ( Choix pertinent des index, répartition judicieuse des tablespaces, ...)
- Optimisation des requêtes.

#### **4) Approches à la gestion des bases de données réparties ou fédérées**

- Définitions et typologies des BDs réparties/fédérées.
- Techniques utilisées dans la répartition des données.

#### **Contenu du TP**

- Mise en œuvre d'une base de données relationnelle;
- Création d'une base de données (instance Oracle) : tablespaces, tables, vues, index, trigger, procédure, ....
- Gestion des droits d'accès (Grant, Revoke)
- Alimentation d'une base de données.
- Manipulation d'une base de données.

**Mode d'évaluation :** (Contrôle écrit \* 2 + Interrogation écrite + TP) / 4

#### **Références bibliographiques :**

1. Raghu Ramakrishnan, Johannes Geheke, « Database Management System », second edition, McGraw-Hill Higher Education, ISBN 0-07-246535  
<http://www.cs.wise.edu/~dbbook>.
2. Abraham Silberschatz, Henry F. Korth, "DATABASESYSTEM CONCEPTS - SIXTH EDITION", McGraw-Hill, ISBN 978-0-07-352332-3.
3. Philippe Rigaux, « Cours de bases de données », disponible à <http://deptinfo.cnam.fr/new/spip.php?pdoc6637>.
4. <http://www.oracle.com/>

## **Semestre :3**

### **Unité d'enseignement :UEM 2.1.1**

#### **Matière : Programmation Orientée Agents**

#### **Objectifs de l'enseignement**

La programmation orientée agents est l'évolution de la programmation orienté objets. Le but de ce cours est d'introduire la conception à base d'agents logiciels. Il étudie des méthodologies de développement et de la programmation orientées agents mais aussi des différents protocoles d'interactions entre agents afin de maximiser leur efficacité.

#### **Connaissances préalables recommandées**

Programmation objet, Notation UML

#### **Contenu de la matière :**

1. Introduction aux Systèmes Multi-Agents
2. Programmation Orientée Multi-Agent
  - a. Introduction
  - b. Programmation Orientée Agent
    - i. Introduction
    - ii. Langages et Plateformes
    - iii. Plateforme JACOMO
    - iv. Langage JASON
  - c. Programmation Orientée Environnement
    - i. Introduction
    - ii. Plateforme CArtaGo
  - d. Programmation Orientée Interaction
    - i. Introduction
    - ii. Topologie des Modèles d'Interaction
    - iii. Langages de Communication Agent
    - iv. Protocoles d'Interaction
  - e. Programmation Orientée Organisation
    - i. Origine et Fondements
    - ii. Quelques Approches
    - iii. Approche MOISE

#### **Mode d'évaluation :**

- Un examen final : coeff 2
- Examens continus : Coeff 1
- TP : coeff 1

#### **Références bibliographiques**

1. Russell and Norvig (2003). Artificial Intelligence, A Modern Approach (second edition).

2. Ferber, J. (1999). Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence. Addison-Wesley.
3. Wooldrige, M. J. and Jennings, N. R. (1995). Intelligent agents : Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152.
4. Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2011). Multi-agent oriented programming with jacamo. *Science of Computer Programming*
5. Bordini, R., H., Hübner, J. F., and Wooldrige, M. *Programming Multi-Agent Systems in AgentSpeak using Jason* John Wiley & Sons, 2007.
6. Hannoun, M., Boissier, O., Sichman, J. S., and Sayettat, C. (2000). MOISE: An organizational model for multi-agent systems. In Monard, M. C. and Sichman, J. S., editors, *Proceedings of the International Joint Conference, 7th Ibero-American Conference on AI, 15th Brazilian Symposium on AI (IBERAMIA/SBIA'2000)*, Atibaia, SP, Brazil, November 2000, LNAI 1952, pages 152–161, Berlin. Springer.

**Semestre : 03**

**UE : UEM 2.1.1**

**Intitulé de matière:** Méthodes Agiles

### **Objectifs de l'enseignement**

L'approche agile s'est imposée depuis une dizaine d'années comme une alternative aux méthodes classiques de développement de logiciels. L'agilité permet de mieux répondre aux évolutions des besoins des clients en cours de projet. Son approche itérative et incrémentale permet de disposer d'un produit opérationnel dès les premières phases du développement. L'objectif de ce cours est de donner aux étudiants les compétences, concepts et pratiques agiles. Il constitue également une introduction aux différentes méthodes agiles (SCRUM, XP, LEAN).

### **Connaissances préalables recommandées**

Conception orientée objet, notation UML

### **Contenu de la matière**

#### **Chapitre 1 : Caractéristiques des méthodes agiles**

- Les bases de l'agilité
- L'offre agile
- Conformité au modèle agile
  - o Conformité agile du RAD
  - o Analyse de RUP
  - o Analyse des développements libres
  - o Limitations des approches agiles
  - o Les supports agiles
- Distinction entre les approches traditionnelles et agiles
  - Propriétés communes et spécifiques

#### **Chapitre 2 : Méthode ExtremeProgramming**

- o Origines de XP
- o Description de la méthode

#### **Chapitre 3 : Méthode SCRUM**

- o Origines de SCRUM
- o Description de la méthode

#### **Chapitre 4 : Famille Crystal**

- o Origine de Crystal
- o Description de la méthode

#### **Chapitre 5 : Autres méthodes agiles**

- o Lean Software Development
- o Dynamic System DevelopmentMethod
- o FeatureDrivenDevelopment
- o Adaptive System Development

**Mode d'évaluation : (type d'évaluation et pondération)**

- Un examen final : coeff 2
- Examens continus : Coeff 1
- TP : coeff 1

**Références**

- Ken Schwaber, Agile Software Development with Scrum, Pearson Education Edition, 2008
- Tom Poppendieck, Implementing Lean Software Development, From Concept to Cash, Mary and Addison Wesley 2007
- Craig Larman, Agile and Iterative development, Addison-Wesley Professional, 2003