

# **Mastère de recherche en informatique**

## **Description des matières– semestre 1 et 2**

### **Programmation déclarative**

#### **OBJECTIF**

Le but de ce cours est d'introduire le paradigme de la programmation déclarative qui consiste à énoncer les propriétés d'un système de résolution -programme- (à les déclarer) plutôt qu'à décrire les opérations à effectuer comme dans le cas de la programmation impérative. Elle permet de séparer clairement les trois parties d'un logiciel :

- les connaissances sur le domaine d'application
- la formulation du problème à résoudre
- le mécanisme de résolution du problème, général et indépendant des deux autres parties bien qu'opérant sur elles.

Comme représentants de la programmation déclarative, nous étudions la programmation logique (e.g., Prolog) et la programmation fonctionnelle (e.g., Caml, Lisp). Programmer dans ce style permet à l'étudiant de s'initier à une autre façon d'appréhender l'activité de programmation. En effet, le programmeur décrit le problème qu'il veut résoudre en termes de déclarations sans se soucier de la manière avec laquelle son programme sera évalué. Cette tâche est laissée à la charge du système. Il n'y a pas d'affectation. Des mécanismes d'unification (programmation logique) ou de filtrage (programmation fonctionnelle) sont mis en œuvre. La programmation logique mène naturellement à la programmation logique par contraintes (puissance et efficacité d'algorithmes de résolution sur les domaines finis).

*Le plan suivant se base sur Prolog comme exemple de langage de programmation logique et Caml comme exemple de langage de programmation fonctionnelle. Il peut être adapté en fonction d'autres langages choisis.*

#### **PLAN**

##### **Première partie : Programmation logique**

1. Introduction à la programmation logique
2. Bases du langage Prolog
3. Résolution en Prolog
4. Extensions du langage Prolog
5. Les listes en Prolog
6. Problèmes de satisfaction de contraintes (Domaines finis)
7. Résolution d'un problème de satisfaction de contraintes

##### **Deuxième partie : programmation fonctionnelle (basée sur Caml)**

1. Introduction à la programmation fonctionnelle
2. Bases du langage Caml
3. Sémantique de la programmation fonctionnelle

#### **REFERENCES BIBLIOGRAPHIQUES**

1. L. Sterling, E. Shapiro, L'art de Prolog, Ed. Masson, 1990.
2. W. F. Clocksin, Ch. S. Melish, Programmer en Prolog, Ed. Eyrolles, 1987.
3. J.P. Delahaye, Outils logiques pour l'Intelligence Artificielle, Ed. Eyrolles, 1988.
4. C-L. Chang et R.C-T. Lee, Symbolic Logic and Mechanical Theorem Proving, Ed. Academic Press, 1973.
5. A. Thayse et al., Approche logique de l'Intelligence artificielle, De la logique classique à la programmation logique. Ed. Dunod, 1990.

6. A. Tarski, Introduction à la logique, Ed. Gautier-Villars, 1969.
7. J.W. Lloyd, Fondements de la programmation logique, Ed. Eyrolles, 1988.
8. I. Bratko, Programmation en Prolog pour l'intelligence artificielle, Ed. InterEdition, 1988.
9. G.Cousineau et M. Mauny, Approche fonctionnelle de la programmation, Ed. Ediscience, 1995.

## Algorithmique et complexité

### OBJECTIF

L'objectif de ce module est de donner un panorama de structures et de méthodes phares dans divers domaines d'applications algorithmiques. Aborder la résolution de problèmes difficiles en particulier par les techniques d'énumération implicites. On insistera sur l'importance d'analyser et de comparer les performances de différentes solutions algorithmiques. Les thèmes traités porteront essentiellement sur les arbres et les graphes.

### PLAN

1. Arbres-B versus hachage extensible
2. Arbres et arborescence (arbres couvrants de poids minimum, codage de Huffman, ensembles disjoints)
3. Représentation de files de priorités
4. Enveloppe convexe, Triangulation de Delaunay
5. Composantes connexes et fortement connexes
6. Chemins minimaux et flots maximaux dans les graphes
7. Résolution de problèmes difficiles sur les graphes (coloration, cycles eulérien et hamiltonien)
8. Complexité
  - a. Comparaison asymptotique de fonctions
  - b. Complexité d'un algorithme (complexité spatiale, complexité variable ; exemple illustratif (un problème, plusieurs algorithmes))
  - c. Classes de complexité (L et NL ; P, NP et Co-NP, EXPTIME, inclusions des classes)
  - d. Problèmes C-complets ou C-difficiles (C=Classe) et Réduction

### REFERENCES BIBLIOGRAPHIQUES

1. T. Cormen & al., *Introduction à l'Algorithmique*, Dunod, 2002.
2. J. Dréo, A. Pérowski, P. Siarry & E. Taillard, *Métaheuristiques pour l'Optimisation Difficile*, Eyrolles, 2003.
3. S.M. Sait & H. Youssef, *Iterative Computer Algorithms with Applications*, IEEE C.S, 1999.
4. S.S. Skiena, *The Algorithm Design Manual*, Springer-Verlag, 1997.

## Architectures et systèmes évolués

### Plan du cours

1. Paradigmes des systèmes d'exploitation.
2. Avancées technologiques et conceptuelles.
3. Architectures et systèmes parallèles.
4. Architectures et systèmes répartis.
5. Architectures et systèmes mobiles.
6. Systèmes à large échelle.
7. Le Cloud Computing.

## REFERENCES BIBLIOGRAPHIQUES

1. C. Szyperski. Component Software - Beyond Object-Oriented Programming. Addison-Wesley, 2002.
2. B. Goetz, et al. S.D. Java Concurrency in Practice. Addison-Wesley, 2006.
3. D. Serain. Middleware and Enterprise Application Integration. Springer-Verlag 2002.
4. A. Silberschatz, P. Galvin, G. Gagne, Operating System Concepts, 6th ed., Addison-Wesley, 2001
5. A. Grama, A. Gupta, G. Karypis, V. Kumar Introduction to Parallel Computing, Addison-Wesley, 2nd Edition, 2003
6. G. Coulouris, J. Dollimore, K. Kindberg. Distributed systems: concepts and design. 4<sup>th</sup> Edition, 2005.
7. P. Samuel. Réseaux et systèmes informatiques mobiles : Fondements, architectures et applications. Ecole Polytechnique de Montréal, 2003.

## Réseaux nouvelles génération

### Plan du cours

1. Paradigmes des systèmes d'exploitation.
2. Avancées technologiques et conceptuelles.
3. Architectures et systèmes parallèles.
4. Architectures et systèmes répartis.
5. Architectures et systèmes mobiles.
6. Systèmes à large échelle.
7. Le Cloud Computing.

## REFERENCES BIBLIOGRAPHIQUES

8. C. Szyperski. Component Software - Beyond Object-Oriented Programming. Addison-Wesley, 2002.
9. B. Goetz, et al. S.D. Java Concurrency in Practice. Addison-Wesley, 2006.
10. D. Serain. Middleware and Enterprise Application Integration. Springer-Verlag 2002.
11. A. Silberschatz, P. Galvin, G. Gagne, Operating System Concepts, 6th ed., Addison-Wesley, 2001
12. A. Grama, A. Gupta, G. Karypis, V. Kumar Introduction to Parallel Computing, Addison-Wesley, 2nd Edition, 2003
13. G. Coulouris, J. Dollimore, K. Kindberg. Distributed systems: concepts and design. 4<sup>th</sup> Edition, 2005.
14. P. Samuel. Réseaux et systèmes informatiques mobiles : Fondements, architectures et applications. Ecole Polytechnique de Montréal, 2003.

## Processus stochastiques

### Plan du cours

- 1) Rappel et Complément de probabilité pour le cas multivarié
- 2) Fonctions aléatoires et processus Stochastique, Fonction de covariance
- 3) Stationarité et fonction d'autocorrelation, Ergodicité

- 4) Processus discret et chaîne aléatoire,
- 5) Processus et chaîne de Markov application à la reconnaissance de la parole
- 7) Processus Gaussien et processus de Viner
- 8) Processus de Poisson, Application aux files d'attente
- 9) Application à la modélisation aléatoire des signaux. application aux codeurs CDMA
- 10) Champs aléatoires et application à la modélisation des images (2D, 3D et 4D)

## **Modélisation et résolution numérique**

### **Plan du cours**

1. Complément de calcul différentiel
  - Introduction
  - Les espaces  $L(E,F)$  des application linéaires
  - Ensemble compact et théorème de compacité
  - Dérivation des fonctions a plusieurs variables
  - Théorème des accroissements finis
  - Théorème des fonctions implicites
  - Extremum des fonctions réelles
  - Multiplicateurs de Lagrange
  - Critère d'optimalité des fonctions réelles
  - Méthode de Newton
2. Modélisation mathématique et simulation numérique
  - Introduction générale
  - Exemples de modélisation (équation de la chaleur, équation des ondes, le Laplacien...)
  - Calcul numérique en différence finie
  - Remarques sur les modèles mathématique (Pb bien pose, classification des EDP...)
3. Optimisation générale
  - Théorème de projection
  - Existence et unicité de solution
  - Problèmes d'optimisation non contraints (Méthode de relaxation, méthode de gradient, méthode de gradient conjugué)
  - Problèmes d'optimisation sous contraintes (Gradient projeté, méthode de pénalisation, Conditions de Kuhn-Tucker, caractérisation des directions admissibles, méthode d'Uzawa...)

## **Modélisation et Méthodes d'aide à la décision**

### **Objectifs de l'ECUE**

La prise de décision est une activité intrinsèque au métier d'ingénieur/manager. Plus que jamais dans une économie mondialisée, complexe, et pleine d'imprévis, l'entreprise se trouve dans l'obligation de prendre des décisions stratégiques, tactiques, opérationnelles et de lourdes conséquences (financières, humaines...) pour sa compétitivité.

Pour appréhender les problèmes de décision complexes, les futurs ingénieurs et managers doivent disposer des concepts et méthodologies permettant de formaliser un problème de décision. Le cours vise à introduire en première partie un certain nombre de modèles classiques permettant de

représenter et résoudre des problèmes de décision dans différents contextes (décision dans l'incertain, décision multicritère, ...) puis en seconde partie, le cours ciblera particulièrement les méthodes et l'approche statistiques exploitées pour l'aide à la décision.

Comme il est rare qu'une étude d'aide à la décision n'implique pas, une phase de collecte et de traitement de "données". L'analyse d'importantes quantités de données, de nature souvent hétérogène, implique d'aller au delà de l'utilisation de techniques statistiques élémentaires à une ou deux dimensions. Les méthodes utiles se rattachent à l'analyse de données multidimensionnelles qui se fera l'objet de la troisième partie.

## **PLAN**

### **Partie I : Modélisation pour l'aide à la décision**

- Introduction à l'activité d'aide à la décision, concepts de base,
- Décision dans l'incertain, théorie de l'utilité, arbres de décision,
- Décision multicritère et modélisation des préférences,
- Présentation de modélisations de problèmes de décision utilisant les graphes, la programmation linéaire,....

### **Partie II : Méthodes statistiques d'aide à la décision**

- Introduction à l'apprentissage numérique
- Evaluation d'un classifieur
- Apprentissage par mesure de similarité
- Principe du maximum de vraisemblance
- Classifieur bayésien
- Classification multi-classe
- Classification multi-label
- Comparaison des classifieurs

### **Partie III : Aide à la décision sur la base d'analyse de données multidimensionnelles**

- L'analyse en composantes principales;
- L'analyse des correspondances;
- L'analyse des correspondances multiples;
- Les méthodes de classification automatique;
- Les méthodes d'analyse discriminante;
- Le modèle linéaire, le modèle logistique;
- L'analyse conjointe.

### **Références bibliographiques**

1. H.P. Williams. Model building in mathematical programming. J. Wiley, New York, 1999.  
4ème édition, Ph. Vallin
2. D. Vanderpooten. Aide à la décision : une approche par les cas. Ellipses, Paris, 2002., 2ème édition
3. W. Cooper, L. Seiford, and K. Tone. Introduction to Data Envelopment Analysis, an its use, Springer, 2006

4. Godbole, S. & Sarawagi, S. (2004), 'Discriminative Methods for Multi-labeled Classification', Conference on Knowledge Discovery and Data Mining (PAKDD 2004).
5. Crucianu, M., Asselin de Beauville, J.-P., Boné, R., Méthodes factorielles pour l'analyse des données, Hermès - Lavoisier, 2004.
6. Escofier, B., Pagès, J., Analyses factorielles simples et multiples, Dunod, Paris, 1998.
7. Lebart, L., Morineau, A., Piron, M., Statistique exploratoire multidimensionnelle, Dunod, Paris, 2000.

## **Intelligence Artificielle**

### **OBJECTIF**

Ce cours introduit les briques de base de l'IA et s'intéresse surtout à l'aspect représentation des connaissances et résolution de problèmes à base de recherche arborescente heuristique.

### **PLAN**

1. Paradigmes de l'IA
2. Représentation de connaissances
3. Systèmes experts
4. Résolution arborescente :
  - 4.1. Recherche aveugle :
    - 4.1.1. Recherche en profondeur
    - 4.1.2. Recherche en largeur
  - 4.2. Recherche heuristique
    - 4.2.1. L'algorithme A\* et ses dérivés
    - 4.2.2. L'algorithme AO\* et ses dérivés

## **Reconnaissance de formes et images**

### **OBJECTIFS DU COURS**

Ce cours présente les principales méthodes de base de la reconnaissance des formes. Ces méthodes seront illustrées dans le cadre d'applications en vision par ordinateur et en analyse d'images médicales.

### **PLAN DU COURS**

- Introduction : Exemples, Chaîne de Reconnaissance.
- Descripteurs d'images et extraction des primitives.
- Prétraitement : Modélisation, Analyse en Composantes Principales (ACP), Critère de Fisher et Analyse Factorielle Discriminante (AFD).
- Méthodes Statistiques : décision Bayésienne, apprentissage supervisé, apprentissage non supervisé.
- Fonctions discriminantes linéaires et réseaux neuronaux : fonctions discriminantes linéaires, Machines à vecteurs supports (SVMs), réseaux neuronaux.
- Applications en imagerie.

### **REFERENCES BIBLIOGRAPHIQUES**

- M. Milgram, Reconnaissance des formes. Dunod, 1993.
- M. Kunt, Reconnaissance des formes et analyse de scènes. PPUR, 2000.

- J. Asselin de Beauville et F. Kettaf, Bases théoriques pour l'apprentissage et la décision en reconnaissance des formes. Cépaduès, 2005.
- D. Lingrand, Introduction au traitement d'images. Vuibert, 2004.
- Henri Maître, Le traitement des images. Lavoisier, 2003.
- P. Coquerez, Analyse d'images : filtrage et segmentation. Masson, 1997.

## Calculabilité et décidabilité

### OBJECTIF

Ce cours a pour objectifs de :

- consolider les notions de langages, grammaires et automates étudiées dans le cours de licence *théorie des langages* et
- permettre une ouverture vers les notions de calculabilité et de décidabilité. Ce qui se ramène à l'étude de ce qu'il est possible ou non de résoudre par un algorithme (un ordinateur) indépendamment de l'évolution de la technologie.

Les *machines de Turing* et les *fonctions récursives* sont les modèles de calcul étudiés. Démontrer l'indécidabilité d'un problème est étudiée par les deux techniques de diagonalisation et de réduction. Un éventail de problèmes non décidables est présenté. Enfin sont introduites les notions de complexité et les problèmes NP-complets.

### PLAN

1. Introduction et rappels
2. Propriétés des langages, fermetures,...
3. Les Machines de Turing
4. Les fonctions récursives
5. La non-décidabilité
6. La complexité

### REFERENCES BIBLIOGRAPHIQUES

1. Introduction à la calculabilité, Cours et exercices corrigés. Pierre Wolper. Dunod 2001.
2. Elements of the theory of computation. Harry R. Lewis et Christos H. Papadimitriou. Prentice-Hall 1981.
3. Introduction to automata theory, languages, and computation. John E. Hopcroft et Jeffrey D. Ullman. Addison-Wesley 1979.
4. Logique, informatique et paradoxes. Jean-Paul Delahaye. Belin 2003.
5. Langages formels, Calculabilité et Complexité. Formation interuniversitaire en informatique de l'École Normale Supérieure. Olivier Carton. VUIBET 2008. Version téléchargeable de : <http://www.win.tue.nl/~gbisson/tea/lfcc.pdf> (juillet 2010).

## Paradigmes des langages de programmation

### OBJECTIF

- Etudier les concepts et caractéristiques des langages de programmation
- Se familiariser avec les principes de conception, les différentes sémantiques des langages de programmation

- Se familiariser avec les principales méthodes de la sémantique formelle : opérationnelle, dénotationnelle, et axiomatique
- Introduire les étudiants à la théorie des langages de programmation y compris : calculs, calcul lambda et algèbres sémantiques.

## PLAN

- Conception et implémentation de langage: Influences sur la conception du langage, critères et trade-offs. Compilation, interprétation
- Revu des langages de programmation majeurs: Environnement, contributions, motivation de développement
- Langages de programmation Logiques: Calcul de prédicats, Prolog, résolution
- Sémantiques: sémantique dénotationnelle des expressions et des affectations conditionnelles, et boucles. Introduction à la sémantique axiomatique et opérationnelle.
- Variables et leurs attributs: Binding, scope (statique et dynamique), durée-de-vie (lifetime), environnement de référencement, types.
- Types des données abstraits: Conception et implémentation des différents types de données-- types primitifs, types énumérés, arrays, records, pointers. Equivalence de type – structurelle, nom. Vérification de type, inférence de type.
- Abstraction de données et abstraction de types de données. Encapsulation, génériques
- Expressions et opérateurs: Overloading, coercion, short-circuit opérateurs, functional side effects.
- Langages de programmation fonctionnelle: Lambda-calculus, ML, Scheme
- Statements: Conception et implémentation des affectations de sélection, de répétition.
- Sousprogrammes: Conception et implémentation des environnements surcharges (overloaded), génériques, référencement local et non-local, scope statique, scope dynamique, side effects.
- Passation de paramètres: Méthodes (par valeur, résultat, etc), implémentations, et fonctions comme paramètres.
- Objets

## REFERENCES BIBLIOGRAPHIQUES

1. Kenneth C. Loudon, *Programming Languages : Principle and Practice*, 2<sup>nd</sup> edition, PWS Publishing Company, 2004

## Ingénierie des méthodes et des processus

### OBJECTIF

Ce thème recouvre la *modélisation* des processus de développement de systèmes d'information, leur *spécification*, la définition et l'implémentation de *mécanismes d'exécution* et leur *évolution*, y compris de manière dynamique.

On s'intéresse essentiellement aux deux aspects complémentaires que sont

- 
- 
- la *modélisation/spécification* des processus de développement
  - le *monitoring de leur exécution* par des mécanismes logiciels appelés moteurs d'exécution.



Nous nous plaçons alors dans un référentiel à trois niveaux : celui des processus, celui des modèles de processus et celui des méta-modèles de processus. Les recherches portent en premier lieu sur le niveau méta avec comme objectif

- de définir un (ou plusieurs) méta-modèles permettant de représenter des aspects des processus que ne capturent pas les modèles centrés 'activités' tels que le 'Waterfall model'
- de définir un moteur générique adapté au méta-modèle qui réponde à différents objectifs de trace, de guidage, d'adaptation au contexte et de contrôle

Afin de valider les propositions du méta niveau, on s'intéresse au mécanisme d'*instanciation* par lequel un modèle de processus peut être obtenu à partir du méta-modèle et par lequel le moteur peut être personnalisé au modèle.

Les approches développées sont au nombre de trois :

- Un **cadre de référence** de l'ingénierie des processus de développement de systèmes
- L'approche **NATURE**
- L'approche **MAP**

Le **cadre de référence** a été défini pour *faciliter la compréhension et classer les problèmes relatifs aux processus de développement* de systèmes d'information ou logiciels. Il s'agit d'un cadre à facettes inspiré du système à facettes introduit par Prieto Diaz pour classer des composants orientés objet en vue de leur réutilisation

L'approche **NATURE** s'intéresse à des processus qui ne peuvent pas se représenter comme un flux prédéfini d'activités. Les activités ne sont pas toutes d'un type connu à l'avance, elles s'enchaînent en fonction des circonstances et des situations rencontrées 'on the fly' pendant le déroulement du processus, selon un ordre qu'il n'est pas possible de préfigurer à l'avance ; les agents y prennent des décisions et ne sont pas seulement des exécutants de tâches typées et ordonnancées à l'avance.

Le système de représentation des processus d'ingénierie de systèmes d'information, **Map** est une extension du modèle NATURE. Il vise à permettre la représentation dans un même modèle, de plusieurs façons de conduire un processus d'ingénierie. Il s'agit donc d'une représentation *multi-modèle* permettant la construction du modèle suivi pendant le déroulement d'un projet de manière dynamique à partir des *multi* modèles contenus dans la représentation. Le système *Map* échappe à la linéarité des modèles de processus classiques, va plus loin que la possibilité de sélectionner une alternative parmi plusieurs pour exécuter une tâche telle que l'offre NATURE afin de permettre une construction dynamique du modèle de processus suivi.

## REFERENCES BIBLIOGRAPHIQUES

- [Armenise93] P. Armenise, S. Bandinelli, C. Ghezzi, A. Morzenti, "A survey and assessment of software process representation formalisms", Int. Journal of Software Engineering and Knowledge Engineering, Vol. 3, No. 3, 1993.
- [CMM] <http://www.sei.cmu.edu/cmm/>
- [Dowson93] M. Dowson, "Software Process Themes and Issues ", IEEE Int. Conf. , 1993.
- [Jarke94] M. Jarke, K. Pohl, C. Rolland, J. R. Schmitt, "Experienced-Based Method Evaluation and Improvement : A Process Modeling Approach ", Int. IFIP WG8. 1 Conference in CRIS series : Method and associated Tools for the Information Systems Life Cycle", North Holland (Pub. ), 1994.

- [Olle91] T. W. Olle, J. Hagelstein, I. MacDonald, C. Rolland, F. Van Assche, A. A. Verrijn-Stuart, "Information Systems Methodologies : A Framework for Understanding ", Addison Wesley (Pub. ), 1991.
- [Pohl 96] Pohl K., (1996). *Process centred requirements engineering*, J. Wiley and Sons Ltd.

## **Méthodes formelles**

### **OBJECTIF**

Introduire les techniques et méthodes formelles ainsi que leur utilisation dans la spécification et la vérification de systèmes logiciels.

Le contenu du cours est déployé sous forme de composants. Au sein de chaque composant plusieurs volets sont proposés. La méthode de vérification qui sera adoptée dans le composant 5 et l'outil utilisé dans le composant 6 dépendront du choix cohérent des volets, qui sera fait par l'enseignant au sein des composants 3 et 4.

### **PLAN**

#### **Composant 1 : Motivations**

1. Cohérence du logiciel, vérification et validation
2. Nécessité de méthodes formelles pour la spécification et la vérification
3. Définition des méthodes formelles
4. Motivation à travers un exemple illustratif

#### **Composant 2 : Spécification formelle**

1. Différents types de spécification
2. Types abstraits
3. Spécification algébrique
4. Notation Z

#### **Composant 3 : Formalismes de modélisation de systèmes**

1. Automates
2. Réseaux de Petri
3. Logiques Modales
4. Algèbres de processus

#### **Composant 4 : Formalismes d'expression de propriétés**

1. Logiques temporelles (Linéaires, arborescentes)
2. Automates sur des mots infinis (Büchi, Muller, Rabin)

#### **Composant 5 : Méthodes de vérification**

1. Vérification basée modèles (Model-checking)
2. Preuve de théorèmes

#### **Composante 6 : Outils**

1. UPPAL, SPIN, Kronos, Coq, PVS ...

## **REFERENCES BIBLIOGRAPHIQUES**

1. Introduction aux méthodes formelles, Jean François Monin, Ed Hermes 2000
2. Systems and Software Verification , **Berard**, B., **Bidoit**, M., **Finkel**, A., **Laroussinie**, F., **Petit**, A., **Petrucchi**, L., **Schnoebelen**, P. , Ed Vuibert 1999
3. Formal Method Europe, <http://www.fmeurope.org/>

# **Conception des bases et entrepôts de données**

## **Objectifs de l'ECUE**

Ce cours vise à familiariser les étudiants aux concepts :

- D'intégration des données hétérogènes dans un SI
- De conception des bases de données OLTP et les bases de données OLAP
- D'entrepôts de données (Datawarehouse)
- De prise de décisions stratégique au sein d'un système d'information (SI) à travers l'utilisation des outils d'intelligence d'affaire (Business Intelligence) et de Datamining

## **Chapitre I**

Introduction : Systèmes d'information et prise de décision

Section I: Rappel des systèmes d'information

Section II: Cycle de prise de décisions dans un SI

Section III : Définition et typologie des décisions dans un SI

Section III: Caractéristiques des décisions

Section IV: Sources des décisions

## **Chapitre II**

Introduction : Hétérogénéité des données dans un SI

Section I : Bases de données OLTP : Conception et normalisation

Section II : Sources de données hétérogènes et besoins d'intégration

Section III : Les concepts et outils d'intégration des données

Section IV : Le processus Extract Transform Load (ETL)

## **Chapitre III**

Introduction : Entrepôts de données ou Data warehouse (DWH)

Section I : Caractéristiques des DWH

- Bases de données OLAP
- Processus de dé-normalisation

Section II : Conception des Data warehouse

- Conception multidimensionnelle
- Notion d'indicateur et de faits
- Notion de dimension et d'axes d'analyse
- Typologie des schémas de Data Warehouse (modèle en étoile, flocon et constellation)

## **Chapitre IV**

Introduction : Restitution des données et reporting

Section I : Les mécanismes de restitution de données

- Reporting
- Tableau de bord des SI

Section II : Data Mart

- Notion de cube olap
- Extension olap du langage SQL

## **Chapitre V**

Introduction : le datamining

Section I : Les algorithmes du datamining

- Classification
- Estimation
- Prédiction
- Groupement par similitudes
- Analyse des clusters

## **Références bibliographiques.....**

1. Jarke M. et al., Fundamentals of data warehouses, Springer-Verlag, 2003
2. Kimball R.,(1997), Entrepôts de données : guide pratique du concepteur de "data warehouse", International Thomson publishing France.
3. Muckenhirn P., Le système d'information décisionnel, Hermès Science Publication, 2003
4. Thomsen E., OLAP Solutions, Wiley, 2002
5. Les nouveaux tableaux de bord des managers, Alain Fernandez, Editions Eyrolles, 5<sup>ème</sup> édition 2011.

## **Indexation et recherche par le contenu des images et des vidéos**

### **Objectifs de l'ECUE**

Les progrès technologiques sont tels en matière de stockage des données que le problème n'est plus d'obtenir des informations mais bien de pouvoir récupérer une information pertinente contenue dans des bases de données gigantesques. Afin de répondre à ce besoin, des outils de plus en plus efficaces ont été développés au cours du temps : catalogues (descriptions des documents par des éléments extérieurs au contenu : auteur, date,...), classification (description globale du contenu d'un document) et indexation. Cette dernière consiste à extraire des documents (textuels, sonores ou visuels) une information condensée analysant leur contenu et permettant la récupération du document original. Les applications sont multiples : web sémantique, aide à la conservation, à l'exploitation (voire la vente) des œuvres culturelles (sons, images...), aide au diagnostic médical,

...

### **Plan**

- Principe et enjeu de l'indexation
  - Indexation et récupération de l'information
  - Evaluation des systèmes d'indexation
- Indexation de textes
  - Extraction de l'information contenue dans les textes
  - Problématique des documents textuels numérisés
- Indexation des images (fixes ou animées)
  - Extraction de l'information contenue dans les images (couleur, forme et texture)
  - Extraction de signatures
  - Mise en forme de cette information (segmentation, description sémantique, modèles 3D,...)
  - Requêtes à base d'exemples (notion d'apprentissage) et requêtes sémantiques
- Indexation audio
  - Extraction de l'information contenue dans les documents audionumériques (en particulier gestion du signal temporel)
  - Mise en forme de cette information
  - Requêtes à base d'exemples et requêtes sémantiques
- Indexation dans des masses de données, fouille de donnée
- Application : Séquences d'Images - de la pellicule au flux vidéo
  - Restauration et archivage de films cinématographiques
  - Sémantique de l'indexation video

- Découpage en plans
- Détection et Suivi d'objet
- Indexation de flux vidéo
- Analyse multi-modale
- Récupération de l'information (*retrieval*)
- Application : Système d'Informations Documentaires - du Patrimoine au document numérique

## **Références bibliographiques**.....

S. Deb, Multimedia Systems and Content-Based Image Retrieval, Information Science, 2003.  
O. Marques et B. Furht, Content Based Image and Video Retrieval, Multimedia System and Applications, Springer, 2002.

## **Parallélisme et calcul de haute performance**

### **Objectifs de l'ECUE**

- Comprendre les concepts de base de l'architecture et de la programmation parallèles.
- Etudier les techniques de parallélisation.

## **PLAN**

### **Chapitre I** : Introduction

Introduction :	.....
Section I : Historique.....	.....
Section II : Les deux mythes du parallélisme.....	.....
Section III : Domaines d'application du parallélisme .....	.....
Section IV : Plan du cours.....	.....

### **Chapitre II** : Architectures parallèles

Introduction	
Section I : Architecture séquentielle classique.....	.....
Section II : Architectures parallèles.....	.....
Section III : Taxonomie de haut niveau.....	.....
Section IV : Machines SIMD.....	.....
Section V : Machines MIMD.....	.....
Section VI : Les clusters.....	.....
Section VII : Les grilles.....	.....
Section VIII : Systèmes distribués à grande échelle et systèmes P2P.....	.....
Section IX : Cloud -Computing.....	.....
Section X : Les architectures multi-cœurs.....	.....

### **Chapitre III** : Topologies des réseaux

Introduction :	.....
Section I : Réseaux d'interconnexion .....	.....
Section II : Réseaux dynamiques .....	.....
Section III : Réseaux statiques .....	.....

Section IV : Exemples de machines et leurs topologies .....	
---	--

## **Chapitre IV** : Métriques d'évaluation de performances

Introduction : .....	
Section I : Mesure des temps d'exécution .....	
Section II : Expression des performances.....	
Section III : Sources de perte de performances.....	
Section IV : Loi de Amdahl .....	
Section V : Loi de Gustafson .....	
Section VI : Liens Amdahl-Gustafson .....	

## **Chapitre V** : Sources du parallélisme

Introduction : .....	
Section I : Trois sources .....	
Section II : Parallélisme de contrôle/tâches.....	
Section III : Parallélisme de données .....	
Section IV : Parallélisme de flux .....	
Section V : Exemple illustratif .....	
Section VI : Principales limitations des sources .....	

## **Chapitre VI** : Parallélisme de tâches

Introduction : .....	
Section I : Segmentation en tâches .....	
Section II : Analyse de dépendance/précédence.....	
Section III : Exercice .....	
Section IV : Calcul du temps optimal $T_{opt}$ .....	
Section V : Décomposition d'un graphe de précédence .....	
Section VI : Calcul des degrés de parallélisme $P_{opt}/P_{max}$ .....	
Section VII : Le problème de l'ordonnancement .....	
Section VIII : Algorithmes de détermination des degrés de parallélisme.....	

## **Chapitre VII** : Parallélisation de programmes polyédriques

Introduction : .....	
Section I : Etude et analyse des dépendances .....	
Section II : Transformations et restructurations de nids de boucles.....	
Section III : Etude de boucles conditionnelles.....	

## **Chapitre VIII** : Environnement MPI

Introduction : .....	
Section I : Structure d'un programme MPI .....	
Section II : Communications point-à-point.....	
Section III : Envoi synchrone bloquant .....	
Section IV : Envoi synchrone non-bloquant .....	
Section V : Types de données dérivés.....	

## **Références bibliographiques**

1. J. CHERGUI & al., Message Passing Interface (MPI-1), Institut Du Développement et des Ressources en Informatique Scientifique, [http://www.idris.fr/data/cours/parallel/mpi\\_orig/mpi1\\_cours\\_couleurs.pdf](http://www.idris.fr/data/cours/parallel/mpi_orig/mpi1_cours_couleurs.pdf), 2009.
2. D. E. CULLER & J. P. SINGH, Parallel Computer Architecture, A hardware / Software Approach, Morgan Kaufmann Publishers, San Francisco, California, 1999.
3. Z. MAHJOUB, Architectures et Algorithmique Parallèles, Cours pour Ingénieur Diplômé en Informatique, 1997.
4. Z. MAHJOUB, Architectures et Algorithmique Parallèles, Cours pour Mastère de Recherche en Informatique, 2011.
5. H. S. MORSE, Practical Parallel Computing, academic Press, Cambridge, 1994.
6. J.P. SONSONNET, Architecture des machines parallèles, <http://www.limsi.fr/~jps/enseignement/tutoriels/archi/archi.htm>.
7. S. VIALLES, Calcul parallèle et distribué et Grilles de calculs, Supelec, <http://www.metz.supelec.fr/metz/personnel/vialle/course/SI-PP/index.htm>, 2009.
8. M. J. FLYNN, Computer Architecture : Pipelined and Parallel Processor Design, Jones and Bartlett, 1995.
9. E. D'HOLLANDER, G.R. JOUBERT, F.J. PETERS & D. TRYSTRAM, Parallel Computing: State-of-the-Art and Perspectives, Elsevier, 1996.
10. R. CYPHER & J.L.C. SANZ, The SIMD Model of Parallel Computation, Springer-Verlag, 1994.
11. D. SKILLICORN, Foundations of Parallel Programming, Cambridge University Press, 1994.
12. H.S. MORSE, Practical Parallel Computing, AP Professional, 1994.
13. D. E. CULLER & J. P. SINGH, Parallel Computer Architecture, Morgan Kaufmann Publishers, 1999.

## **Initiation à la recherche pédagogique**

### **Objectifs de l'ECUE**

Il s'agit d'une formation sous forme de séminaires qui permet de consolider les acquis théoriques des différents unités d'enseignements et qui vise à initier l'étudiant à la recherche scientifique et aux différents aspects pédagogiques